Machine learning for early detection of structure loss in the Czochralski process

Leander Jacob Nielsen Eikås

Thesis for Master of Science Degree at the University of Bergen, Norway

2024

Year:	2024
Title:	Machine learning for early detection of structure loss in the Czochralski process
Author:	Leander Jacob Nielsen Eikås
Supervisor:	Martin Møller Greve
Co-supervisor:	Nello Blaser, Frank Øvstetun, Michael Lindbak

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Martin Møller Greve, for his support, guidance, and constructive feedback throughout the work on this thesis. His availability at all times, quick responses to my questions, and motivating discussions around the topic were invaluable. I also extend my thanks to Nello Blaser for his crucial guidance in machine learning, especially at the start of this work. His expertise and direction were particularly important in shaping this research.

My heartfelt thanks go to Frank Øvstetun and Michael Lindbak for their extensive knowledge about the Czochralski process and for providing the necessary data. Their willingness to deliver additional data whenever needed and their active interest in the project, demonstrated through regular meetings, greatly contributed to the success of this thesis.

I also wish to thank my fellow students for their camaraderie and support throughout this journey. Last but not least, I am deeply grateful to my family for their constant support. Your encouragement and care have been a pillar of strength for me. I would especially like to thank my sister, Linelotte, for her assistance in reviewing the thesis.

Leander Jacob Nielsen Eikås
Bergen, June 3, 2024

Abstract

This thesis investigates the use of machine learning techniques to predict structural loss in the Czochralski process. The Czochralski process is the industry standard for producing high-quality mono-crystalline silicon ingots for solar cells. The study evaluates the performance of three machine learning models; logistic regression, random forest, and neural network models across four regions of the ingot: neck, crown, shoulder, and body. The primary goal is to determine which model offers the best predictive performance for early detection of structural loss, thereby enhancing the efficiency and yield of the Czochralski process.

The research reveals that the random forest model consistently delivers the highest accuracy, precision, and recall, especially in the neck and crown regions. This model effectively identifies the early signs of structural loss, making it a valuable tool for improving the process. However, all models faced difficulties in the shoulder and body regions, indicating the need for further refinement and more targeted features.

Additionally, a time-saving model was developed to find time saved during the process by using the random forest model. By maintaining an accuracy threshold of 70%, this model achieved significant time savings, reducing the time required for remelting operations by 16 to 21.5 hours when tested on 51 ingots. These results highlight the potential of machine learning to enhance the Czochralski process, reducing production time and improving the quality of silicon ingots.

Overall, the results demonstrate the potential for machine learning to significantly improve the Czochralski process, by enabling early detection of structural loss. Thereby, reducing the time required for remelting operations and enhancing ingot quality.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Solar energy is one of the most promising renewable energy sources, with silicon solar cells currently leading the market in terms of price and efficiency. They constitute over 95% of the solar cells produced worldwide today, and the fabrication market dominated by China. According to the International Energy Agency (IEA) in 2022, China accounts for more than 80% of the global market share across all manufacturing stages of solar energy, projected to rise to more than 95% in the coming years [10]. Currently, China supplies over 95% of the solar panels used in Europe [11]. Most European manufacturers of silicon solar cells struggle to compete with their counterparts in China. However, the last European manufacturer of silicon wafers is in fact located in Årdal, Norway, and the company is NorSun. NorSun is a Norwegian company that specializes in manufacturing and marketing high-performance mono-crystalline silicon ingots and wafers [1]. In March 2023, the Nanophysic-group at the university had the privilege of visiting the NorSun factory, as depicted in Figure 1.1. We received a guided tour from Frank Øvstetun, during which we observed each stage of the production process firsthand, from the stacking of pure silicon to the cleaning of the wafers. This visit provided me with a valuable introduction to the production of mono-crystalline silicon wafers before I started this thesis.

The Czochralski process is the primary method for the production of mono-crystalline silicon for solar cells. The process begins with high purity silicon being melted in a crucible. A mono-crystalline seed crystal is then dipped into the melt, after which the seed is pulled upwards to create the ingot. This is subsequently cut into thin wafers. The process is complex and involves numerous delicate steps. It is therefore essential to maintain strict control of

Figure 1.1: Guided tour through the NorSun factory.

the material properties and mechanics in order to guarantee the quality and performance of the final solar cell. [12]

One of the key challenges in the Czochralski process is the formation of dislocations within the crystal structure of the ingots, known as structure loss. Dislocations may influence the recombination characteristics of solar cells by creating defect states in the silicon bandgap, which can reduce the lifetime of minority carriers. Consequently, this may affect the performance of silicon solar cells. A successful ingot is considered to be free from dislocations and structure loss. Currently, operators at the NorSun factory detect structural loss by observing the disappearance of the so-called 'growth ridge' or 'node' during the growth phase [13]. If structure loss is detected in an ingot, the affected area can either be remelted or cut away. Early detection of such defects is crucial to minimize the time spent growing ingots with structural loss.

To maintain their competitive edge, the mono-crystalline silicon industry must focus on enhancing efficiency while maintaining high quality standards. One potential solution would be to integrate machine learning into the Czochralski process, thereby enabling the early detection of structural loss in the ingot before it becomes visible to the operator. The data generated by the process could then be used to train a machine learning model, which would learn patterns between the parameters to predict between two outcomes (structure loss or not structure loss), also known as binary classification. Multiple machine learning algorithms are capable of performing precise binary classifications. Among these, logistic regression is one of the most popular due to

Figure 1.2: The Czochralski puller area in NorSun's factory. [1]

its ease of implementation and reliable results. There are more advanced machine learning algorithms, such as random forest and neural networks, but this does not guarantee that they will outperform logistic regression. Depending on the data, it is possible for a model like logistic regression to outperform more advanced models, especially when minimal computing power is required.

Over the past few years, machine learning algorithms have seen a notable increase in use in research. They are used in a wide range of scenarios, from healthcare to manufacturing, where they are often used to make predictions or decisions. Machine learning algorithms analyse and learn patterns from data to improve performance of a given task. This approach has revolutionised many industries, enabling them to work more efficiently and reduce costs.

1.2 Objectives

The primary objective of this thesis is to ascertain the efficacy of three machine learning algorithms in predicting early structure loss in ingots before it has been detected by the operator. In order to achieve this objective, three predictive algorithms will be employed: logistic regression, random forest and neural network. The three models will be compared in order to ascertain which is the most effective for this type of data.

The secondary objective is to use the most effective model to determine the potential time savings in the Czochralski process that could be achieved by trusting the predictions made by the model. This will be accomplished by comparing the remelting times when the model demonstrates optimal performance with the times when structural loss is detected by the operator.

1.3 Contribution

The contribution of this work will enhance the efficiency of mono-crystalline silicon ingot production. Implementing this model into the Czochralski process to predict structural loss could enable early detection of issues before they become visually apparent to operators. This early detection would likely reduce production time and increase yield, resulting in lower costs for wafers and boosting solar cell production for the European market.

1.4 Thesis Outline

This thesis is structured to systematically explore the application of machine learning models for predicting structural loss during the Czochralski process. Chapter 3 provides an overview of the fundamental concepts necessary to understand this thesis. It begins with an introduction to semiconductor basics, before going on to describe the Czochralski process in detail. Finally, it presents an analysis of the machine learning models used in this study: logistic regression, random forest, and neural networks. Chapter 3 introduces the data utilized in this study, explaining its role in the Czochralski process and its preparation before integration into the machine learning models. Chapter 4 outlines the selection of machine learning models and the procedures for enhancing these models through data input adjustments. Additionally, this chapter introduces the time-saving model, which estimates potential time savings based on its predictions. Chapter 5 presents the results of the thesis and provides a detailed discussion of the findings. Finally, Chapter 6 offers conclusions and suggestions for future work.

Chapter 2

Theoretical Background

The production of high-quality mono-crystalline silicon ingots through the Czochralski process is essential for the semiconductor and photovoltaic industries. This chapter provides the theoretical foundation necessary to understand the Czochralski process and the application of machine learning models for predicting structural loss.

First, semiconductor basics are explored, covering key concepts such as energy bands, doping, recombination, and properties of crystalline silicon. These concepts form the basis of the material science underlying the Czochralski process. Next, the Czochralski process itself is detailed, describing its stages: neck, crown, shoulder, and body. Each stage presents unique challenges that influence the quality of the silicon ingots. Following this, supervised learning is introduced, a subset of machine learning where models are trained on labeled data to make predictions. The supervised learning models used in this study are discussed: logistic regression, random forest, and neural networks. Each model's theoretical framework, advantages, limitations, and the key features influencing their predictive performance are examined.

2.1 Semiconductor

Semiconductors are fundamental materials in the production of silicon solar cells, which are pivotal to the renewable energy sector. This section provides an overview of the physical properties of semiconductors, focusing on silicon. It covers the atomic structure, the significance of energy bands, and the role of dopants in enhancing conductivity. Understanding these properties is crucial for optimizing the Czochralski process and improving the efficiency and performance of silicon solar cells.

Semiconductors, such as Silicon (Si), consist of atoms arranged in a regular, periodic lattice. Each atom is part of a structure that collectively shares eight electrons, achieving stability and electrical neutrality. At the center of each atom is a nucleus containing positively charged protons and neutral neutrons, with an equal number of negatively charged electrons orbiting the nucleus. This balance of charges makes the atom electrically neutral. In semiconductors like silicon, each atom typically forms four covalent bonds with its nearest neighbors, resulting in the sharing of eight electrons with four adjacent atoms to achieve a stable electronic configuration. This bonding structure is essential to the semiconductor's properties and capacity to conduct electricity under specific conditions.

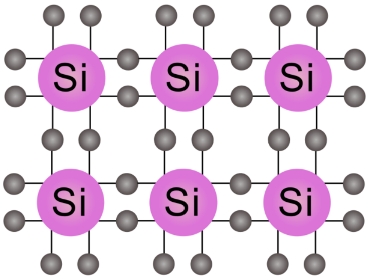


Figure 2.1: Illustration of the semiconductor structure.

The electrons are not considered ”free” as they are held in place by the bond at absolute zero. However, electrons can gain enough energy at higher temperatures to escape their bonds. Then, the electrons can move freely and participate in conduction. This makes a semiconductor behave as an insulator at low temperatures and a conductor at higher temperatures. When the electrons have gained a certain minimum energy called the ”band gap,” the free electrons will participate in conduction. When an electron moves, it allows the covalent bond to move from one electron to another, leaving a space behind called a ”hole” [14].

2.1.1 Energy Bands

As mentioned above, a semiconductor's band gap (E_G) is the minimum energy needed to excite an electron from its bound state and into a free state where it can participate in conduction. The band structure is called a band diagram; the semiconductor's lower energy level is called the valence band (E_V), and the higher energy level where electrons are considered free is called the conduction band (E_C), as shown in Figure 2.2. Therefore, the bandgap is the gap between the valence band and conduction band [14].

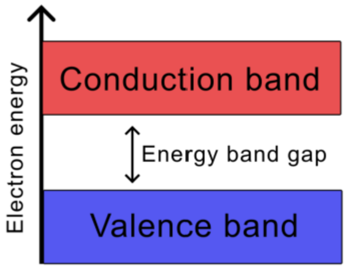


Figure 2.2: The energy band diagram of a semiconductor.

Suppose the minimum energy for the conduction band and the maximum energy for the valence band occur at the same value of wavenumber. In that case, the energy bandgap is called direct. Otherwise, the energy band gap is called indirect. This is essential to know due to the effect it has on the absorption and emission of light [15].

Temperature can affect the energy gap of semiconductors, with the band gap decreasing as temperature increases. This poses a significant challenge in the solar cell industry, where cells are exposed to varying temperatures. According to Ravindra and Srivastava [16], the temperature-dependent electron lattice interaction causes a shift in the relative positions of the valence and conduction bands, which accounts for the majority of the temperature dependence of the energy gap in semiconductors. As a result of the increased thermal energy, the amplitude of the atomic vibrations rises, increasing the interatomic distance. This means that the average potential that the material's electrons detect drops with increasing interatomic distance, which therefore reduces the size of the energy gap [15].

2.1.2 Dopants

Intrinsic semiconductors are undoped and contain both electrons and holes. The density of electrons and holes is equal because the thermal excitation of an electron from the valence band to the conduction band creates a free electron in the conduction band and a free hole in the valence band. The concentration of free electrons and holes is called the intrinsic carrier concentration, defined as the number of electrons in the conduction band or the number of holes in the valence band. The band gap and temperature of the material affect the number of carriers. Higher band gaps result in lower intrinsic carrier concentration because a large band gap makes it more difficult for a carrier to be thermally excited across the band gap [14, 15].

The process of doping involves modifying the concentration of electrons and holes within semiconductors with other atoms. Silicon (4 valence electrons) can be doped with various materials. Boron (3 valence electrons) and phosphorus (5 valence electrons) are the most common materials used. The number of valence electrons defines the doping type; here, the dopant gets into the crystal's lattice structure and affects its conductivity. The doping techniques are classified into two categories: n-type and p-type doping. N-type doping involves atoms with one more valence electron than silicon; here, the dopant will form covalent bonds with silicon's 4 valence electrons. It only needs 4 valence electrons to form covalent bonds. Therefore, the fifth electron will move around freely and act as the charge carrier. Less energy is needed to excite the free electron from the valence band to the conduction band. Only the negative electrons can move. For n-type doping, the dopant is positively charged due to the loss of negative charge carriers, and the dopant is known as an electron donor.

P-type doping is the opposite of n-type doping, where the dopant has one less valence electron than silicon. For p-type doping, the dopants get an extra outer electron and create a vacancy in the valence band of silicon in the form of a hole. The holes move in the opposite direction to that of electrons. In contrast to n-type, the dopant is negatively charged in p-type doping, and therefore often called acceptors [17, 14, 15].

2.1.3 Recombination

Recombination is the process by which an electron recombines with a hole, releasing its energy in the form of light or heat. This phenomenon occurs due to the electrons in the conduction band being in a metastable state, which may be considered a kind of temporary energy trap [18]. Consequently, the electron will stabilise into a lower energy level in an empty valence band state, thereby also removing a hole in the valence band.

The lifetime of a photocarrier, or minority carrier, in a semiconductor is dependent on the recombination rate, which is influenced by the concentration of minority carriers. Recombination rate is a parameters in solar cell which tells at what rate recombination occurs. The minority carrier lifetime of a material is the period during which a minority carrier (electron or hole) remains in an excited state after the generation of an electron-hole pair, before undergoing recombination. [14]

In order to gain an understanding of the fundamental operating principle of solar cells, it is sufficient to investigate the P-N junction, which serves to separate the electron and hole carriers in a solar cell in order to create a voltage. A P-N junction is a combination of two types of semiconductor materials, namely an n-type material and a p-type material, as shown in Figure 2.3. The excess electrons from the n-type side diffuse to the p-type side, while the excess holes from the p-type side diffuse to the n-type side. This movement of electrons results in the exposure of positive ion cores in the n-type material, while the movement of holes results in the exposure of negative ion cores in the p-type material. These movements result in the formation of an electric field at the junction, which gives rise to the depletion region. The depletion region functions as a barrier, preventing the further flow of electrons from the n-type side to the p-type side. [14]

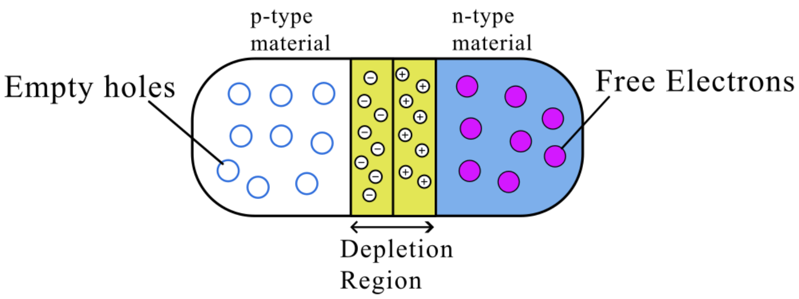


Figure 2.3: Illustrations of a P-N junction. Inspired by [2].

2.1.4 Crystalline Silicon Solar Cells

The process of crystallization is defined as the formation of solids and the subsequent organization of atoms or molecules into a well defined crystalline structure [19]. Silicon, in its crystalline form, can be divided into two categories: polycrystalline silicon (polysilicon), which is composed of small crystals, and mono-crystalline silicon (monosilicon), which is a continuous crystal. Mono-crystalline silicon (often called single-crystal silicon) stands out as the most widely used silicon material in the solar industry. It has continuous crystal lattice, which make it easy for electrons to move around. Silicon crystals are manufactured by gradually pulling a rod upward from a pool of molten silicon, a process called Czochralski. With precise control, crystallisation occurs at the end of the rod as it emerges, forming a cylindrical crystal. This cylindrical crystal is later sliced into thin pieces for incorporation into solar cells. Prior to crystallisation, elements are added to the molten silicon to impart n-type or p-type characteristics to the silicon. [20]

Dislocation, a common extended defect in crystalline silicon solar cells, disrupts the recombination properties by creating deep-level defect states within the silicon bandgap. This leads to a decrease in the lifetime of minority carriers, thereby impairing the solar cell's performance. Dislocations frequently occur during the silicon crystal growth process, underscoring the importance of controlling relevant parameters during growth to mitigate their impact. [21]

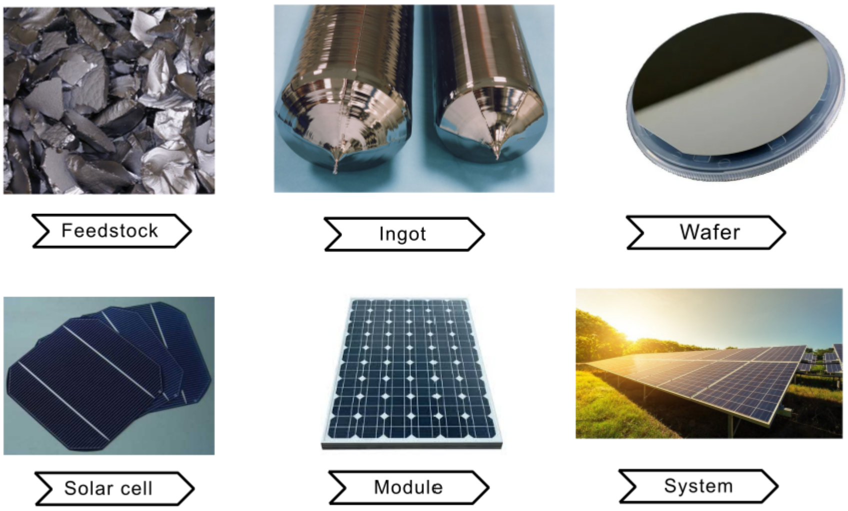


Figure 2.4: Flowchart of the silicon value chain. Inspired by [3]

The material used for manufacturing silicon is silicon dioxide (SiO_2), which is reduced in electric furnaces using charcoal to produce metallurgical silicon with approximately 98% purity. Mono-crystalline solar cells are created by melting poly-crystalline silicon in a crucible. From this molten silicon, a cylindrical mono-crystal, known as a silicon ingot, is pulled in a process known as the Czochralski process. This thesis will focus on the Czochralski process. After the silicon ingots are produced, they are sliced into wafers about 0.15 to 0.3 mm thick using a wire saw [22]. These wafers are then doped to create an electric field and transformed into cells by adding metal contacts.

Subsequently, the cells are assembled into modules, which are integrated into the final solar panel systems, including all additional electric components such as wiring and inverters as shown in Figure 2.4. Each step of the value chain leads to different industries, each dependent on the others [23]

2.2 Czochralski Process

The Czochralski (Cz) crystal pulling process is the dominant method for producing mono-crystalline silicon ingots for solar cells. In 1916, Polish scientist Jan Czochralski developed a technique for growing crystals by inserting a crystal seed into a melt in a crucible. The seed is pulled upwards, creating a single crystal [24]. Since then, the technique has been modified. The pull-from-melt method, widely used for high-efficiency solar cells today, was modified in 1950 by Teal, Little, and Dash. It contributes to almost 90% of the worldwide production of silicon mono-crystalline [25, 26].

In figure 2.5 a schematic setup of a Cz crystal puller can be seen, with the most important components named. Heat distribution is a crucial step to ensure proper temperature during the process of the growth of crystals. The crystal puller contains two crucibles, one made of graphite and the other of silica. The high temperature makes the silica crucible soften. Therefore, a crucible of graphite is attached around it for mechanical support. The graphite crucible also has a role in the heat distribution. For heaters, graphite material is used and connected to two or four electrodes at the bottom edge, which deliver power between 10 kW and 100 kW. The carbon heater has vertical slits cut into it to ensure the electric current flows up and down. The heat shield is used to lower the power consumption and also distribute the heat around the crucible more efficiently. There is a critical risk that the crucible can break, and the molten silicon can be made through the water-cooled chamber. Therefore, a spill tray has been equipped to collect the molten silicon before it damages other components like the crucible support, which rotates and lifts the crucible [12].

Stacking the crucible with poly-crystalline silicon is the initial step in the growth process. The poly-crystalline silicon is typically stored in clean double bags to prevent contamination within the crucible. Argon is used as an inert gas to create a protective environment due to its purity, which helps prevent the introduction of impurities. The constant flow of argon is essential for warding off carbon monoxide and silicon oxide, which can adversely affect the crystal structure [27]. As the temperature increases, the silicon feedstock undergoes thermal expansion. It is crucial not to stack large blocks in the crucible, as this may cause the crucible to break. However, the same crucible can be used for multiple runs, it will gradually wear out and must eventually be replaced. To ensure complete melting of the silicon, the temperature is maintained slightly above the melting point of silicon, which is 1414° C. [12, 28].

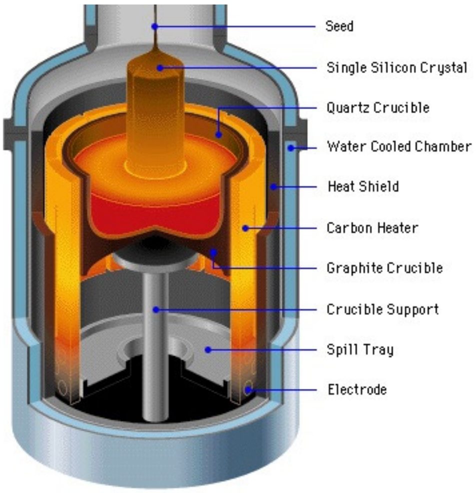


Figure 2.5: Schematic setup of a Czochralski crystal puller. With permission from [4].

Once the silicon has been completely melted and the temperature has been stabilised, it is essential to ensure that the crystal seed merges with the molten silicon in order to initiate the crystallisation process. The mono-crystalline silicon seed is immersed in the molten silicon, initiating the crystallisation process. In order to maintain uniform and cylindrical crystal growth, the seed and the crucible rotate in opposite directions. This allows the pulling process to begin. Figure 2.6 illustrates the steps in the Czochralski process, from the melting of pure silicon to the formation of a fully grown ingot.

To avoid crystal dislocations, which often occur due to thermal stress when the crystal seed comes into contact with the melt, Dash improved the technique in the 1950s by adding a necking step. Dislocations are excluded from the material when the crystal is grown fast and thin [27]. When the seed is in contact with the melt surface, a portion of the seed melts and forms a meniscus. Then, the seed is pulled upwards, and crystallisation at the end of the seed occurs. The first step of the crystallisation is the growth of the neck, where the diameter varies from 2mm to 4mm, and the length is about 30mm. During necking, the seed speed is high, with a growth rate up to 6mm/min, to ensure that the crystal is dislocation-free [12, 26, 29].

Afterward, the seed speed is reduced, and the diameter begins to increase. Here, the crown and shoulder parts are formed, transitioning from the neck to the full ingot body. During the crown process, maintaining the correct growth rate and temperature is crucial. A slow growth rate results in an excessively

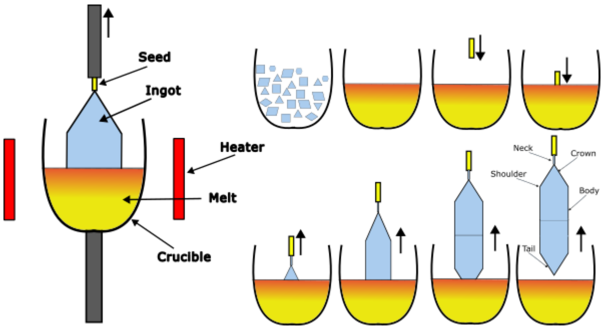


Figure 2.6: The principle of the Czochralski method and illustration of the different steps

long process time, making the melt warmer and less stable. A fast growth rate may cause structural loss in the crystal. If the temperature is too high, the diameter becomes too small, and if too low, the diameter becomes too large. Variations in the interface shape during the transition from crown to body can lead to structural loss. Therefore, a slow increase in diameter at the crown is important for a smooth transition from crown to body. When the desired diameter is reached, the pulling speed increases rapidly, forming the shoulders and then the body. The Czochralski process continues with a constant seed speed, forming the body as a cylinder with a constant diameter. [12, 29, 26]

When the body reaches the desired length, the pulling speed and temperature increases to form the so-called tail at the end of the body. During this phase, the diameter gradually decreases until the crystal separates from the melt. If the diameter of the crystal is not properly controlled during the tail formation, it can lead to crystal dislocations. These dislocations in the tail can adversely affect the quality of the wafers produced during the body stage. [12] [30].

2.2.1 Remelting of Cz Silicon

If structure loss is detected by the operator, the silicon ingot will be remelted. When structure loss occurs in the early stages of growth, the grown part is remelted. However, if it occurs at a later stage, growth is stopped, and the failed part is cut off [13]. Figure 2.7 illustrates the remelting process: 1) the seed is pulled upwards from the molten silicon, 2) the pulling process continues, 3) structure loss is detected and the ingot is remelted, 4) the ingot is melted up to the neck, allowing the pulling process to restart.

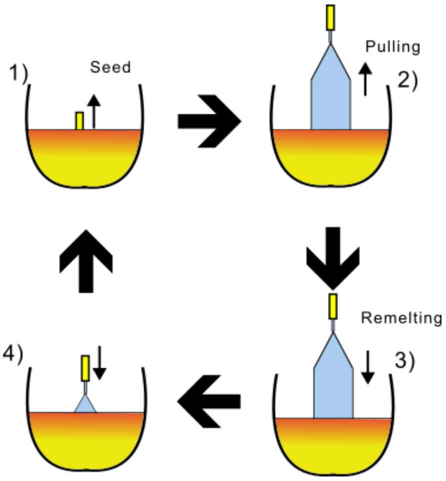


Figure 2.7: Description of the remelting process.

2.2.2 Impurities and Defects in Cz Silicon

Impurities are substances that affect another substance and its properties; they can be added intentionally or unintentionally. In Czochralski silicon, multiple impurities can arise. The most common unintentional impurities are oxygen and carbon, while intentional impurities are dopant elements like boron and phosphorus. Defects in Czochralski silicon can lead to structural loss in the crystal, resulting in the transition from mono-crystalline to multi-crystalline [31, 32].

Hoshikawa and Huang describe in [33] that there are four stages of oxygen transportation in the Czochralski silicon crystal growth: (1) oxygen dissolution from the quartz crucible, (2) oxygen incorporation within the silicon melt, (3) evaporation from the melt as SiO, and (4) oxygen incorporation into the crystal. Several factors play a role in oxygen transportation, including thermal stress, doping, and argon pressure.

Oxygen is incorporated into the crystal during the Czochralski crystal growth process. It often enters the melt from the dissolving quartz crucible wall, where it either integrates into the crystal or evaporates from the melt as SiO. If oxygen enters the melt, the concentration will be lower than the solubility due to the percentage that evaporates as SiO. Studying the transport of impurities is important to identify regions where impurity deposition can cause difficulties and to avoid undue deposition above the melt surface. According to [34], only 1% of the SiO transported into the melt is incorporated into the crystal. This indicates that a large amount of SiO evaporates into the furnace's environment.

During dissolution from the silica crucible, the silicon melt dissolves SiO_2 and absorbs its oxygen. Factors such as temperature and material density define the dissolution rate. Due to the flow pattern of thermal convection, a fraction of the oxygen will evaporate from the silicon melt. Temperature and atmospheric pressure are decisive factors for the oxygen evaporation rate. According to Hoshikawa and Huang [33], argon pressure affects the oxygen evaporation rate and oxygen transportation in an experimental sample. However, it is mentioned that argon pressure will not significantly affect oxygen transportation in practical Czochralski silicon crystal growth due to the limited range in which argon pressure can be adjusted.

Various studies have detected a relationship between oxygen content and the available melt surface. When the crystal diameter is small ($< 125\,\mathrm{mm}$), oxygen evaporation is very sensitive to changes in diameter. Once the crystal reaches its desirable diameter for the body process, oxygen evaporation remains nearly constant. This indicates that oxygen content increases during the crown process of crystal growth because the melt area is large, causing a significant amount of oxygen to evaporate. However, when the diameter is large, it covers a substantial melt area in the crucible, resulting in more oxygen being incorporated into the silicon ingot. [35].

2.3 Supervised Learning

Machine learning algorithms can be trained using various methods and are typically divided into two main types: supervised and unsupervised. This thesis concentrates on the supervised learning method of machine learning algorithms. Supervised learning models the relationship between known inputs, referred to as features x_i, and their corresponding outputs, commonly termed targets y_i. In contrast, unsupervised learning analyzes data based solely on input data without predefined labels.

In Figure 2.8, classification and regression are depicted as the two principal types of supervised machine learning. Classification involves algorithms that learn from data to predict categorical outcomes. The classification algorithms used in this thesis include logistic regression, random forest classifiers, and neural networks, which will be discussed in detail in the following sections. These algorithms often predict binary outcomes, such as ”True” or ”False,” a process known as binary classification. In this thesis, the binary outcomes

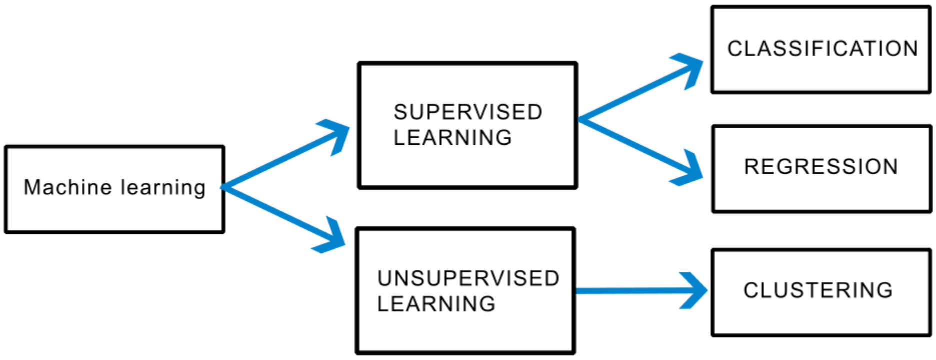


Figure 2.8: Key differences between supervised and unsupervised learning. Inspired by [5].

will be ”non-structure loss” or ”structure loss.” However, if a classification algorithm predicts more than two outcomes, it is known as multiclass classification. Conversely, regression analyzes data to predict continuous outcomes and is commonly used to forecast variables such as sales figures, salary amounts, body weight, or temperature readings. However, regression is not employed in the analysis presented in this thesis.[36]

2.4 Logistic Regression

Binary logistic regression is a type of regression analysis where the dependent variable is binary, meaning it can take on one of two possible values (e.g., 0 or 1, yes or no). The purpose of logistic regression is to model the relationship between one or more independent variables and a binary outcome. It achieves this by estimating the probability that a given input point belongs to a certain category using the sigmoid function (also called the logistic function), which maps predicted values to probabilities between 0 and 1. [37]

2.4.1 Sigmoid Function

Logistic regression has been one of the most preferred models for binary classification because of its ease of implementing machine learning methods; it works well with linearly separable datasets and provides valuable insights by showing how relevant a predictor variable is. By letting X_i denote the predictor and y be the predicted output, the response can be modeled by the linear regression function:

$$E\{y \mid x\} = X\beta \tag{2.1}$$

where the β are the model coefficients of $(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n)$. Predictions and their probability are mapped using a sigmoid function, which involves an S-shaped curve that transforms any real number into a range between 0 and 1, as seen in figure 2.9. The sigmoid function for logistic regression is defined below;

$$f(x) = \frac{1}{1+e^{-x}} \tag{2.2}$$

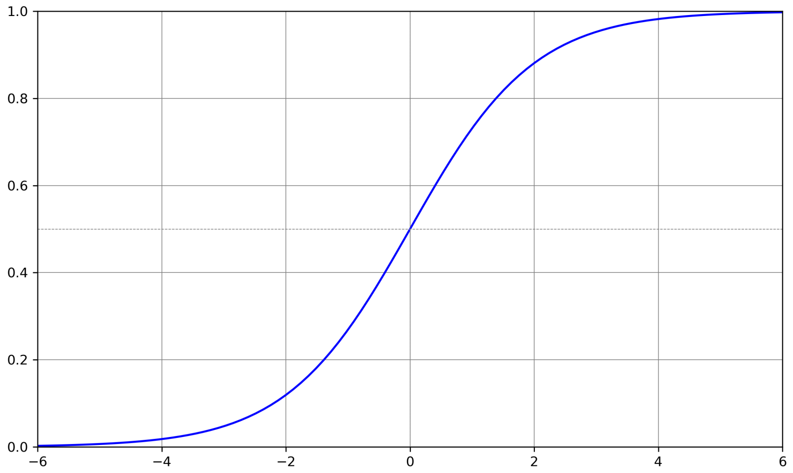


Figure 2.9: Sigmoid function graph.

The logistic or sigmoid function graph has the critical property of limiting the output values between 0 and 1, regardless of the input x value. This property guarantees that the output, which indicates the probability of a particular observation belonging to a specific class, stays within a reasonable range for probabilities. Therefore, by combining equations 2.1 and 2.2, the probability that $y = 1$ given X can be defined as logistic function:

$$P(Y = 1 \mid X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+...+\beta_n X_n)}} \tag{2.3}$$

In logistic regression, the decision rule is based on the output of the Sigmoid function. If the output is greater than 0.5, the observation is classified as class 1 (positive); if it is less than 0.5, it is assigned to the negative class (0). This decision boundary at 0.5 simplifies the binary classification task by dividing the probability space into two distinct regions corresponding to the two classes.

The logistic regression model uses the logistic function to model this probability. However, instead of modeling the probability directly, logistic regression models the log-odds (logit) of the probability. The log-odds is the natural logarithm of the odds of the event occurring, where the odds are defined as the probability of the event occurring divided by the probability of the event not occurring. The log-odds are modeled as a linear combination of the independent variables (predictors). This means that the relationship between the independent variables and the log-odds is linear.

The linearity assumption is crucial because it simplifies the estimation of the model parameters and the interpretation of the coefficients. Each coefficient β_i represents the change in the log-odds of the outcome for a one-unit change in the corresponding predictor X_i, holding all other predictors constant. If the linearity assumption holds, the relationship between each predictor and the log-odds is straightforward and additive. This allows for a clear and interpretable model. However, if the relationship is not linear, the model may not fit the data well, leading to biased estimates and incorrect inferences. [38]

2.4.2 Parameter estimation

In binary logistic regression, the model parameters are estimated using a method called Maximum Likelihood Estimation (MLE). MLE is a statistical technique that identifies parameter values that make the observed data most probable. MLE determines the values that maximise the likelihood function, which measures the probability of observing the given set of data under various parameter values.

Let y_i denote the observed outcome for the i-th observation and $\hat{p}_i$ the predicted probability that $Y_i = 1$. The likelihood function $L(\beta)$ for a dataset of n observations is given by:

$$L(\beta) = \prod_{i=1}^{n} P(Y_i = y_i \mid X_i) \tag{2.4}$$

Given that y_i can be either 0 or 1, this expression can be rewritten as:

$$L(\beta) = \prod_{i=1}^{n} \hat{p}_i^{y_i} (1 - \hat{p}_i)^{1-y_i} \tag{2.5}$$

For computational simplicity, the log-likelihood function, which is the natural logarithm of the likelihood function, is often used:

$$\ell(\beta) = \sum_{i=1}^{n} [y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i)] \tag{2.6}$$

Maximising the log-likelihood function, $\ell(\beta)$, is mathematically equivalent to maximising the likelihood function $L(\beta)$, but computationally more straightforward due to the properties of logarithms. Through iterative adjustments, the parameter estimates are refined to align the predicted probabilities $\hat{p}_i$ as closely as possible with the observed outcomes y_i. This convergence process identifies the parameter values that best fit the data, ensuring the model's predictions are as accurate as possible within the logistic regression framework. The final parameter estimates thus maximise the likelihood of observing the outcomes, making them the most probable given the independent variables. [39]

2.4.3 Scikit-learn

Scikit-learn is an open-source machine learning library that features various classification, regression, and clustering algorithms, including logistic regression. It is built on top of NumPy, SciPy, and matplotlib, and is designed to interoperate with other Python libraries. Scikit-learn's logistic regression implementation is robust, efficient, and easy to use, providing a convenient interface for performing logistic regression. Further details about Scikit-learn's implementation of logistic regression, including specifics about package details and parameters, can be found in [40].

2.5 Random forest

Random Forest is a versatile and powerful machine learning algorithm extensively used for both classification and regression tasks. It is part of the ensemble learning methods family, which improves accuracy and robustness by combining the predictions of multiple models. Developed by Leo Breiman, random forest builds on the concept of decision trees. It creates a 'forest' of numerous trees and aggregates their results, details of which will be elaborated below [41].

2.5.1 Decision tree

Decision trees are the building blocks of the random forest algorithm, which has a hierarchical tree structure consisting of root nodes, branches, internal nodes, and leaf nodes. They provide a good representation of decision-making and potential outcomes [42]. A single decision tree is the fundamental building block of a Random Forest. The construction of a decision tree involves recursively splitting the dataset into subsets based on the values of the input features. To select the best split at each node of the tree, the algorithm evaluates all possible splits across all features to find the one that best separates the data. The 'best' split is usually determined by a metric such as Gini impurity, which measures the impurity of a node. Lower values indicate a more homogenous node and are calculated in Equation (2.7).

$$Gini(D) = 1 - \sum_{i=1}^{c} p_i^2 \tag{2.7}$$

where p_i is the probability of class i in the dataset D.

Once the best split is found, the dataset is divided into two subsets, and the process is repeated recursively for each subset. This creates a hierarchical structure of nodes and leaves. The recursion continues until a stopping criterion is met, such as reaching a maximum tree depth, having a minimum number of samples at a node, or if no further information gain can be achieved. When the stopping criteria are met, a leaf node is created. In classification, each leaf node represents a class label, typically chosen by majority voting among the samples in that leaf. In regression, the leaf node contains a numerical value, typically the mean of the target values of the samples.[43]

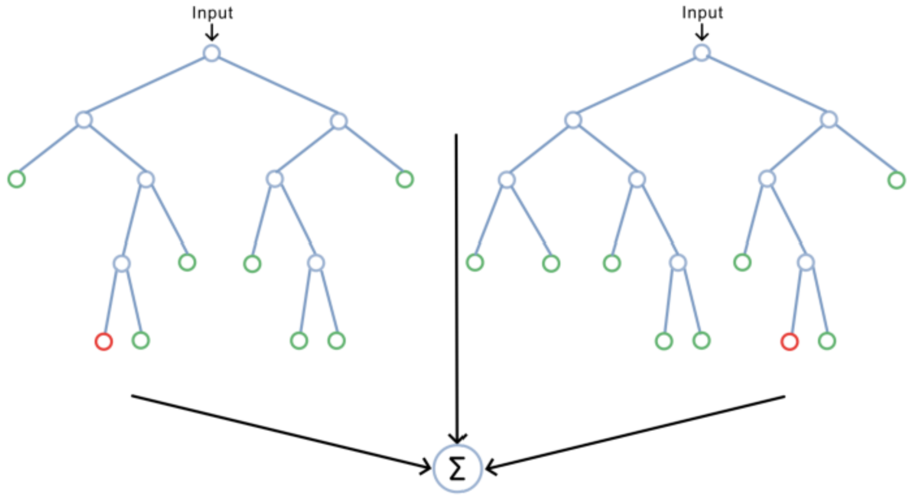


Figure 2.10: Random forest model merged together by two decision trees. Inspired by [6].

2.5.2 Ensemble learning

Ensemble learning is a powerful technique within machine learning, combining the predictions of multiple individual models to create a more robust, more accurate prediction. The basic concept involves merging the outcomes of multiple models, for which the weaknesses of individual models can be compensated, resulting in improved overall performance. The underlying principle is that a diverse set of models, each prone to different errors, can collectively achieve better results than any individual model on its own. Figure 2.10 shows two decision trees merged together to form a random forest. [44]

Bootstrap aggregating (also known as bagging) is an ensemble learning technique that improves the accuracy of any machine learning models, especially for predictions. It creates multiple versions of a predictor and combines them to form an aggregated predictor. This process involves averaging the outcomes for numerical predictions or using a majority vote system for class predictions. The generation of multiple predictor versions is accomplished by creating bootstrap replicates of the original dataset. These replicates serve as new learning sets for training different versions of the model.

Using methods such as classification and regression trees and subset selection in linear regression, bootstrap aggregating has been shown to significantly improve model accuracy in both real and simulated data sets. The effectiveness of bootstrap aggregating lies in the instability of the prediction method. If small changes in the dataset can cause significant variations in the constructed predictor, then bootstrap aggregating can enhance accuracy. The reduction in overfitting and variance that is achieved through the use of bootstrap aggregation results in the generation of a more robust model. [45]

In addition to bootstrap aggregating, random forest introduces an additional layer of randomness to further reduce correlation between trees and enhance model performance. This is achieved by selecting a random subset of features for each split in the decision trees. At each node, instead of considering all features to determine the best split, a random subset of features is chosen. The size of this subset is controlled by the hyperparameter $max_features$. The best split is then determined only from this subset, rather than the full set. This random selection of features aids in decorrelating the trees, as different trees are likely to consider different features for splits, leading to a diverse ensemble of trees. This diversity is crucial for the ensemble's effectiveness. [6]

2.5.3 Averaging and Voting

The final prediction of a Random Forest model is derived by averaging the predictions of all individual decision trees within the forest. In classification tasks, each tree outputs a class prediction, and the class receiving the majority of votes across all trees is selected as the final prediction. This majority voting mechanism enhances the model's accuracy by capitalizing on the collective decisions of multiple trees, thereby reducing potential errors from any single tree. For example, if three trees predict class 0 and two trees predict class 1, the final prediction will be class 0, as it represents the majority. [46]

2.5.4 Scikit-learn

The Scikit-learn library can be used to implement the random forest algorithm, where the same library is used for logistic regression as explain in subsection 2.4.3.

Scikit-learn's implementation of random forest is both robust and user-friendly, allowing for easy application and experimentation. Further details about Scikit-learn's implementation of random forest, including specifics about package details and parameters, can be found in [47].

2.6 Neural network

Neural networks are widely used in modern machine learning and artificial intelligence, inspired by the structure and function of the human brain. They are designed to recognize patterns, make decisions, and solve complex problems across various domains such as image recognition, natural language processing, and predictive outcomes. A neural network consists of interconnected layers of nodes, or neurons, where each connection represents a weighted link between neurons. The basic building block of a neural network is the perceptron, which simulates a biological neuron by receiving input, processing it, and generating an output. These neurons are organized into layers: the input layer, one or more hidden layers, and the output layer [48]. All these concept will be explained in more detail below.

2.6.1 Artificial neuron

Artificial neurons (often called perceptrons or nodes) are the building blocks in a neural network, inspired by the biological neuron cells in the human brain. Each artificial neuron functions as a connection point in an artificial neural network, capable of taking input from other neurons and forwarding output to other neurons in the network. The similarity between artificial and biological neural networks lies in the connections between the neurons, which are defined by synaptic weights. These weights signify the importance of each connection. Learning occurs when new information is fed into the network, causing the synaptic weights to change. [48]

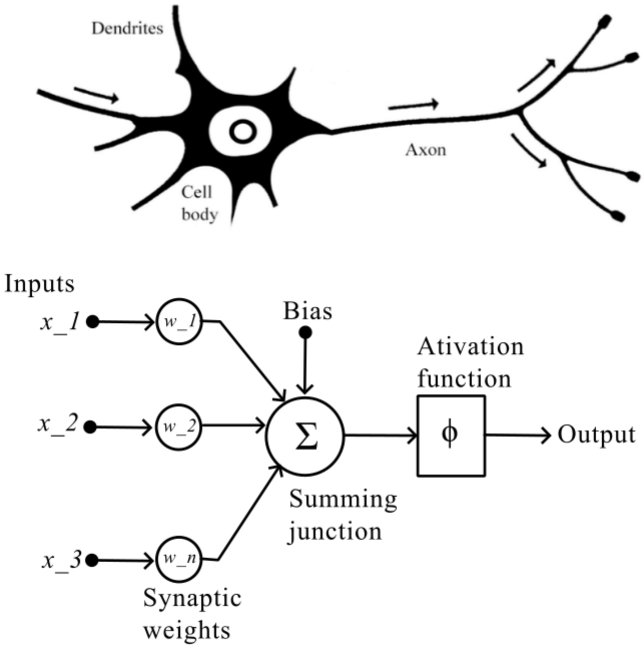


Figure 2.11: Illustration of the structural and functional similarities between biological and artificial neurons. Inspired by [7].

Figure 2.11 compares biological and artificial neurons. A biological neuron consists of dendrites, a cell body, and an axon. The biological neuron receives input through dendrites, which transmit it to the cell body. The output is generated in the cell body and transmitted down the axon, which then sends it to neighbouring neurons via the synapse connection. The artificial neural network operates on the same principle as a biological neural network, where output is generated and transmitted to adjacent neurons.

An artificial neural network (ANN) is built up by layers of artificial neurons, typically consisting of an input layer x, one or more hidden layers (s^1, s^2, s^n), and an output layer $f(x;\theta)$. The artificial neuron will often receive input from multiple inputs, as seen in figure 2.11. The input x_m is a set of weighted input, described with equation 2.8;

$$(w_1x_1 + w_2x_2 + ... + w_nx_n) \tag{2.8}$$

Each artificial neuron has its own specific weight and threshold. Whether the data is passed to the next layer is determined by whether the neuron's output exceeds the threshold. Therefore, by using equation 2.8 we get a linear model;

$$f(x;\theta) = (w_1x_1 + w_2x_2 + ... + w_nx_n) + b \tag{2.9}$$

Here, the weighted input w_n gets added with a bias constant b to form the summing junction; the bias has its own weight with each connected neuron. The result from the summing junction is then passed through an activation function that produces an output. The activation function is a transfer function that determines the output the neuron should produce based on the input. It also introduces nonlinearity, which enables the network to learn complex patterns in data. [7]

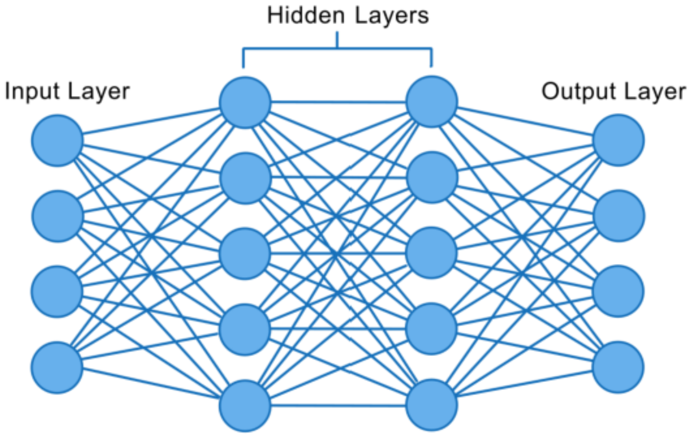


Figure 2.12: Illustration of a neural network with input layer, hidden layers and output layer. Inspired from [8].

Hidden layers in an artificial neural network are layers that provide information between the input and output layers, as seen in figure 2.12. Depending on the problem, multiple hidden layers can be present. If there are not enough hidden layers, the network will fail to learn. Conversely, if there are too many hidden layers, the network will have the risk of overfitting. The connection between layers when passing data from one layer to the next layer defines the network as a feedforward network.

2.6.2 Multilayer perceptrons

A Multilayer Perceptron (MLP) is a feedforward neural network consisting of neurons controlled by a nonlinear activation function. An artificial neural network can only handle linear functions and has a limited ability to handle complex problems. With MLP, it can be used to handle non-linear functions and process complex and large amounts of data.

MLP is commonly used in image recognition, natural language processing, and speech recognition. MPL is known for its flexibility in structure and its ability to recognize functions under certain conditions. This makes MPL an essential building block in deep learning. Multilayer perceptrons have the same key components as a neural network, such as an input layer, hidden layer, output layer, weights, a bias neuron, and an activation function. [49] By introducing the equation 2.9, a general neuron processing unit can be described as;

$$a = \phi(\sum_n w_n x_n + b) \tag{2.10}$$

where the ϕ is the non-linear activation function, and a is the unit's activation. To describe the MLP computations mathematically, various units of the network need to be considered for all three key layers: input units, hidden units, and output units;

$$\begin{aligned} h_i^{(1)} &= \phi^{(1)}(\sum_n w_{in}^{(1)} x_n + b_n^{(1)}) \\ h_i^{(2)} &= \phi^{(2)}(\sum_n w_{in}^{(2)} h_n^{(1)} + b_n^{(2)}) \\ y_i &= \phi^{(3)}(\sum_n w_{in}^{(3)} h_n^{(2)} + b_n^{(3)}) \end{aligned} \tag{2.11}$$

where as before the input unit as x_n, the activation of the output unit as y, the hidden layer is expressed as $h_i^{(n)}$ and $\phi^{(n)}$ is defined as the activation functions. The three equations in 2.11 show how the network is connected with the units in the previous layer, and where each layer has its own activation function [50].

2.6.3 Activation functions

Activation functions are an important component in MLP neural networks. As mentioned above, the activation function determines the output that will be passed on to the next neural layer. The activation function processes the input signal to generate an output, which is then further processed in the network. Without an activation function, the model would only be a simple linear regression function of the input. This is why MLPs can handle large amounts of data and learn complex patterns. The most common activation function for binary classification is the Sigmoid activation function.

The Sigmoid function is widely used due to its combination of both non-linear and differentiable. It is crucial for the hidden layers to have a non-linear activation function. Without the non-linearity in the hidden layers, the neural network would only have linear transformation between layers. In such a case, the MLP network could be replaced by an equivalent single-layer network. The differentiable property is required by the backpropagation algorithm. [51]

The Sigmoid function has some disadvantages. Firstly, the network can suffer from the vanishing gradient problem, which occurs when the input to the Sigmoid function is close to 0 or 1. In this case, the gradient is close to zero, leading to an unstable training process for the network. Secondly, the Sigmoid function gets saturated for large positive and negative values. This means that the gradient becomes close to zero for these values. Consequently, when the output is very close to 0 or 1, the weight updates during training become very small, resulting in a slower learning process for the model.[52]

2.6.4 Training the multiplayer perceptrons network

Training the MLP network is a complex process performed in a supervised machine learning context where each feature has a label for classification. The network learns patterns and relationships in the input data to predict the class to which the feature belongs. The backpropagation algorithm is used for this purpose.

The backpropagation algorithm is based on iteratively adjusting the weights within a neural network to minimise the difference between predicted outcomes and actual observations. This process begins by computing the error at the output layer of the network. The error is then propagated in reverse through the network, layer by layer, from the output layer to the input layer. The goal is to update the weight parameters for all layers by using the error gradient with respect to each weight. This iterative process aims to systematically reduce the overall error of the neural network by refining the weight values from the final layer back to the first, thereby optimising the network's predictive accuracy.

2.6.5 Scikit-learn

The Scikit-learn library also provides an easy to use implementation for building and training a neural network for classifications tasks, called MLPClassifier. MLPClassifier are based upon the multilayer perceptron (MLP) structure in a feedforward artificial neural network (ANN), consist of at least three layers; an input layer, one or more hidden layers, and an outputlayer. Further details about the Scikit-learn's implementation of MLPClassifer, including specifies about pachage details and parameters, can be found in [53].

2.7 Hyperparameters tuning

In machine learning development, each dataset and model requires a unique set of hyperparameters, which are critical variables that significantly impact model performance. Hyperparameter tuning is the process of experimentally testing various combinations of these parameters to refine the model's accuracy and effectiveness. This process is conducted before training begins to ensure optimal settings. Each machine learning algorithm requires specific hyperparameter adjustments tailored to the model's objectives; for example, the number of trees in random forests, the learning rate in gradient boosting machines, and the number of hidden layers in neural networks. The two most commonly used techniques for hyperparameter tuning are grid search and random search.

Grid search is a systematic approach to hyperparameter optimization that involves defining a set of possible values for each hyperparameter of interest. The algorithm then trains models using every possible combination of these values. Each model configuration is evaluated against a predetermined metric to assess performance. This exhaustive method ensures that all possible combinations are explored, enabling the identification of the hyperparameter set that yields the best performance according to the selected evaluation metric.

One significant drawback of grid search is its tendency to become computationally intensive and time-consuming, especially when determining the optimal combination of hyperparameters. The method's exhaustive nature involves evaluating every possible hyperparameter combination, which can lead to the generation of numerous model variations.

Random search differs from the grid search approach by using a distinct strategy for hyperparameter optimization. Instead of specifying a list of discrete hyperparameter values for examination, random search requires defining statistical distributions for each hyperparameter of interest. The optimization algorithm then samples values from these specified distributions. This method introduces randomness into the selection process, which can lead to a broader and potentially more innovative exploration of the hyperparameter space. It is particularly effective when dealing with a high-dimensional space or when the optimal hyperparameter values are unknown. Focusing on randomly selected points rather than an exhaustive combinatorial search provides a more efficient search process. [54]

Random search is often more efficient and effective than grid search because only a few hyperparameters significantly impact a model's performance on a given dataset. Sampling from predefined statistical distributions allows random search to explore a wider range of values for these critical hyperparameters more quickly and broadly than grid search. [55]

2.8 Cross validation

Cross-validation is a statistical technique used in machine learning to evaluate the performance and generalizability of a model. It helps to ensure that the model's predictions are reliable and not overly fitted to the training data. Figure 2.13 shows how the entire dataset is divided into 5 subsets or ”folds”. The model is then trained on k-1 folds, and tested on remaining fold. This cycle is repeated k times, each iteration using a different subset for test set. Upon completing these iterations, the outcomes from each validation phase are combined to calculate an average performance metric. This averaged result provides a more reliable estimate of the model's predictive accuracy. Cross validation is essential in the machine learning framework because it helps verify that the final model is both robust and able to generalize to unseen data effectively. [56]

2.9 Model evaluation

Model evaluation involves using various metrics to assess the strengths and weaknesses of a machine learning model. The most commonly used metrics for measuring the performance of a classification model are accuracy, precision, confusion matrix, and AUC (area under the ROC curve). These metrics provide insight into the model's performance and help compare different models.

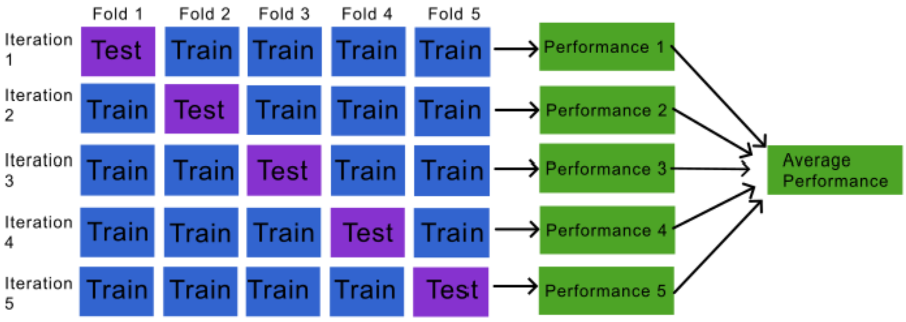


Figure 2.13: Illustration of how the cross validation technique works. Inspired by [9].

In binary classification, the outcome usually consists of a prediction. Therefore, to accurately enumerate both correct and incorrect predictions, it is necessary to clarify the various possible outcomes: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). TP refers to a positive prediction that is correctly classified, TN refers to a negative prediction that is correctly classified, FP refers to a positive prediction that is incorrectly classified, and FN refers to a negative prediction that is incorrectly classified. [57]

2.9.1 Accuracy

The accuracy score represents the proportion of correct predictions made by the model out of all predictions attempted. The accuracy score is defined as;

$$Accuracy = \frac{TP+TN}{TN+TP+FP+FN} \tag{2.12}$$

In the event that the data is imbalanced, the accuracy score may be found to be misleading. This is due to the fact that the model can be highly accurate when the majority of the input data belongs to the same class.

2.9.2 Confusion matrix

The confusion matrix is a visual tool that represents the four outcomes of a classifier: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). For binary classification, the confusion matrix in Table 2.1 is structured as a 2x2 matrix that categorizes predictions as correct or incorrect, where the x-axis represents the predicted class and the y-axis represents the actual class. A TP outcome indicates that the model correctly predicted the positive class. Conversely, an FP outcome means that the model incorrectly predicted an instance as positive. A TN outcome reflects correct predictions of the negative class, and an FN outcome signifies that the model incorrectly labeled an actual positive instance as negative. In this thesis, ingots with structural loss are labeled as the positive class (y=1), and those without structural loss are labeled as the negative class (y=0).

Table 2.1: Representation of a 2x2 confusion matrix.

Actual \ Predicted	**Negative**	**Positive**
Negative	TN	FP
Positive	FN	TP

2.9.3 Precision

Precision is defined as the proportion of true positive cases (TP) to the total number of cases identified as positive by the model, which includes both true positive and false positive cases (TP + FP), as shown in equation 2.13. It quantifies the accuracy of the model in predicting positive outcomes.

$$Precision = \frac{TP}{TP+FP} \tag{2.13}$$

2.9.4 Recall

Recall is the metric that assesses the proportion of actual positive cases accurately classified by the model. It is calculated as the ratio of true positive cases (TP) to the total actual positive cases, which is the sum of true positive cases and false negative cases (TP + FN). This measure reflects the model's ability to capture all relevant instances.

$$Recall = \frac{TP}{TP+FN} \tag{2.14}$$

2.9.5 F1-score

The F1 score is a widely used metric for evaluating the performance of binary classification models, especially in scenarios where there is an imbalance between classes. It represents the harmonic mean of precision and recall.

Achieving an F1 score of 1 signifies that the model has attained perfect precision and recall, reflecting its exceptional accuracy in correctly classifying instances.

$$F1 = 2 \cdot \frac{Precison \cdot Recall}{Pressiosn + Recall} \tag{2.15}$$

2.9.6 AUC-ROC curve

The Receiver Operator Characteristic (ROC) curve is a graphical representation used to evaluate the performance of binary classification models. It displays the true positive rate (TPR), also known as sensitivity, against the false positive rate (FPR), or 1-specificity, at different classification thresholds. This curve helps to assess the trade-off between sensitivity and specificity, allowing for the identification of an optimal balance for decision-making thresholds in the model. [58]

The Area Under the Curve (AUC) is a statistical metric that summarizes the Receiver Operator Characteristic (ROC) curve. Its values range from 0 to 1, with a score of 1 signifying flawless prediction or classification, distinguishing perfectly between positive and negative classes. Conversely, a score of 0 denotes complete misclassification, where positive cases are predicted as negative and vice versa. An AUC score of 0.5 indicates that the model's ability to classify instances accurately is equivalent to random guessing. [59]

2.9.7 Cross-Entropy loss

The cross-entropy loss, also known as the log loss, is a loss function employed in machine learning to assess the efficacy of a classification model. The cross-entropy loss function is employed to assess the discrepancy between the predicted probability distribution and the actual label. This loss function is particularly sensitive to incorrect predictions that are made with a high degree of confidence. For example, if a model predicts a high probability for a class that is incorrect, the loss function will increase significantly. The cross-entropy loss function ranges from 0 to infinity. A perfect model would have a cross-entropy loss of 0, while a model that is consistently incorrect and confident would have a very high loss. Cross-entropy loss function is given by Equation 2.16; [60]

$$\text{Log Loss} = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \tag{2.16}$$

where N is the number of observations, y_i is the actual label (0 or 1), and p_i is the predicted probability of the label being 1.

Chapter 3

Introduction to the Data and Datasets

This chapter provides a concise overview of the data used in this thesis, laying the groundwork for effective machine learning model development and analysis in the Czochralski process. Initially, we describe the sources and characteristics of the collected data, which encompass critical parameters such as ingot diameter, ingot length, seed speed, and heater power. These parameters are essential for understanding the factors influencing the quality of silicon ingots. Therefore, a section on data understanding is included to explore the initial insights gained from the data in all four regions of the ingot: neck, crown, shoulder, and body. Lastly, an overview of the preparation steps is presented, including data cleaning, handling missing values, and standardising, to ensure the data is ready for analysis

3.1 Data Overview

The data utilised in this thesis were provided by NorSun and comprise approximately 1449 files. Each dataset corresponds to a distinct Czochralski (Cz) process, with data points recorded at one-second intervals from the onset of the neck until three hours after the crown commences. Approximately 75% of the datasets correspond to ingots that have not experienced any structural loss, while 25% relate to ingots that have undergone structural loss, as indicated by the filenames. All datasets consistently include four parameters: ingot diameter, ingot length, seed speed, and heater power. Additionally, a column in each dataset indicates the stage of the Cz process—neck, crown, shoulder, or body—identified by numeric codes: neck (170), crown (180), shoulder (190), and body (200), as depicted in Figure 3.1. It is important to note that if structural loss is identified in the crown, subsequent state numbers are not recorded.

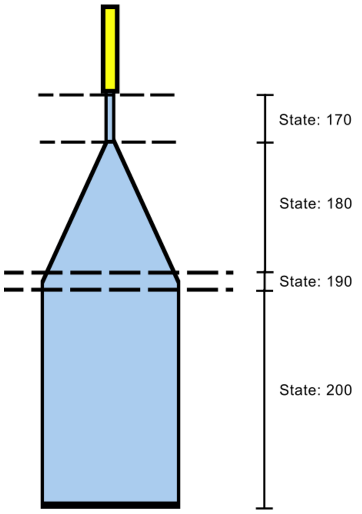


Figure 3.1: Schematic of the ingot-pulling process showing various stages identified by state numbers: neck (170), crown (180), shoulder (190), and body (200). These stages correspond to distinct phases in the dataset, facilitating targeted analysis at each phase of the Czochralski process.

A unique identifier was attached to datasets to mark instances of structural loss, as identified by operator in the ingot. Should intervention be necessary to disrupt the process, the remelting procedure would commence. For the set of datasets used in this thesis, Figure 3.2 illustrates the distribution of ingots exhibiting structural loss and the diameter interval at which it was detected. From the datasets, approximately 59% of the ingots showed structural loss in the crown, while 41% exhibited structural loss post-crown. In total, structural loss was detected in 363 ingots during the pulling process.

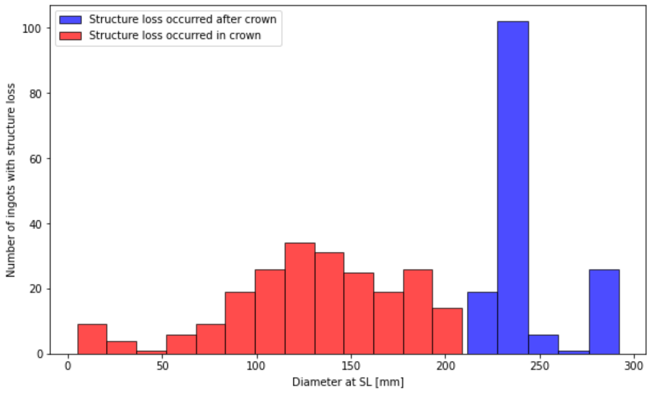


Figure 3.2: Frequencies of structure loss occurrence across ingot diameter.

Each dataset had a distinctive filename containing information about the Cz process. For instance, a filename could be `crownA4PU82_F.csv`. Here, the letter *A* denoted the run number, *A* being the first crucible; subsequent runs with the same crucible were designated as *B*, *C*, and so forth. A new ingot ID was generated when a new ”run” (i.e., a new crucible) commenced. The value ’4’ represented the run number. ’PU’ signified the puller, with the puller number being 82. NorSun provided data from eight distinct pullers with ID numbers (PU80-PU88). To denote if structural loss had occurred during the pulling process, the filename was suffixed with ’`_F`’; datasets without structural loss did not include this notation.

3.2 Data Understanding

In order to comprehend the functioning of the data delivered by Norsun, it is necessary to provide a detailed explanation of each step of the process.

Stabilization: The operator must adjust the heat power to control the melt temperature. Then, the operator dips the seed into the melt and waits for about 30 minutes until a meniscus is observed. The operator will then characterize the meniscus as too cold or hot. If the meniscus is too cold, the temperature will be increased by 1-2 degrees (units), and wait for about 20-30 minutes. When the meniscus has the correct temperature, the neck will form. Here, the diameter will be about 12mm.

Pre-neck: Seed speed will be set to a fixed value (e.g. 2mm/min) for a specific period (e.g. 10min). Here, the goal is to reduce the diameter and estimate whether the temperature is too cold or hot. If it is too hot, the diameter will descend more than expected. If the diameter drops less than expected, it is a sign that the melt is too cold. After the pre-neck, the process will go over to the neck; here, the temperature setpoint will be changed according to the diameter changes done in the pre-neck. In cases where the diameter descends more than expected, the temperature setpoint will decrease.

Neck: The crystal length data will be reset to zero, and the automatic diameter control (ADC1) will be turned on. ADC1 tries to keep a setpoint diameter by adjusting the seed speed and uses observation from the diameter to follow a desired target diameter trajectory. Now, the ADC2, which regulates the temperature in the outer loop in the crucible, is turned on. If the seed speed is too high, it indicates that the melt is too cold, and therefore the setpoint for temperature increases. When the diameter descends below the lower limit, the accumulated limit (AL) starts and must reach a given length (e.g. 100mm). As long as the diameter is within the limit, it is the approved length. When the diameter exceeds a limit, the integration stops. Then, the diameter must descend to the lower limit to start telling again.

A good neck grows fast and thin because dislocation has narrow mobility. Then, the dislocation will grow out from the side of the neck and, with time, be excluded from the material. It's good to have a high seed speed, but also a risk for ”pop-out”; the neck is pulled out of the melt. When the neck becomes very thin, increasing the diameter takes a long time. This is because the heat generated when silicon freezes must be pushed up along a thin neck. Usually, the neck is longer than what is needed to be sure that it is dislocation-free. A long neck can also act as a temperature sensor because the system responds to heater temperature changes relatively slowly. The temperature at which the crown phase could begin should be accurate to within one degree [12]. When the AL has reached a desired length, it will automatically go over to the crown.

Crown: Again, the crystal length will be reset to zero. At this stage, the goal is to increase the diameter from a dislocation-free neck to a defined diameter close to the body diameter. This happens by following a table, which means all parameters will be the same for all crowns unless the operator adjusts the process except the diameter, which results from everything else. Too slow growth rate decreases the length and increases the probability of structure loss due to the melt being warmer and less stable than ideal. A growth rate that is too high can also result in structure loss caused by the melt being too cold. A large diameter in the crown means that the melt must be cooler. This is why there is a steady decrease in the heater power during the crown process [12].

For the crown, the diameter is used as a trigger for ending the crown; this implies that the length of the crown will also vary. When the diameter has reached the trigger, it will automatically go over to the shoulder.

Shoulder: The crystal length will again be reset to zero. In the shoulder process, the goal is to reach the body diameter, which means it has to be reduced by a small amount. This is done by increasing the seed speed, but it is also essential to have a low level of seed speed due to the risk of the diameter growing inwards during body growth. The shoulder also has a diameter trigger before processing over to the body.

Body: Crystal length is reset to zero, and now the diameter control ADC1 and temperature control ADC2 are used to get the desired growth rate. The ADC1 controls the diameter through a fast seed speed. A temperature drop is usually needed in the transition from shoulder to body.

3.3 Labeling

Correct labeling is a crucial concept in machine learning to ensure that the classifier assigns the correct class during model training. In binary classification, there are only two labeling classes. For this thesis, ingots without structural loss are labeled as ’0’, and those with structural loss are labeled as ’1’. Each dataset received its label based on its filename, as mentioned in Section 3.1.

3.4 Data Preparation

Data preparation involved manipulating or removing data from each dataset prior to its integration into the machine learning model to enhance performance quality. This included handling NaN values and correcting errors in the data that could degrade predictive performance. For handling missing data in datasets the `SimpleImputer` class from the `sklearn.imputer` module is used, where it replaces missing values with the mean of the column [61].

As mentioned in Chapter 3.1, when the operator needed to intervene and disrupt the process, the remelting process would begin, causing the seed speed to reverse, resulting in a negative seed speed value. Therefore, it was necessary to remove the data recorded after the occurrence of structural loss.

The datasets were divided into segments corresponding to the neck, crown, shoulder, and body, enabling individual inspection of each ingot region. This segmentation was facilitated using the 'state' column, as illustrated in Figure 3.1. For example, data for the crown were extracted between state numbers 180 and 190. This method allows for predictions to be made for each region of the ingot and facilitates an investigation into which region the machine learning models perform best. Each region was segmented into second intervals, and owing to the varying lengths of the regions, they were investigated across different intervals. More details on this will be explained in Section 4.4. This segmentation strategy enabled determination of the periods during which the model exhibited optimal performance.

However, during the splitting of the ingot into segments, certain rows at the beginning, and the end had to be removed due to potential misleading values, particularly in the ingot length parameter. For instance, when isolating crown datasets, the initial row of the length parameter often exhibited significantly larger values compared to the remainder of the dataset. This discrepancy likely stemmed from the length parameter being reset during the transition to a new ingot region, leading to inclusion of length values from the preceding phase. Therefore, it was necessary to remove the first two rows in each dataset when splitting the ingot into segments.

In Subsection 3.2 it was mentioned that the ingot length was reset to zero at the transition from one ingot region to another. To create more realistic data, it was decided to adjust to a cumulative length throughout the entire ingot. This adjustment was achieved by adding the current length to the last recorded length from the previous region. For example, the cumulative length in the crown was calculated by adding each value in the ingot length column to the final ingot length value from the neck.

In the end, `StandardScaler()` from `sklearn.preprocessing` in python was used to standardize the variables of each datasets by subtracts the mean value of each variables. This centers the data around zero, ensuring that the mean of each feature is zero. It ensures that all variables contribute equally to the model by having the same scale.

Chapter 4

Methods and Machine Learning Models

This section outlines the selection of machine learning models and the procedure for enhancing these models through the adjustment of the data input. Additionally, the model can be employed to estimate the potential time savings that may be achieved based on its predictions.

4.1 Models

Three different machine learning algorithms were chosen for comparison as binary prediction models: logistic regression, random forest, and deep learning (neural network). These selections were made because each algorithm is constructed differently, as clarified in Chapter 2. The Scikit-learn modelling library was utilised for all three algorithms to conduct predictions.

Logistic regression was opted for as it is a widely used supervised machine learning algorithm for binary classifications. Its simplicity in implementation, ease of comprehension, and relatively good predictive performance were key factors. Details of logistic regression's operation are outlined in Chapter 2.4. The logistic regression modelling employed the `LogisticRegression()` function imported from the `Sklearn.linear_model` sub-library.

Random forest was selected for its ensemble method, which aggregates predictions from smaller models, typically decision trees, as described in Chapter 2.5. Random forest demonstrates effectiveness with large and intricate datasets while mitigating overfitting. Modelling random forest entailed using the `RandomForestClassifier()` function imported from the

Sklearn.ensemble sub-library.

Neural network algorithms, built upon multilayer perceptrons, offer a distinct approach from logistic regression and random forest, as elaborated in Chapter 2.6. Neural networks excel in handling non-linearity and feature interconnections, making them adept at processing voluminous and complex data. When modelling the neural network, the MLPClassifier() function from the Sklearn.neural_network sub-library was utilised.

4.2 Feature Extraction

Features are the input variables used to predict the target variable in a model and are crucial in determining the performance and accuracy of the predictive model. They are attributes or properties of the data that describe observations within a dataset. Each feature represents a dimension in the data, collectively defining the space in which the data points reside. Feature extraction involves selecting and transforming raw data into features suitable for modeling [62]. The features for extraction were chosen based on the data discussed in 3.2. Each region of the ingot possesses its distinct features, which are detailed in Tables 4.1, 4.2, 4.3 and 4.4, along with the name and description of each feature.

Table 4.1 presents the features for the neck. Here, the parameters were examined both individually and in correlation. No observable differences were noted when comparing the features between ingots with and without structural loss. Hence, there is no need to include figures for the feature comparison.

Table 4.1: Feature Descriptions for the neck.

Feature Name	**Explanation**
Heater Power Stability	Variance of heater power, indicating stability or fluctuation.
Diameter Adjustment Frequency	Count of diameter direction changes, reflecting adjustment frequency.
Seed Speed Heater Power Ratio	Mean ratio of seed speed to heater power, indicating process balance.
Rate Change Heater Power	Mean absolute change in heater power, highlighting adjustment frequency and significance.
Rate Of Change Length Per 30Sec	Average rate of length change per 30 seconds, indicating growth speed.
Cumulative Length Growth	Total length growth over the observation period, indicating overall growth.
Average Rate Diameter Change	Average rate of diameter change, indicating diameter evolution.
Seed Speed Diameter Change Correlation	Correlation between seed speed and diameter changes, indicating the effect of seed speed adjustments.
Heater Power Diameter Change Correlation	Correlation between heater power and diameter changes, indicating the impact of temperature control.

Table 4.2 displays the feature calculations for the crown; here, the parameters are individually examined and correlated. A key distinction for the crown, compared to the neck, is the exclusion of the Seed speed parameter as a feature. This exclusion is based on NorSun's guidance that the seed speed should remain relatively constant in the crown region.

Table 4.2: Feature Descriptions for the crown.

Feature Name	**Explanation**
Stability Indicator	Instances where heater power decreases and diameter increases simultaneously.
Diameter Increase Rate	Average rate of diameter change.
Length To Diameter Ratio	Ratio of final length to final diameter, indicating crystal shape and quality.
Diameter Increase Variability	Standard deviation of diameter changes.
Length Change Rate	Average rate of length change, showing crystal growth speed.
Heater Power Variability	Standard deviation of heater power changes.
Total Heater Power	Sum of heater power over the observation period.
Total Diameter Increase	Total increase in diameter over the observation period.
Energy Efficiency Indicator	Ratio of total heater power to total diameter increase.

Table 4.3 presents the features calculated in the shoulder region. The shoulder phase of the pulling process is relatively quick yet complex, with the goal of reaching the body diameter. Consequently, the majority of features focus on the diameter when investigating the shoulder of the ingot.

Table 4.3: Feature Descriptions for the shoulder.

Feature Name	**Explanation**
Stability Indicator	Instances where seed speed decreases and diameter increases.
Diameter Increase Rate	Average rate of diameter change, crucial for reaching body diameter.
Length to Diameter Ratio	Ratio of final length to final diameter, indicating shape and proportionality.
Diameter Variability	Standard deviation of diameter changes, assessing stability.
Length Change Rate	Average length change per time step during the shoulder phase.
Seed Speed Variability	Standard deviation of seed speed changes, reflecting adjustment consistency.
Total Heater Power	Sum of heater power used, indicating energy expenditure.
Total Diameter Change	Total diameter change during the shoulder process.
Energy Efficiency Indicator	Heater power used per unit of diameter change.
Seed Speed Adjustments	Sum of absolute changes in seed speed.

Table 4.4 displays the features considered for the body phase, where Chapter 2.2 notes that the seed speed remains relatively constant to control the diameter. Consequently, the focus is placed on diameter control through speed adjustments. Additionally, it is noted that a temperature drop occurs during the transition from shoulder to body; therefore, changes in heater power will also be examined.

Table 4.4: Feature Descriptions for the Body

Feature Name	**Explanation**
Seed Speed Response to Diameter	Responsiveness of seed speed changes to diameter changes.
Heater Power Response	Average change in heater power.
Diameter Control Efficiency	Ratio of absolute diameter change to absolute seed speed change.
Heater Power Drop Indicator	Sum of all heater power changes during the body phase.
Growth Rate	Average diameter change during the body phase.

4.2.1 Feature Importance

Feature importance scores, play a vital role in assessing the significance of individual features within a dataset when constructing a machine learning model. Each of the three machine learning algorithms utilises unique methodologies to ascertain the importance of each feature. Consequently, the relative importance of features can differ between models. For simplicity, the feature importance was evaluated for the interval with the best performance. The feature importance plots can be seen in the Appendix A.

Logistic Regression

From the features listed in Table 4.1, it is observed that 'Rate change heater power' and 'Seed speed heater power ratio' are the two features with the highest importance. This indicates that changes in heater power are crucial predictors, significantly influencing the model's performance. Conversely, features such as 'Cumulative length growth' and 'Diameter adjustment frequency' have minimal impact on the model’s predictions. Consequently, they will be removed to enhance the model’s performance.

The ”Diameter Increase Rate”, ”Energy Efficiency Indicator”, and ”Total Diameter Increase” are the features exerting the most significant influence on the model’s predictions in the crown, as listed in Table 4.2. Conversely, the ”Length to Diameter Ratio” appears to have the weakest impact on the model’s predictions.

The features of the shoulder region are listed in Table 4.3. The two most influential features identified are ”Total Heater Power” and ”Seed Speed Variability”. Conversely, ”Diameter Increase Rate” and ”Total Diameter Change” were found to have minimal impact on the predictions. In the body region, as detailed in Table 4.4, ”Seed Speed Response to Diameter” is the most impactful feature. ”Growth Rate”, however, shows very low impact compared to the other four features, suggesting that excluding ”Diameter Control Efficiency” from the final predictive model could be beneficial.

Random Forest

When identifying the importance of features for the random forest in the neck for those described in table 4.1. The analysis revealed that the two most significant features were ”Seed speed heater power ratio” and ”Rate change heater power” The feature with the least importance are ”Diameter adjustment frequency.”

Analyzing the crown with the features described in Table 4.2, it was found that the ”Energy Efficiency Indicator” was the feature with the highest importance. ”Heater Power Variability” and ”Total Heater Power” were found to be the features with the least influence on the model, suggesting they should be excluded when performing predictions.

For the features used in the shoulder region, as explained in Table 4.3, it was identified that ”Total Heater power” and ”Length change rate” was the features with the strongest influence on the predictions. ’Energy efficiency indicator’ was the feature with the least influence on the predictions. For the features described for the body, in Table 4.4, it was found that ”Seed speed response to diameter”, ”Diameter control efficiency” and ”Growth rate” had the highest importance, and ”Heater power response” was the least important feature when performing predictions.

Neural Network

When investigating the importance of the neck feature with the Neural network, it was found that the ”Cumulative length growth” and ”Average rate diameter change” is the features with the strongest influence on the prediction model. The ”Rate change heater power” and ”Heater power stability” features has the weakest influence on the model of those listed in Table 4.1. For the crown, ”Total diameter increase” and ”Diameter increase rate” was the feature with the highest impact, while ”Length change rate” had the least influence on the predictive model.

For the features used in the shoulder region, the ”Total heater power” was found to have the strongest influence, while the ’Length to diameter ratio’ was found to have the weakest influence. The ”Seed speed response to diameter” has the strongest impact on the body, while the ”Diameter control efficiency” has the least impact on the neural network model.

4.3 Optimising Data Selection

Optimising the selection of input data was crucial for achieving realistic and accurate model performance. Adjustments were made to specifically extract datasets relevant to the region under investigation, significantly enhancing model predictions. For instance, selecting datasets where structural loss was detected close to the region of interest led to notable improvements in performance and reliability. Various scenarios were explored to achieve optimal data selection, which are described in more detail below.

Early Attempts

In the first attempts of performing predictions with the machine learning models, it was used all files after data preparation explained in Section 3.4. This model modification resulted in an accuracy of approximately 51% for the neck and 78% for the crown. Although the model performed well with crown data, validation was crucial due to inconsistencies between datasets with and without structural loss. When structural loss was detected, data was clipped, creating unrealistic parameter differences between the classes and compelling the model to classify the presence of structural loss.

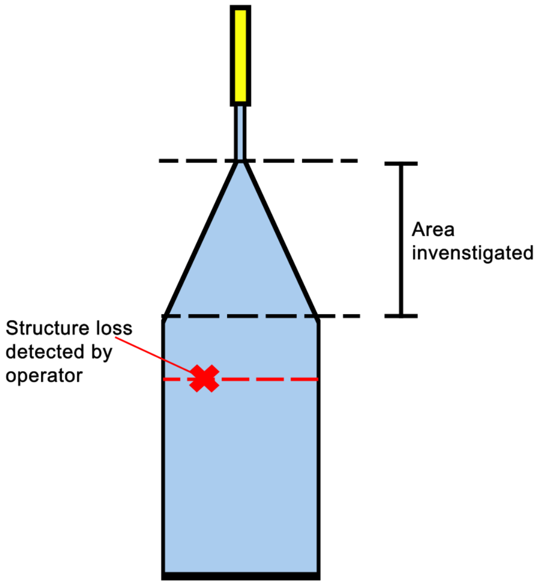


Figure 4.1: Structure of datasets with structure loss when investigating the crown.

It was decided to use datasets where structural loss had been detected by the operator in the next region of the ingot. For instance, when investigating the crown it was used datasets with structure loss had been detected in shoulder or body, as depicted in Figure 4.1. This approach ensured all datasets represented a complete crown. When these datasets were utilized in the predictive model, the accuracy was around 50%, equivalent to randomly guessing by the model. The low accuracy can be attributed to the uncertainty of where the error leading to structural loss occurred; thus, a dataset labeled as having structural loss could still possess a structurally good crown, with errors occurring later in the process. This outcome led to the conclusion that a reevaluation of the datasets labelled with structural loss to be included as input to the predictive models was necessary. Thus, the solution for the final model was established, as described below.

Final Model

Given the previously mentioned unsatisfactory outcomes, it was assumed that datasets where structural loss was detected closer to the area under investigation would yield better results. Therefore, the data for each ingot region was segmented into intervals, with a post-interval used to determine if structural loss was detected by operator within this post-interval. The post-interval begins immediately after the initial interval, and if a dataset concludes within the post-interval, the data within the interval being investigated is used as input. Figure 4.2 illustrates this procedure, where the red-line indicates where the operator detected structure loss and the interval is the area where the predictions are being made. However, it should be noted that this method is subject to a limitation in that only a limited dataset is available within each interval due to the selective nature of the post-interval. Consequently, the amount of datasets in the training and test sets will be considerable reduced.

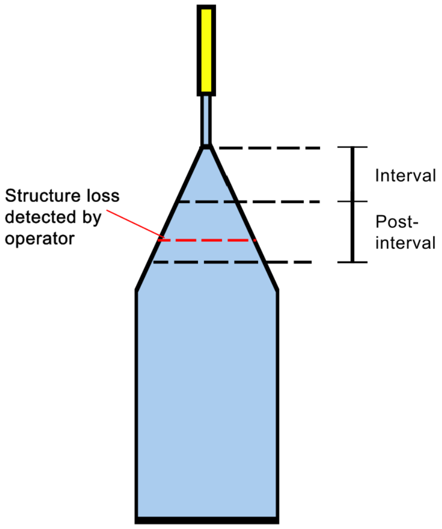


Figure 4.2: Illustration of the procedure when investigating the crown.

4.4 Development of the Final Model

This section details the development of the final model for predicting structural loss in ingot production. The process involves segmenting ingot data into distinct regions: neck, crown, shoulder, and body—and applying tailored training approaches for each. A key focus was balancing datasets between instances with and without structural loss to enhance model accuracy.

Each region of the ingot differs in length, resulting in varied interval lengths and the use of post-intervals for extracting datasets. The post-intervals, defined uniquely for each phase, play a crucial role in dataset selection and model performance. Details on how each post-interval is determined and its impact on the model are further explained below.

Model Development

When employing the procedure illustrated in Figure **??**, it was necessary to investigate each ingot phase individually. This requirement emerged from the differing lengths of the regions, resulting in varied intervals and quantities of datasets used for training and testing the model. The intervals are defined based on the dataset lengths, recorded every second of the Czochralski (Cz) process.

In the context of binary classification, it is essential to maintain a balance between datasets with and without structural loss. To achieve this, each interval was balanced by selecting the minimum number of datasets from the two categories. The number of datasets chosen varies based on the post-interval length; a longer post-interval leads to the selection of more datasets, including those where structural loss is detected further from the area under investigation. This method results in the filtering out of multiple datasets, thereby reducing the amount of data in each model. It is common practice to split the data into training, validation, and test sets. Due to the limited data available, 70% was allocated for training and 30% for testing. To optimise the use of the training data, cross-validation was performed.

Neck

No structural loss was detected in the neck region by the operator; hence, the post-interval commenced at the start of the crown. This strategy allows for the utilisation of ingots with the nearest detection of structural loss for training and testing the model. To analyse the neck region of the ingot, the interval was divided into 4200-second segments with 200-second increments. This division suggests that the neck-pulling process lasted for 4200 seconds. The endpoint was chosen based on the assumption that most datasets would have a neck length of 4200 seconds. Beyond this, the number of ingots exceeding 4200 seconds in length would be too few for effective training and testing.

Crown

A significant proportion of the ingots exhibiting structural loss was observed in the crown region. This observation suggests that it was possible to set the post-interval straight after the interval investigation. The intervals are defined so it would start from the start of the crown at each interval and increase by 200 seconds until it reaches 6400 seconds. The endpoint is selected based on the mean length of the crowns.

Shoulder

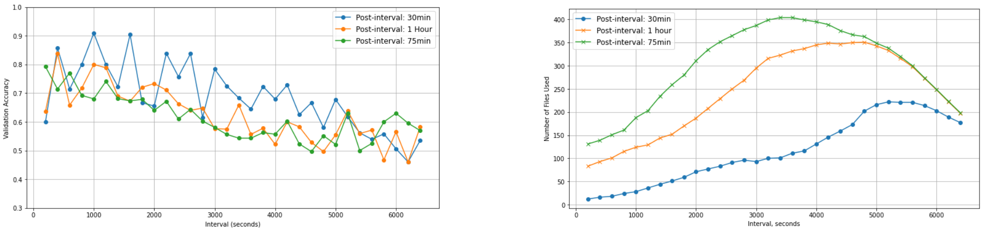
The length of the shoulder is considerably shorter than that of the neck and crown. Consequently, the intervals commence at the beginning of the shoulder and increase by 100 seconds until they reach 600 seconds. The post-interval is defined in a manner similar to that of the neck, with the beginning set at the beginning of the body and extending further into the body.

Body

As mention in section 3.1 the datasets contains data from the start of neck and three hours after the start of the crown, this means that data of the whole body region of the ingot will not be included. That's why the intervals would only exceeds 3800 seconds with step every 200 seconds from the beginning of the body. The post-interval is defined similar as done for the crown, where it starts right after the interval looked at.

Selections of Post-Interval

The length post-interval is selected based on the findings presented in Figure 4.3, which analyse different post-intervals in the crown region, namely 30 minutes, one hour and 75 minutes. Figure 4.7(a) illustrates the accuracy of the selected post-interval, while Figure 4.7(b) shows the number of files used. A closer examination of the models was found necessary, and thus the 1-hour post-interval was selected for analysis. This approach utilises a post-interval with a relatively stable accuracy curve, a greater availability of datasets, and a distance that is close to the interval under investigation.



(a) Validation Accuracy Across Intervals

(b) Number of Files Used Across Intervals

Figure 4.3: Analysis of post-interval effects.

4.5 Hyperparameters Tuning

Hyperparameter tuning is a crucial technique aimed at identifying the optimal hyperparameters to enhance model performance. As outlined in Section 2.7, each machine learning algorithm possesses its unique set of hyperparameters. In this thesis, Sklearn's automated function GridSearchCV has been employed for hyperparameter tuning. GridSearchCV systematically explores combinations of parameters to identify the best combination for the specific dataset and model. Additionally, GridSearchCV utilises cross-validation to evaluate the model's robustness by testing it on multiple subsets of the training data. Given the model's definition with intervals, it's important to note that each interval will have its own specific hyperparameter combinations.

Table 4.5 presents the hyperparameters that were prioritised during the tuning of the logistic regression model, while leaving the remaining hyperparameters at their default values. 'C' serves as a parameter that represents the inverse of regularization strength, aiding in the prevention of overfitting by penalising larger coefficients in the model. It was determined to explore a wide range of 'C' values (0.01 to 100) to assess the model's sensitivity to changes in regularization strength.

Table 4.5: Hyperparameters for Logistic Regression

`Hyperparameter`	**Default Value**	**Search Values**
`C`	1.0	0.01, 0.1, 1, 10, 100
`penalty`	'l2'	'l1', 'l2', 'none'
`solver`	'lbfgs'	'liblinear', 'lbfgs'

The 'penalty' parameter in logistic regression refers to a form of regularization. 'l1' penalty aims to reduce the number of features by driving the coefficients of less important features to zero. On the other hand, 'l2' penalty shrinks the coefficients towards zero without setting any to exactly zero, making it suitable for cases where many small or medium effects are expected. Setting the penalty to 'none' indicates that no regularization is desired, and the model will be trained without any regularization.

The 'solver' parameter represents the optimization algorithm used by logistic regression. Different solvers are optimized for various types of data and models, and the choice of solver can impact both the speed and stability of the model.

For more detailed information on the different parameters available for logistic regression, refer to the documentation provided in [40].

The parameters being used for tuning the RandomForestClassifier model are shown in Table 4.6. n_estimators are the number of trees in the forest, max_depth are the maximum depth of the trees, where the default is set to None, which mean that the nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples. min_samples_split is the minimum number of samples required to split an internal node. For further details regarding the parameters available for the random forest classifier, refer to the documentation provided in [47].

Table 4.6: Hyperparameters for RandomForestClassifier

`Hyperparameter`	**Default Value**	**Search Values**
`n_estimators`	100	100, 200, 300
`max_depth`	None	10, 20, 30, None
`min_samples_split`	2	2, 5, 10

Table 4.7 shows the parameters used for tuning the MLPClassifier. Hidden_layer_sizes represent the number of neurons in the hidden layer, while activation is the activation for the hidden layer. The learning_rate_init controls the step-size in updating the weights. For additional details regarding the other parameters available for the MLPClassifier, refer to the documentation provided in [53].

Table 4.7: Hyperparameters for MLPClassifier

`Hyperparameter`	**Default Value**	**Search Values**
`hidden_layer_sizes`	(100,)	(50,), (100,), (100, 50)
`activation`	'relu'	'identity', 'logistic', 'relu'
`learning_rate_init`	0.001	0.001, 0.01, 0.1

4.6 Time-Saving Model

If structure loss occurs in the early stages of growth, the ingot will be remelted back to the seed. If structure loss occurs at a later growth stage, the growth is stopped, and the failed part is cut off. The structure loss is determined by the operator by visual disappearance of so-called ’growth ridge’ or ’node’ during the growth phase. The remelting process is time consuming and the earlier structure loss it is detected after it happens the less time is lost.

The machine learning model that utilises the best performance, aims to predict early-stage structure or non-structure loss during ingot growth. The model continuously evaluates whether ingot pulling should proceed or undergo remelting. This decision is contingent upon the model’s accuracy within specific intervals along the ingot. Accordingly, an accuracy threshold is set to determine whether remelting should be considered.

The time-saving model should follow the same procedure and data selection outlined in Figure 4.2. For intervals that meet the accuracy threshold, ingots labeled with structural loss and their lengths and diameters at these intervals will be used to estimate the remelting time. Additionally, the lengths and diameters of these ingots will be recorded when the operator detects structural loss. This approach enables the estimation of time saved by relying on the model, comparing remelting times when the model shows high accuracy to those when the operator detects structural loss.

The remelting time was calculated using a remelting rate, R_m, an estimate provided by NorSun, and varies from 4 to 7 kg/h. The initially volume was calculated using the length and diameter of the ingot. The weight of the ingot was calculated by multiplying the volume by the density of silicon $(2330kg/m^3)$. Finally, the remelting time, R_t was determined by dividing the weight by the remelting rate, as shown in equation 4.1.

$$R_t = \frac{weight}{R_m} \tag{4.1}$$

The estimate will only be conducted for the crown, as using a machine learning model is most beneficial for this region when considering remelting time. Although the machine learning models showed great potential in the neck, the remelting time for this area is significantly low, so it will not be considered for the time-saving model. Estimates for time saved will also not be performed for the shoulder and body due to poor prediction accuracy in these regions.

The accuracy can never be 100%, meaning that prediction errors will occur, leading to False Negative and False Positive cases. In the event of a false negative, the model may suggest that the Cz process should continue for an ingot with structure loss. If a false positive is identified, the model will suggest remelting an ingot that has not undergone any structural loss. In such a scenario, both time and material would be wasted.

The calculation of time lost due to unnecessary remelting operations involved determining the average time taken for a remelting process and multiplying this by the number of false positives. Conversely, the time saved was calculated by multiplying the number of true positives by the time saved per correct prediction to determine the total time saved for each interval.

Chapter 5

Results and Discussion

This chapter presents the findings of this study, focusing on the predictive performance of logistic regression, random forest, and neural network models in detecting structural loss during the Czochralski process. This chapter evaluates the effectiveness of these models across different regions of the ingot: neck, crown, shoulder, and body. The results presented here are based on predictions performed on test data.

The analysis begins with the neck region, examining model stability and key features influencing predictions. Next, the crown region is discussed, noting the initial high performance of the models and subsequent decline. The shoulder and body regions are then explored, highlighting the challenges faced by the models in these areas and evaluating their performance metrics and feature importance. Additionally, the chapter introduces a time-saving model based on the random forest algorithm, demonstrating its ability to reduce remelting time through accurate predictions.

Throughout the analysis, hyperparameter tuning and feature importance assessments, as explained in Section 4.5 and 4.2.1. Will be applied to optimize model performance and understand the critical factors driving predictions.

By providing a comprehensive analysis and discussion, this chapter aims to assess the strengths and limitations of each machine learning model and their potential to enhance the Czochralski process, paving the way for future research and practical applications.

5.1 Neck

The same number of datasets was used across all machine learning models, forming a training set of 85 ingots, which was subsequently tested on an unseen set of 37 ingots. This relatively small number results from employing a post-interval extraction method for datasets with structural loss detected within the first hour of the crown.

5.1.1 Logistic Regression

Figure 5.1 illustrates the Receiver Operator Characteristic - Area Under the Curve (ROC AUC) scores of the neck with a logistic regression model. This score indicates the model's ability to distinguish between ingots without structure loss and those with structure loss. At first glance, it may appear that the model varies considerably, but it remains relatively stable between 0.70 and 0.78 throughout the neck. However, at the start of the neck, there is a larger variation. This period is designated as the pre-neck, during which the diameter is reduced from approximately 12mm to 5mm. The objective is to estimate whether the temperature is too cold or hot, as stated in Section 3.2. The poor performance of the model in this phase may be due to the parameters being more fixed in the pre-neck.

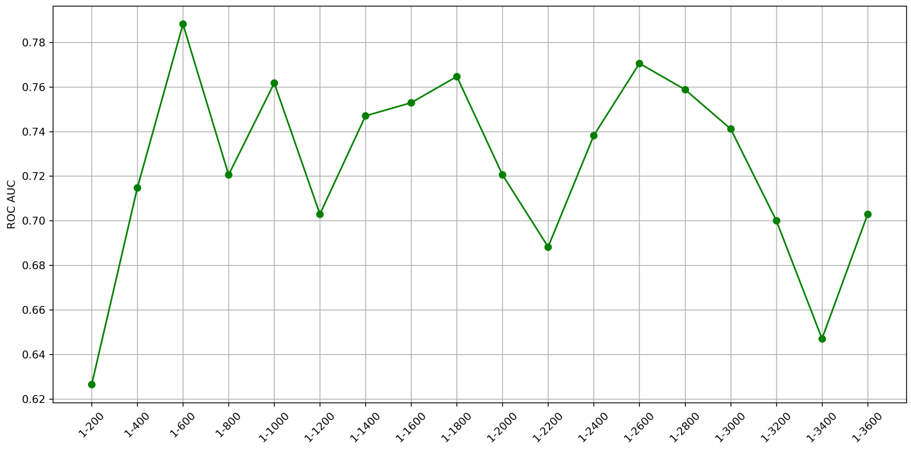


Figure 5.1: ROC AUC scores with logistic regression in the neck.

A feature importance analysis was conducted with the objective of identifying which feature had the greatest impact on the predictions. As previously stated in Section 4.2.1, it was determined that the two most influential features on the model's predictions were ”Rate change Heater power” and ”Seed speed heater power ratio”. These are further explained in Table 4.1. This indicates that the change in heater power and the ratio between seed speed and heater power exert a significant influence on the model's performance. The seed speed is controlled by the automatic diameter control (ADC1), which attempts to maintain a setpoint diameter by adjusting the seed speed. ADC2 regulates the temperature by controlling the heater power. As stated in Section 3.2, an elevated seed speed indicates a too low melt temperature, necessitating an increase in heater power. This could indicate that changes in seed speed or heater power might lead to structural loss. Potentially, this explains why the ”Rate change heater power” and ”Seed speed heater power ratio” features significantly influence the model's predictions.

Table 5.1 presents performance scores for three different intervals, representing early neck, middle neck, and end neck. For interval 1-1200, the accuracy rate is 62% and the precision rate is 57%. This indicates that 62% of predictions are correct on the test dataset, and 57% of cases identified as structure loss are classified correctly. With a recall rate of 71%, it can be seen that the model is correctly identifying the portions of ingots labelled with structure loss. This high recall rate, compared to the overall accuracy, suggests that the model is better at identifying ingots with structural loss than those without it.

Table 5.1: Performance scores across three intervals in the neck with logistic regression.

	1-1200	1-2400	1-3600
Accuracy	0.62	0.60	0.65
Precision	0.57	0.54	0.60
Recall	0.71	0.82	0.71

For the interval 1-2400, there is a slight decline in the accuracy and precision of the results, with an accuracy rate of 60% and a precision rate of 54%. In contrast, the recall rate has increased significantly, reaching 82%. Consequently, the model will have fewer incorrect predictions regarding the ingots label with structure loss. In the case of predictions conducted on ingots without structure loss, the model would have made more mistakes compared with the previous interval due to the decline in accuracy and precision. When the Czochralski process of the neck proceeds the accuracy and precision increase to a rate of 65% and 60%, respectively, while the recall rate declines slightly to 71%.

To summarise the performance scores from Table 5.1, a confusion matrix will be used to compare predicted labels with true labels for the 37 ingots in the test set. Table 5.2 displays the confusion matrix for interval 1-1200. The model correctly predicted 12 ingots as having structural loss, termed True Positive (TP), and 11 ingots as not having structural loss, termed True Negative (TN). Five ingots were incorrectly predicted as not having structural loss, termed False Negative (FN), where their actual labels indicated structural loss. The matrix also recorded nine cases termed False Positive (FP), where the model incorrectly predicted structural loss for ingots without such loss. If the model were completely trusted, the FN cases might only be identified in a later interval. Conversely, with FP cases, the model might erroneously initiate the remelting process for nine ingots without structural loss in interval 1-1200. Given this high error rate, it might be beneficial not to trust the model at this early stage in the neck and allow it to continue uninterrupted.

Table 5.2: Confusion Matrix for interval (1-1200) with logistic regression in the neck.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	11	9
Structure loss	5	12

Table 5.3 presents the confusion matrix for interval 1-2400. Observing a slightly decline in accuracy and precision rate compared with interval 1-1200 in Table 5.1, but the recall rate would increase. This can been seen in the confusion matrix, where it has a greater amount of True Positive with 14 ingots predicted correctly with structure loss. A decline will be seen for the True Negative cases with just 8 ingots predicted correctly without structure loss. For False Negative case it was only 3 ingots, but for False Positive cases it was 12 ingots predicted as structure loss but actual label as non-structure loss. This interval indicates that the model may contain a significant number of errors, whereby a substantial quantity of ingots may be remelted unnecessarily. This demonstrates that it could be more beneficial to have a greater number of false negatives than false positives for a machine learning model in the Czochralski process.

Table 5.3: Confusion Matrix for interval (1-2400) with logistic regression in the neck.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	8	12
Structure loss	3	14

The confusion matrix for interval 1-3600 are presented in Table 5.4, which represents the end of the neck. The model utilize 12 cases of True Positive and 12 cases of True Negative. For the False Negative it was 5 ingots that was predicted without structure loss but actual they had structure loss. 8 cases of false positive results were observed, indicating that the model continues to exhibit significant uncertainty in its predictions regarding the remelting decision for the neck.

Table 5.4: Confusion Matrix for interval (1-3600) with logistic regression in the neck.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	12	8
Structure loss	5	12

Figure 5.2 illustrates the cross-entropy loss scores across intervals in the neck using logistic regression. As discussed in Section 2.9.7, cross-entropy measures the discrepancy between the predicted probability distribution and the actual label, with a low score indicating good performance. Overall, the model demonstrates a fair performance, maintaining a consistent mean of 0.64 and a standard deviation of 0.03 throughout the neck, with loss function ranges from 0 to infinity. Interval 1-600 shows a significant improvement in loss score, coupled with a high ROC AUC score from Figure 5.1, suggesting reliable model performance. However, in interval 1-2600, despite a relatively high ROC AUC, the model presents a poor cross-entropy loss score. This indicates overconfident predictions, where the model assigns very high probabilities to incorrect predictions, a scenario heavily penalized by the function. These overconfident predictions could be attributed to the small size of the training and test sets, which prevents the model from effectively learning the underlying patterns in the data.

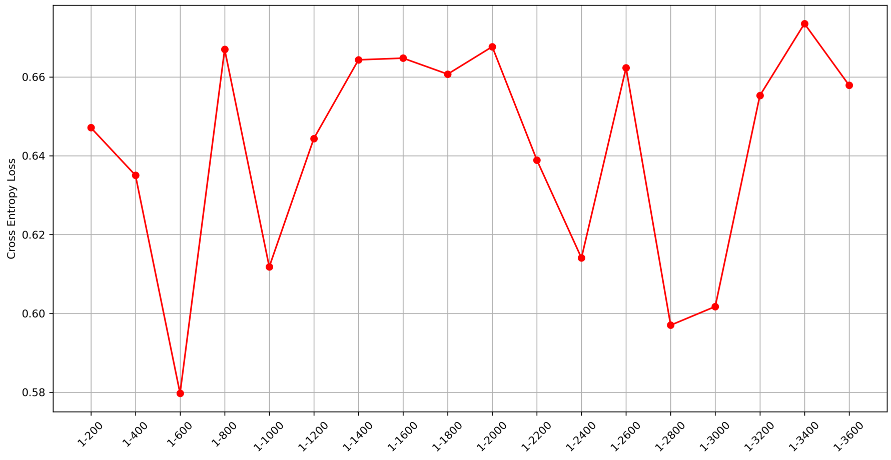


Figure 5.2: Cross Entropy Loss with logistic regression in the neck.

5.1.2 Random Forest

In order to provide an overview of the manner in which the random forest model differentiates between ingots with and without structural loss, the ROC AUC score for each interval is presented in Figure 5.3. Initially, the performance is relatively weak, but subsequently remains relatively stable from interval 1-1000 until the end of the neck. The low ROC AUC score can be explained by the fact that in this phase of the neck called the pre-neck, the parameters are more fixed, which would mean that the data are more similar. After the pre-neck phase, the model demonstrates a stable ROC AUC score around 0.67 and 0.75.

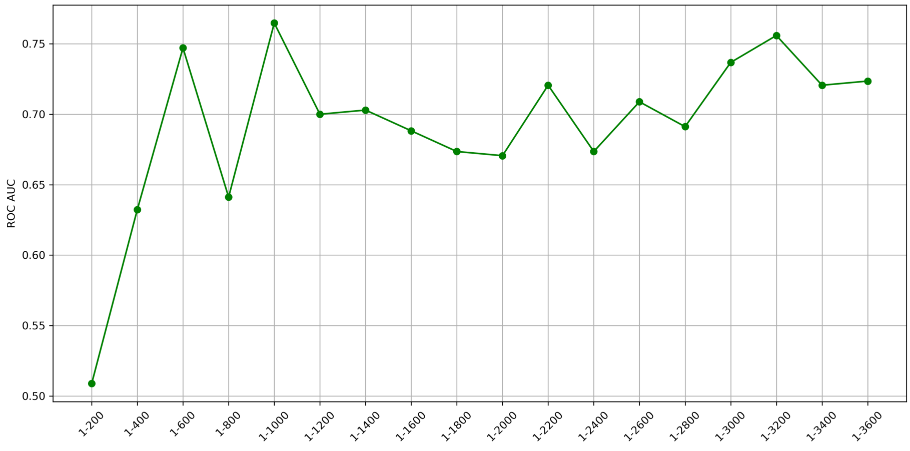


Figure 5.3: ROC AUC scores with random forest in the neck.

As stated in [27], a good neck grows fast and thin to avoid dislocations in the crystal that would result in structural loss. To achieve this, seed speed and heater power are important parameters to control. Therefore, ”Seed speed heater power ratio” was shown to be a crucial feature in the predictions, indicating the balance between heater power and seed speed. The diameter should stay within an accumulated limit and reach a given length, as explained in Section 3.2. When the diameter exceeds the limit, the integration stops, and the heater power and seed speed must be adjusted to bring the diameter within the limit. Therefore, changes in the diameter could be an indicator of errors in the process and may explain why the feature ”Average rate diameter change” significantly impacts the predictions. When the diameter descends below the limit, it is a sign that the melt is too hot, indicating that the heater power needs adjustment and that the seed speed is too high. According to [12], high seed speed is beneficial for achieving a thin neck but also carries the risk of ”pop-out,” where the neck is pulled out of the melt.

Table 5.5 presents the performance scores across three intervals, representing the early, middle, and end stages of the neck. Interval 1-1200 exhibits an accuracy rate of 65%, a precision rate of 59%, and a recall rate of 77%. The good recall rate indicates that the model correctly predicts more ingots labeled with structural loss. However, lower accuracy and precision weaken the overall performance of the model in this interval. For interval 1-2400, all the performance scores decline, with an accuracy rate of 62%, a precision rate of 58%, and a recall rate of 65%. This decline suggests that it would probably not be beneficial to trust the model in this interval.

Table 5.5: Performance scores across three intervals in the neck with random forest.

	1-1200	1-2400	1-3600
Accuracy	0.65	0.62	0.70
Precision	0.59	0.58	0.69
Recall	0.77	0.65	0.65

For the interval 1-3600, there was a slight increase in performance scores, with an accuracy rate of 70%, a precision rate of 69%, and a recall rate of 65%. This indicates that the model is more trustworthy at the end of the neck, which could be due to the importance of having an accurate temperature before the transition to the crown.

The confusion matrix for all three intervals summarises the performance scores in Table 5.5. The confusion matrix for interval 1-1200 is presented in

Table 5.6. It shows 13 cases where the model correctly predicted structural loss, known as True Positives. For the True Negative cases, the model correctly predicted 11 ingots without structural loss. For the incorrect predictions, the model predicted 4 ingots as False Negatives, where they were predicted without structural loss but were actually labeled with structural loss. This would result in the model indicating that the Czochralski process should continue. 9 cases of False Positive were predicted by the model, which is a relatively high percentage of the ingots in the test data that would end up being remelted without any structural loss.

Table 5.6: Confusion Matrix for interval (1-1200) with random forest in the neck.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	11	9
Structure loss	4	13

Table 5.7 presents the confusion matrix of interval 1-2400, where the effect of the recall decline can be seen with 11 ingots predicted correctly as structure loss and 6 cases of False Negative. For the ingots without structure loss, it got predicted 12 ingots as True Negative, and 8 ingots got predicted as False Positive. As previously stated, the occurrence of false positives indicates that the ingots will be remelted, yet they remain structurally intact. This is of critical importance for the model to function effectively in a real-world scenario, as the number of false positives must be kept to an absolute minimum.

Table 5.7: Confusion Matrix for interval (1-2400) with random forest in the neck.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	12	8
Structure loss	6	11

From the confusion matrix in Table 5.8 it can be observed that the performance of the model has improved slightly. The only difference between the two intervals is the number of instances where the model predicted correctly without structure loss (15) and where it predicted structure loss but the actual label was without structure loss (5). As previously stated, this number is crucial for the model to function properly. Therefore, this confusion matrix indicates that the model is more credible. With regard to false negative cases, it is not of great importance to minimise the error, since the error can be detected by an interval that occurs later in the ingot.

Table 5.8: Confusion Matrix for interval (1-3600) with random forest in the neck.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	15	5
Structure loss	6	11

Figure 5.4 illustrates the cross-entropy loss scores across intervals in the neck using a random forest model, which achieves a mean score of 0.64 and a standard deviation of 0.04. This indicates moderate prediction accuracy with relatively low variability. Notably, intervals 1-1000 and 1-3200 both display excellent loss scores. Interval 1-1000 shows robust performance with one of the lowest loss scores and a high ROC AUC score, signifying high predictive accuracy. Similarly, Interval 1-3200 also exhibits a high ROC AUC, just above 0.75, suggesting that the predictions are reliable. Conversely, Interval 1-200, which shows a higher loss score and a poor ROC AUC score, reflects the model's struggle to make reliable predictions. The poor performance in this interval could be attributed to its limited size of only 200 data points from the start of the neck, whereas larger intervals contain more data points, providing a richer understanding of the process.

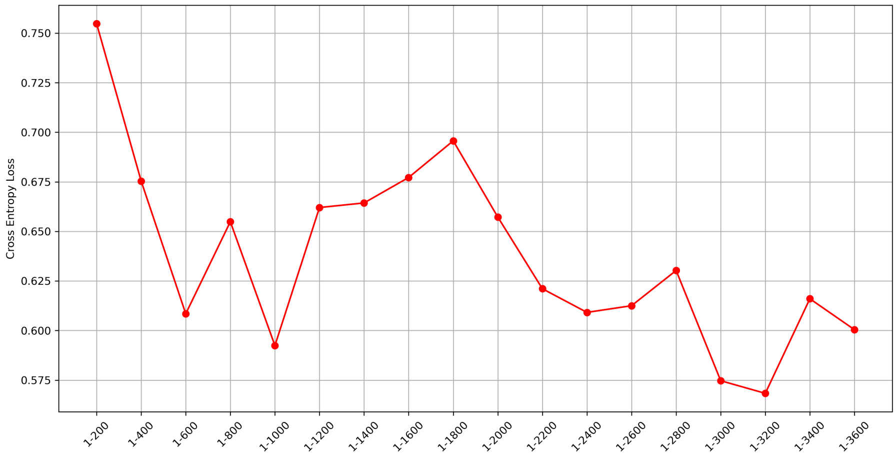


Figure 5.4: Cross Entropy Loss with random forest in the neck.

5.1.3 Neural Network

Figure 5.5 presents the ROC AUC score for the neck, obtained using a neural network model to demonstrate the overall performance. Upon initial observation, it appears that the scores are relatively stable, with the exception of one interval where the score drops to approximately 0.35. This drop is difficult to explain, as it could be caused by many factors like physical aspect in the Czochralski process, uncertainty in the data, where to few datasets to train and test this specific interval could have large impact on the drop in performance. After the drop in interval 1-800, the model performs relatively stable around a ROC AUC score of 0.65 and 0.70. This is initial after the pre-neck phase, where the diameter should increase from around 12mm to 5mm. This could indicate that the pre-neck phase is a area of interest when it comes to further quality of the neck.

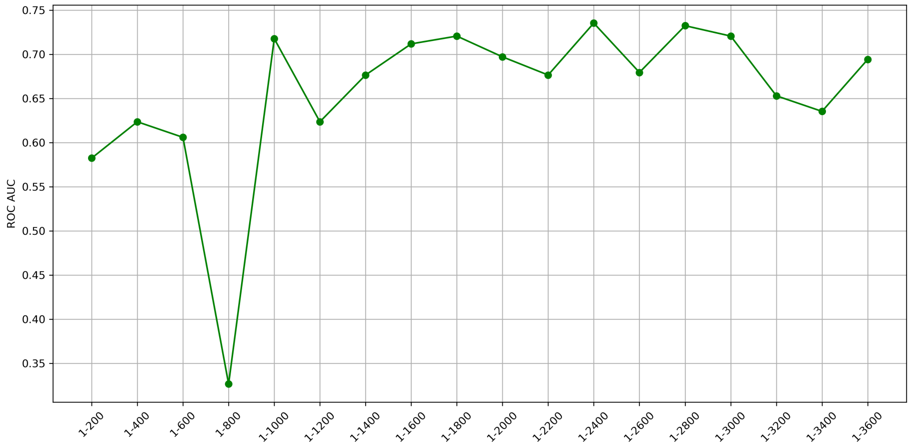


Figure 5.5: ROC AUC scores with neural network in the neck.

The feature importance analysis, as outlined in Section 4.2.1, revealed that the ”Seed speed heater power ratio” and ”Heater power diameter change correlation” had a significant impact on the model’s predictions. This suggest that the heater power is crucial parameter to control during the Czochralski process. The heater power together with seed speed controls the diameter and length, as stated in article [27] that a good neck should be grown fast and thin. After the decrease of the diameter in the pre-neck, the neck should be grown stable within the accumulated length. Meaning that the seed speed and heater power should not exhibit drastically changes during the neck, that will also make the diameter change drastically. The feature importance of ”Heater power diameter change correlation” indicates that the correlation between changes in heater power and diameter could have impact on those ingots with detection of structure loss.

In order to gain a more detailed understanding of the performance of the neural network model, three intervals representing the early, middle and end stages of the neck have been selected for closer investigation. These intervals are presented in Table 5.9. Interval 1-1200 demonstrates a poor performance from the model, with an accuracy rate of 51%, a precision rate of 48% and a recall rate of 59%. This indicates that the model would not perform well in this interval. Given that it is in the early stages of the neck, it may be beneficial to allow the Czochralski process to continue.

Table 5.9: Performance scores across three intervals in the neck with neural network.

	1-1200	1-2400	1-3600
Accuracy	0.51	0.62	0.62
Precision	0.48	0.60	0.58
Recall	0.59	0.52	0.65

In interval 1-2400, the performance improves slightly, with an accuracy rate of 62% and a precision rate of 60%. The recall rate drops slightly to 52%. For this interval, it would probably be more beneficial to allow the Czochralski process to continue. For interval 1-3600, representing the end stage of the neck, the performance does not improve significantly; only the recall improves slightly to 65%.

The evaluation scores for the model presented in Table 5.9, indicated a poorly overall performance of the neural network model. To closer look at the outcomes from the predictions it will be presented confusion matrix for the three intervals. For interval 1-1200 the confusion matrix is presented in Table 5.10 where is showcase the poorly predictions made by the model. The model predicted 10 ingots correctly with structure loss and 9 ingots correctly as without structure loss. For the incorrect predictions, it resulted in 7 cases of False Negatives and 11 cases of False Positives. This shows that the model should not be used in this interval.

Table 5.10: Confusion Matrix for interval (1-1200) with neural network in the neck.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	9	11
Structure loss	7	10

Table 5.11 demonstrates the confusion matrix for interval 1-2400, showing a slight improvement in the predictions. The number of True Positive cases declines to 9 ingots correctly predicted with structural loss, and as a consequence, the False Negative cases increase to 8. The most promising outcome from these results is the decrease in False Positive cases to 6, which means the model would suggest remelting fewer ingots without any structural loss.

Table 5.11: Confusion Matrix for interval (1-2400) with neural network in the neck.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	14	6
Structure loss	8	9

For the final interval, 1-3600, which represents the end stage of the neck, the confusion matrix is presented in Table 5.12. The high number of false negatives (6) and false positives (8) suggests that the model is uncertain about its predictions, which makes it unreliable.

Table 5.12: Confusion Matrix for interval (1-3600) with neural network in the neck.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	12	8
Structure loss	6	11

Figure 5.6 illustrates the cross-entropy loss scores across intervals in the neck using a neural network model, achieving a mean of 0.96 and a standard deviation of 0.66. This indicates significant variation in performance across different intervals. Specifically, the model shows low cross-entropy loss scores, below 0.75, during three periods. The first period, between intervals 1-200 and 1-600, records ROC AUC scores around 0.6, indicating moderate model

performance. The period from interval 1-1800 to 1-2800 also presents low loss scores, and the ROC AUC score slightly increases, suggesting reliable and accurate performance. Similarly, the period between intervals 1-3200 and 1-3600 indicates that the model makes reliable predictions. The figure also displays two significant peaks in loss scores, the precise causes of which remain uncertain. However, these peaks suggest that the model may exhibit unreliable predictions in these intervals, despite the model's high ROC AUC scores.

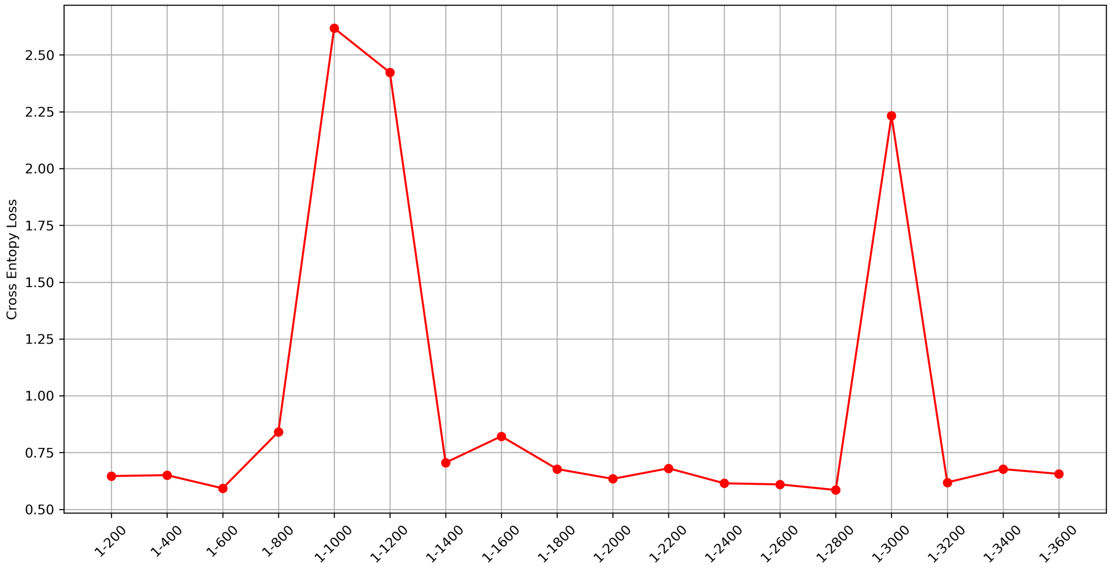


Figure 5.6: Cross Entropy Loss with neural network in the neck.

5.2 Crown

The three machine learning models were trained on 119 ingots, which were subsequently tested with unseen data from 51 ingots. Each model will also iterates over multiple intervals from the start of crown to the end with steps of 200 seconds.

5.2.1 Logistic Regression

Figure 5.7 illustrates the use of logistic regression as a prediction model, yielding an ROC AUC score on the test dataset across intervals. It is evident that the model demonstrates effective predictions in the initial stages of the crown. However, after approximately one hour, the ROC AUC score declines to around 0.5, indicating that the model performs random guessing from one hour until the end of the crown. The improved classification at the beginning and end of the crown may result from more complex adjustments in these regions during the Czochralski process. This is due to the need for rapid adjustments in seed speed and heater power, which are essential for the diameter increase and shaping of the crown, as stated in 3.2. From the start of crown to interval 1-1200, the logistic regression model shows ROC AUC scores between 0.80 and 0.90, suggesting that this period is crucial for classifying structure loss.

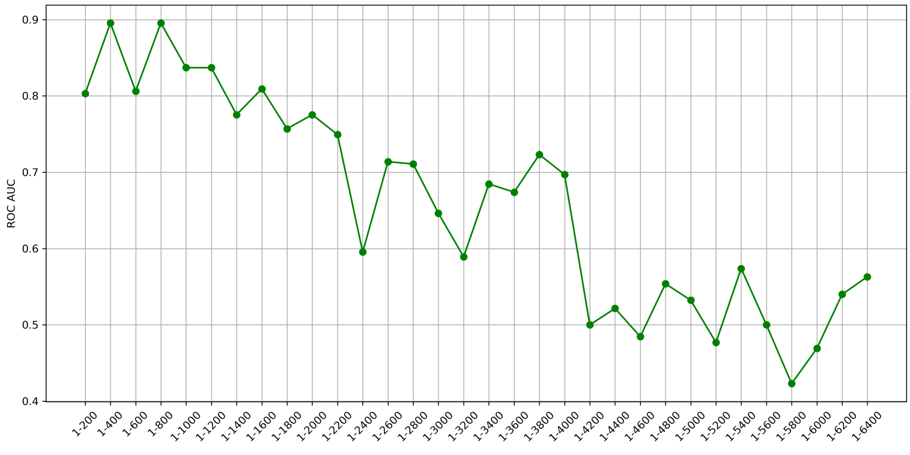


Figure 5.7: AUC scores with logistic regression in the crown.

In section 4.2.1, it was found that the 'Diameter Increase Rate' is the feature with the most influence on the model's predictions, indicating how quickly the diameter changes on average. This demonstrates the importance of diameter control and how the shape of the crown can influence predictions, especially in the early stages of the crown. The diameter change results from the seed speed and heater power; these parameters should be consistent across all crowns unless the operator adjusts the process. If the diameter changes are too small, the growth rate will also be slower, indicating that the melt is too hot and less stable, as stated in 3.2. Conversely, a rapid diameter change can indicate that the melt is too cold. Therefore, heater power variability is also an important feature. In the crown, the heater power should decrease steadily throughout, but in the beginning, the heater power might exhibit more variability, which could result in structure loss.

To closer investigate the performance of the model, Table 5.13 shows performance scores for three different intervals. It can be seen that the performance of the model decreases longer into the crown, where for the interval 1-1800 it has a accuracy of 71% and precision of 86%. This means that out of all predictions the model classifies 71% of the correct, and in number of cases identified as structure loss, 85% of them are classified correct. One hour (1-3600) into the crown the model display a small decrease in the performance, with a accuracy of 69% and precision of 70%. When the crown reaches the end, the model is essentially randomly guessing the predictions, with an accuracy of just 53% and a precision of 52%.

Table 5.13: Performance scores across three intervals in the crown with logistic regression.

	1-1800	1-3600	1-5400
Accuracy	0.71	0.69	0.53
Precision	0.86	0.70	0.52
Recall	0.48	0.64	0.64

The only score to increase further into the crown is the recall score, which measures the ratio of correctly predicted cases of structure loss to the total cases of actual structure loss. For the first half-hour (1-1800), the recall score is about 0.48, which may result in cases where the model predicts non-structure loss, but it actually is structure loss. This indicates that the model can better predict non-structure loss at the beginning of the crown. Closer to the end of the crown, the model can predict structure loss. However, overall, it utilizes a performance which is almost equal to random guessing, which suggest that the interval 1-3600 has the best performance.

To summarise the performance scores from Table 5.13, a confusion matrix will be used to compare predicted labels to the true labels. Table 5.14 presents the confusion matrix for interval 1-1800. It demonstrates how many of the 51 ingots from the test dataset were correctly and falsely predicted. As known from Table 5.13 interval 1-1800 had relatively good performance score and the function of these will be seen in the confusion matrix. The model predicted 12 ingots correctly as structure loss also known as True Positive, while 24 ingots got correctly predicted as non-structure loss, True Negative. For the falsely predicted it was in total 15 ingots that got predicted wrong by the model. 13 of them was actual with structure loss but the model predicted them as without structure loss, termed as False Negative. Only two ingots got predicted with structure loss, but actual was sampled without structure loss, also called False Positive.

As illustrated in Table 5.13, interval 1-1800 exhibited a relatively high performance score, as detailed in the confusion matrix. The model correctly predicted 12 ingots with structure loss, which is defined as a True Positive (TP). It also accurately identified 24 ingots without structure loss, which is defined as a True Negative (TN). A total of 15 ingots were incorrectly predicted by the model. Of these, 13 were cases of structure loss, yet were predicted as without, resulting in a false negative. Two ingots were incorrectly identified as having structural loss, despite the absence of such loss. This is termed a false positive.

Table 5.14: Confusion Matrix for interval (1-1800) with logistic regression in the crown.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	24	2
Structure loss	13	12

When the model is iterated continuously over intervals across the crown, it is possible for the model to identify errors in a subsequent interval. Consequently, the 13 ingots that were predicted as having no structural loss, but in fact did have structural loss, could be predicted correctly in a later interval. Because of this the recall score could be less weighted when evaluating the performance of the model in a specific interval. For instance, for the two false positive cases the model would indicate that the remelting process should start and then result in perfectly fine ingots would be remelted.

When passing around a hour into the Czochralski process of the crown it was found that the accuracy and precision had slightly decrease. Table 5.15 present the confusion matrix for interval 1-3600, with more balance distribution compare with one in Table 5.14 for interval 1-1800. Confusion matrix for interval 1-3600 utilize a larger portion of True Positive (TP), with 16 ingots predicted correctly with structure loss. For True Negative (TN) it was 19 ingots that got predicted without structure loss. As mention before the recall score would increase slightly the longer into the Czochralski process, that's why this interval has a greater portion of True Positive (TP) cases with just 9 ingots. On the other hand the model gets larger portion of the False Positive (FP), then by trusting the model would 7 good ingots be remelted. When considering the overall performance of the model in this interval (1-3600), it could be beneficial to trust the model when i comes to if the Czochralski process should continuous or remelting should start.

Table 5.15: Confusion Matrix for interval (1-3600) with logistic regression in the crown.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	19	7
Structure loss	9	16

Table 5.16 presents the confusion matrix for interval 1-5400, which is here most of the ingots reaches its end of crown before transition to the shoulder region. As previous mention the performance of the model in this period decreases, and from the confusion matrix below it can be seen that the model is almost randomly guessing. With 16 cases of True Positive (TP) predictions, and 11 cases of True Negative (TN). 9 cases of False Negative (FN), where the model predicts non-structure loss but the actual is samples as structure loss. The model will have larger portion of False Positive (FP) predictions with 15 cases. Due to the this performance of the model it could means that it would not be beneficial to trust the model in this interval.

Table 5.16: Confusion Matrix for interval (1-5400) with logistic regression in the crown.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	11	15
Structure loss	9	16

Figure 5.8 illustrates the cross-entropy loss for each interval in the crown using logistic regression, which assess the confidence of the model's predictions with lower values indicating better model performance. The perfect model has a cross-entropy loss score of 0. It penalizes confident wrong predictions

heavily, ensuring that the model's probability estimates are well calibrated. Here it can be observed that the early intervals have a cross-entropy loss score above 1.2, indicating that the model's probability estimates were far from the true labels, despite showing high ROC AUC score from Figure 5.7. Indicating that while the model makes correct predictions, the confidence in those predictions is not well calibrated. The high cross-entropy loss score for the intervals in early stages of the crown is due to the overconfident predictions, where the model assigning very high probabilities to incorrect predictions. This lead to higher log loss score since the logarithm function penalizes incorrect confident predictions more severely. The overconfident predictions may be attributed to the limited size of the training and test datasets, which may have prevented the model from learning the underlying patterns sufficiently. This could result in poor generalisation and a higher cross-entropy loss score on the test set. It is also possible that the logistic regression model is too simple, and therefore unable to capture the complexities of the data.

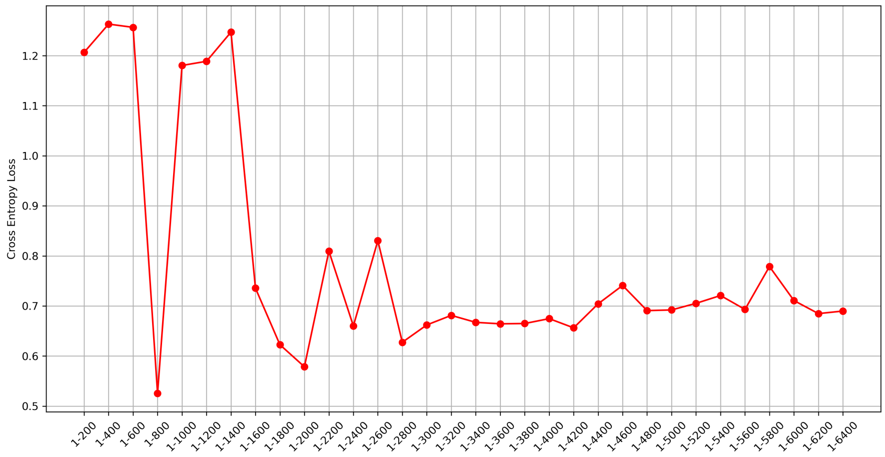


Figure 5.8: Cross Entropy Loss with logistic regression in the crown.

It can be observed that intervals with cross-entropy loss scores in the range of 0.5 to 0.6 indicate superior model performance and well-calibrated probabilities. For example, interval 1-800 exhibits a low cross-entropy loss of approximately 0.5 and a high ROC AUC score, indicating a reliable model for that interval. Following the initial 1,400 intervals, the cross-entropy loss scores stabilise between 0.6 and 0.8, indicating a more consistent model performance. This consistency suggests that the model becomes more reliable in this area of the crown. Therefore, it could be argued that the model should not be trusted until this area is reached.

5.2.2 Random Forest

Figure 5.9 illustrates the ROC AUC score in the crown across intervals utilising random forest. It can be observed that the model is more effective at differentiating between cases with and without structural loss in the early stages of the crown, where the ROC AUC score varies from approximately 0.72 to 0.85 until it reaches interval 1-2200. The ROC AUC score declines slightly until it reaches interval 1-4200 in the crown, with scores rarely exceeding 0.60 until the end of the crown.

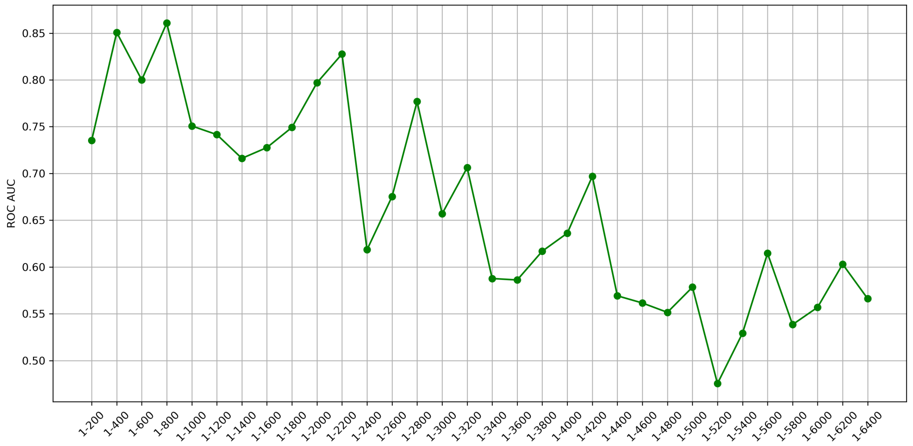


Figure 5.9: ROC AUC scores with random forest in the crown.

From the feature importance analysed in 4.2.1 it was found that the feature ”Energy efficiency indicator” had strongest influence on the predictions, which is the ratio of total heater power to the total diameter increase, highlighting the importance of the energy efficiency of the diameter growth. As stated in [12], the heater power is a crucial parameter to control the growth rate of the ingot and that’s why it is a steady decrease of the heater power is implemented into the crown during the Cz process. This could be the reason for why the early stages of the crown is more complex, where the heater power should reach a steady decrease, straight after the transition from the neck. It is also worth mention that two other features that had large influence on the prediction was ”Diameter increase rate” and ”Total Diameter Increase”, which shows that the shape of the crown is crucial in the early stage.

The favourable ROC AUC scores in the early stages of the crown can be attributed to the complex adjustments required when the diameter expands from a thin neck. The score subsequently declines slightly towards the end of the crown, where changes in parameters become less rapid. In Section 2.2.2

it is explained that the oxygen evaporation is very sensitive to changes in diameter. This sensitivity affects the crystal growth process, as varying oxygen levels can influence the defect structure in the silicon lattice, where the oxygen incorporation into the crystal can also be influenced by factors like seed speed and heater power. However, it is difficult to argue definitively about the impact of oxygen evaporation without concrete evidence in this thesis.

As the diameter stabilizes towards the end of the crown, the rate of oxygen incorporation and other critical parameters become more uniform, leading to a slight reduction in predictive model performance. Additionally, the stabilization phase at the end of the crown often involves fewer adjustments to the seed speed and heater power, resulting in a more predictable but less dynamic environment for the machine learning model to analyze. Consequently, while the early crown stages present a rich dataset of fluctuating conditions for the models to learn from, the later stages offer less variation, slightly diminishing the models' ability to maintain high ROC AUC scores.

Feature importance analysis in section 4.2.1 revealed that the 'Energy efficiency indicator'—a ratio of total heater power to total diameter increase—had the strongest influence on predictions, highlighting the importance of energy efficiency in diameter growth. As noted in [12], heater power is a crucial parameter for controlling the growth rate of the ingot, explaining the steady decrease implemented during the crown phase immediately after transitioning from the neck. Additionally, the 'Diameter increase rate' and 'Total Diameter Increase' also significantly influenced predictions, underscoring the importance of the crown's shape in the early stages.

When closer investigating the three different intervals across the crown, as presented in Table 5.17. The table shows the same trend as shows for the ROC AUC scores in Figure 5.9, where the performance relatively high in the start of crown and slightly decreases towards the end of crown. The interval 1-1800 exhibits excellent performance scores, with an accuracy rate of 73%, a precision rate of 76%, and a recall rate of 64%. These scores demonstrate that the random forest model is highly effective during the initial 30 minutes of the crown. However, as the crown progresses, the model's performance significantly declines, for interval 1-3600 it utilize a accuracy rate of 55%, precision rate of 55% and recall rate on 48%. this indicates that the model is basically randomly guessing the predictions. The model's performance decreases even more when analysing the end of the crown, with a accuracy rate of 51%, precision rate of 50% and recall rate of 40%. This is a low performance of the model and could suggest that the random forest model good performance in the early stages of the crown.

Table 5.17: Performance scores across three intervals in the crown with random forest.

	1-1800	1-3600	1-5400
Accuracy	0.73	0.55	0.51
Precision	0.76	0.55	0.50
Recall	0.64	0.48	0.40

Table 5.18 presents the confusion matrix for interval 1-1800, which illustrates the proportion of correctly and incorrectly predicted ingots from the test dataset. The confusion matrix indicates that 16 ingots with structure loss were correctly predicted, representing a true positive (TP) result. Conversely, 21 ingots were identified as true negatives (TN), demonstrating that they were correctly predicted without structure loss. This interval had relatively great recall rate, which can been seen by the 9 False Negative (FN) cases, where they got predicted as without structure loss but actual was label as structure loss. In the real-world scenario of false negative cases, the model indicates that the Czochralski process should continue. Given that this occurs at an early stage in the process, it is possible that these errors will be identified correctly in a later interval by the model. On the other hand 5 were cases of non-structure loss, yet were predicted as with structure loss, resulting in a False Negative. For this scenario the model would indicate that the remelting process should start, and thereby 5 ingots without structure loss would be remelted. For this reason it could be crucial for the model to have the amount of False Positive cases as low as possible.

Table 5.18: Confusion Matrix for interval (1-1800) with random forest in the crown.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	21	5
Structure loss	9	16

As mentioned above, as the Czochralski process of the crown progresses, the model's performance declines. This is evident in the confusion matrix for interval 1-3600, presented in Table 5.19. In this interval, 12 ingots in the test dataset were correctly predicted as having structural loss (True Positive), and 16 ingots were correctly identified as non-structural loss (True Negative). The confusion matrix shows 13 cases where the model predicted non-structural loss, but the ingots were actually identified with structural loss. This represents a large number of False Negative cases. As shown in Figure 5.9, the performance does not improve significantly after this interval, suggesting that the error may not be correctly identified in a later interval. For the False Positive cases, the model incorrectly predicted 10 ingots as having structural loss.

Table 5.19: Confusion Matrix for interval (1-3600) with random forest in the crown.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	16	10
Structure loss	13	12

Finally, the performance of the model declines further, as evidenced by the confusion matrix for interval 1-5400 in Table 5.20. It can be observed that the model is deficient in its ability to predict instances of structural loss, as demonstrated by the low recall rate in Table 5.17. Where only 10 ingots from the test data were predicted correctly as structure loss, where 16 is predicted correctly as non-structure loss. The outnumber of amount correctly predicted ingots without structure loss, could be explain by that the structure loss occurs later in the shoulder or body for those ingots used for predictions in end of crown.

Table 5.20: Confusion Matrix for interval (1-5400) with random forest in the crown.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	16	10
Structure loss	15	10

Figure 5.10 illustrates the cross-entropy loss scores for each interval in the crown using random forest. It can be observed that the cross-entropy score is relatively low until it reaches interval 1-2400, which reflects excellent model performance in this period with scores always below 0.60. Figure 5.9 also demonstrates that the model exhibits high ROC AUC scores in this period, thereby confirming its ability to make reliable predictions. In particular, during the interval 1-800, the model demonstrated strong predictive performance, with a loss score below 0.45 and an ROC AUC of 0.75.

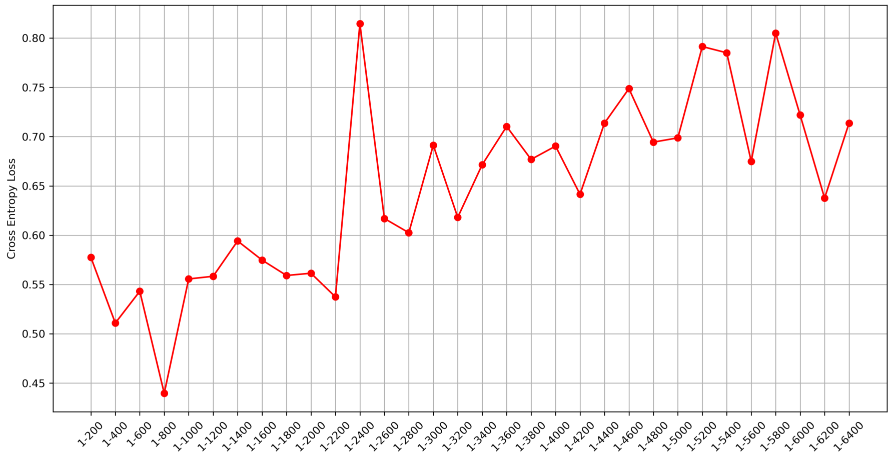


Figure 5.10: Cross Entropy Loss with random forest in the crown.

In interval 1-2400, it is observed a sudden peak which suggest that the model demonstrates poor reliability . The sudden drop in ROC AUC score in Figure 5.9 reflect the model's struggle to make reliable predictions. Furthermore, the model exhibit a stable increase in cross-entropy loss score, inverse propositional to the ROC AUC scores reflecting the model's struggle to make reliable and correct predictions.

5.2.3 Neural Network

Figure 5.11 illustrates the ROC AUC scores across intervals with a neural network model. It can be observed that the model's performance declines consistently across the intervals. The model demonstrates a relatively high level of performance during the initial 70 minutes (interval 1-4200), with a ROC AUC score consistently exceeding 0.60. Until interval 1-2000 the scores varies between 0.7 and 0.9, which show that the model works fine for distinguish between non-structure loss or structure loss in this period of the crown. After interval 1-4400 the model shows a lower performance and larger variation in the scores, where it never above 0.60 and at one point all the way down to 0.3. This suggest that the model find it difficult to distinguish between the two classes in the end of the crown.

It can be observed that the model's performance declines consistently across the intervals. The model demonstrates a relatively high level of performance during the initial 70 minutes (interval 1-4200), with a ROC AUC score consistently exceeding 0.60. Until interval 1-2000, the scores exhibited a range of 0.7 to 0.9, indicating that the model was effective in distinguishing between non-structure loss and structure loss during this period of the crown. Following the 70 minute interval, the performance of the model declines, with greater variation in the scores observed. The scores never exceed 0.60, at one point reaching as low as 0.3. This indicates that the model encounters difficulty in distinguishing between the two classes towards the end of the crown.

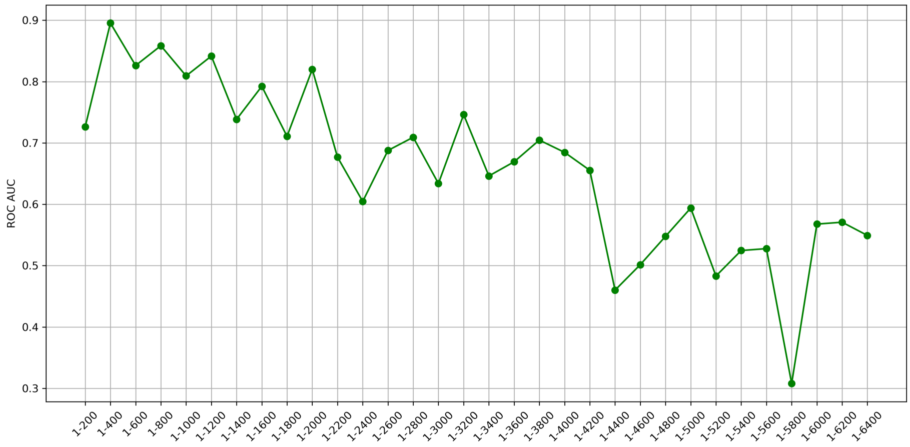


Figure 5.11: ROC AUC scores with neural network in the crown.

For the feature importance analysis for neural network in the crown it was ”Total diameter increase” and ” Diameter Increase rate” that had the larger influence on the predictions. This shows the importance the diameter growth parameters is to distinguish between ingots without structure loss and with structure loss. In a region where the diameter changes occurs rapidly, as done in transition from neck to crown and early stages of crown. The diameter growth is result from the seed speed and heater power, and seed speed should stay relatively constant through the crown and heater power should have steady decrease from the start of the crown to the end, as stated in [12]. Another feature that had a great influence on the model was ”Total heater power”, so when the model shows better performance in the start of the crown, it could be because of the heater power does not have has steady decrease, which makes the diameter to have more variation in the growth.

When taking a closer look at the intervals in Figure 5.11, it has been evaluated the accuracy, precision and recall for the three different interval presented in 5.21. For interval 1-1800, the model achieves an accuracy rate of 65%, a precision rate of 71%, and a recall rate of 48%. The relatively high accuracy and precision combined with a low recall rate indicate that the model finds it easier to predict ingots without structural loss. For ingots labeled as structural loss, the predictions are almost randomly guessed by the model. This could mean that the ingots with structural loss used in the training and test sets have occurrences of structural loss later in the Czochralski process, suggesting that the post-interval should be adjusted closer to the interval.

Table 5.21: Performance scores across three intervals in the crown with neural network.

	1-1800	1-3600	1-5400
Accuracy	0.65	0.63	0.53
Precision	0.71	0.65	0.52
Recall	0.48	0.52	0.48

For interval 1-3600 the accuracy and precision of the model decreases, with accuracy rate of 63% and precision rate of 65%. The recall increases a small amount to 52%, which means that the model’s ability to predict structure loss correctly increases. For the interval between 1 and 5400, all performance scores decreased, with an accuracy rate of 53%, a precision rate of 52%, and a recall rate of 48%. This is approaching the end of the crown, which could indicate that the occurrence of structure loss is concentrated in the shoulder or body region. Consequently, the crowns labelled with structure loss may be equivalent to those without structure loss, making it challenging for the model to distinguish between the two classes.

Table 5.22 presents the confusion matrix for interval 1-1800. It shows the consequence of the weak recall rate with just 12 cases of correctly predicted ingots with structure loss. In addition, 13 ingots were predicted without structure loss when they are label with structure loss, resulting in a False Negative. To have more false negative outcomes than true positive, indicates a weak performance of the predictive model. Although it is a possibility that the model can correct the error in later intervals.

Table 5.22: Confusion Matrix for interval (1-1800) with neural network in the crown.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	21	5
Structure loss	13	12

For the ingots in the test datasets label with non-structure loss, the model shows great predictive abilities, with 21 cases of correctly predicted non-structure loss, known as True Negative. Only 5 ingots were predicted with structure loss but actual labels as without structure loss, suggesting that these would end up to be remelted by trusting the model.

The confusion matrix for interval 1-3600 is presented in Table 5.23, where it can be observed that the recall rate has increased slightly. This is evidenced by the fact that 13 ingots were correctly predicted with structure loss, while 12 ingots were identified as False Negatives. On the other hand the decrease of accuracy and precision is also visible, with 19 ingots identified as True Negative and 7 ingots identified as False Positive. These results show that the model has a weak performance in this interval and should probably not be trusted in a real-life scenario.

Table 5.23: Confusion Matrix for interval (1-3600) with neural network in the crown.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	19	7
Structure loss	12	13

Table 5.24 presents the confusion matrix for interval 1-5400 in the crown. It is evident that the model continues to perform poorly in the end region of

the crown where the majority of cases exhibit equivalent results. As previously stated, this may be attributed to the fact that the parameters are more stable in the middle and end regions of the crown, and that the occurrence of structure loss is concentrated in the shoulder or body region.

Table 5.24: Confusion Matrix for interval (1-5400) with neural network in the crown.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	15	11
Structure loss	13	12

Figure 5.12 illustrates the cross-entropy loss scores across intervals using the neural network model in the crown. The average loss score is 1.297, with a standard deviation of 0.965, indicating a high degree of variability in the model's performance across different intervals. When the loss score is below 1, the model utilises reliable predictions, as evidenced by three periods in the crown. The initial period, spanning from interval 1-400 to 1-1200, exhibits low cross-entropy loss scores and relatively high ROC AUC scores above 0.8, as illustrated in Figure 5.11.This indicates that the model is performing well in this period of the crown. Once more, the model demonstrates low loss scores between intervals 1-1600 and 1-2200. Nevertheless, it exhibits slightly lower ROC AUC scores, indicating a higher number of mispredictions, although these are not as numerous as might be expected. Furthermore, for the period between intervals 1-3200 and 1-4000, the model demonstrated high loss scores and ROC AUC scores approaching 0.7, indicating reliable performance with a few mispredictions.

Intervals with very high cross-entropy loss scores are often due to the model making overconfident predictions, which are heavily penalized by the cross-entropy function. This issue may arise from the neural network model being too complex, making it prone to overfitting, especially with a small dataset. Overfitting occurs when the model closely fits the training data, including noise, and thus fails to generalize effectively to new, unseen data. Such overfitting can result in high confidence in incorrect predictions. Since the cross-entropy loss function heavily penalizes these errors, even a few mispredictions can lead to a significantly high loss score. Additionally, small training sets may be insufficient to capture the underlying patterns in the data.

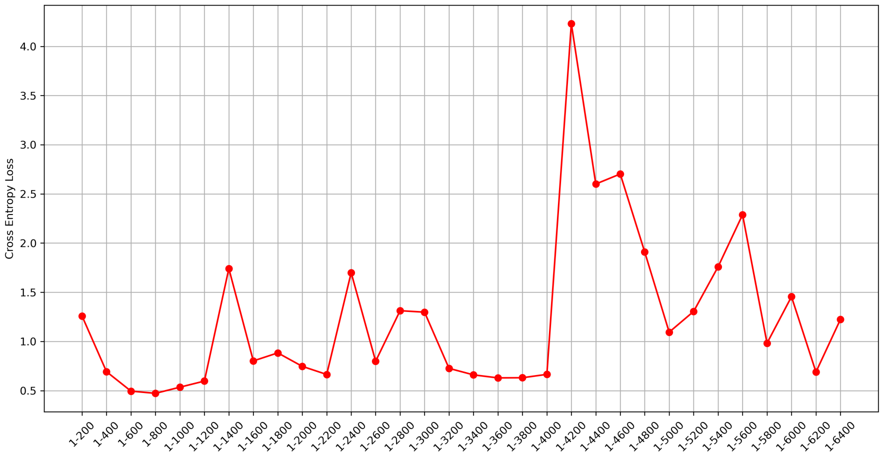


Figure 5.12: Cross Entropy Loss with neural network in the crown.

5.3 Shoulder

A total of 131 ingots were included in the training datasets, while 57 ingots were included in the test datasets. Given that the shoulder region is significantly shorter than the other regions, the interval was used with steps of 50 seconds to reach 500 seconds of the Czochralski of the shoulder. This was done in order to approximate the point at which most shoulders transition to the body region.

5.3.1 Logistic Regression

Figure 5.13 illustrates the ROC AUC scores across intervals in the shoulder region, using logistic regression as a predictive model. For the first three intervals, the model shows a relatively high score with a peak of 0.70 in interval 1-150. In interval 1-200, the model shows a significant decline in performance, dropping to around 0.60. There is a subsequent period of stability until the end of the shoulder region. The cumulative interval may introduce uncertainty into the model. For instance, the first interval (1-50) only contains 50 datapoints from the start of the shoulder-pulling process. As the interval extends, it contains more data on the process, potentially leading to more stable performance. However, the increase in performance in the early stages may be due to the rapid physical changes that occur during the transition from the crown to the shoulder. During this transition, the diameter will be reduced slightly. This can be achieved by increasing the seed speed, although it is crucial to ensure that the seed speed is not too high, as this could result in the diameter growing inwards during body growth. This is also discussed in section 3.2.

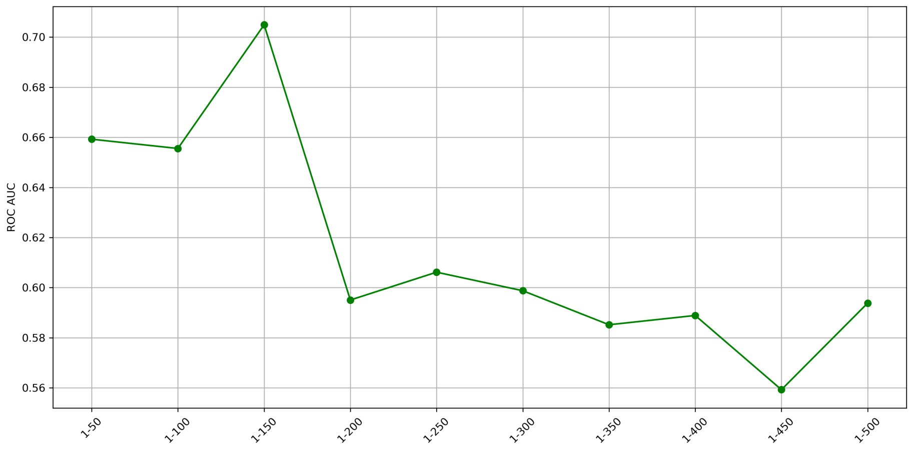


Figure 5.13: ROC AUC scores with logistic regression in the shoulder.

When performing the feature importance analyse in section 4.2.1 is was found that ”Total heater power” and ”Seed speed Variability” were the two features with most influence on the predictions. These features indicates of the energy involved and the consistency of the seed speed adjustments for the specific interval. This can show that for the ingots with structure loss contains more variability in the seed speed, specially in the early stages of the shoulder. This could then cause the diameter to grow inwards, which is critical cause of dislocations in the ingot. The ”Total heater power” feature can indicate that the ingots with detection of structure loss has higher energy involved, and as consequence a higher temperature in the crucible, which might lead to the diameter growing inwards.

Table 5.25 presents performances scores for three different interval across the shoulder of the ingot. Interval 1-150 demonstrates good performance with a accuracy rate of 65%, precision rate of 60% and recall rate of 78%. The indicates that the model would perform stable predictions, only downside would be the precision score of only 60%, which means that the model will predict a relatively high amount of ingots as False Positive. For interval 1-300, it can be observed that the performance of the model declines with a accuracy rate of 54%, precision rate 51% and recalls rate of 70%. These scores indicates that the model performs poorly in this interval. The performance of the model declines even more in interval 1-450, with accuracy rate of 49%, precision rate of 47% and recall rate of 56%. These performance scores indicates that the model would not beneficial to trust in the in this interval.

Table 5.25: Performance scores across three intervals in the shoulder with logistic regression.

	1-150	1-300	1-450
Accuracy	0.65	0.54	0.49
Precision	0.60	0.51	0.47
Recall	0.78	0.70	0.56

A detailed examination of the performance scores presented in Table 5.25 will be conducted, with a particular focus on the confusion matrix for each interval. Table 5.26 illustrates the confusion matrix for interval 1-150, where the model predicted 21 ingots correctly as structure loss, also known as true positive. With regard to the correctly predicted ingots without structure loss, there were 16 cases of true negatives. The model exhibited only 6 instances of false negatives, where the predicted outcome was the without structure loss label, yet the actual outcome exhibited structure loss. This would suggest that the model would permit the Czochralski process to continue, with these 6 ingots remaining undetected by the model within this interval. It is also possible that the false negative cases could be identified in a subsequent interval. Conversely, there were 14 instances of false positives, indicating that the model may have problem identifying these ingots without structure loss. If the model were to be trusted in this interval, these 14 ingots would be remelted, potentially leading to unnecessary costs.

Table 5.26: Confusion Matrix for interval (1-150) with logistic regression in the shoudler.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	16	14
Structure loss	6	21

From Table 5.25 it was found that the interval 1-300 had declines in the performance, and this is decline is also displayed in the confusion matrix of the model in Table 5.27. Where the model predicted 19 ingots correctly with structure loss, also known as True Positive. For True Negative cases it was only 12 ingots that got correctly predicted without structure loss and 8 cases of False Negative. The False Negative exhibited a considerable number of ingots (18) with structure loss, yet the actual label was without structure loss. This quantity of False Positive could be the determining factor in the model's lack of trustworthiness within this interval.

Table 5.27: Confusion Matrix for interval (1-300) with logistic regression in the shoulder.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	12	18
Structure loss	8	19

For the interval in the end stages of the shoulder, it is presented in Table 5.28. Here the it is found that the model performs extremely poor and it would not be beneficial to trust the model in this interval with logistic regression model.

Table 5.28: Confusion Matrix for interval (1-450) with logistic regression in the shoulder.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	13	17
Structure loss	12	15

Figure 5.14 displays the cross-entropy loss scores across intervals for the logistic regression model, with an average loss score of approximately 0.68 and a standard deviation of 0.02, indicating moderate variability in model performance. Intervals with low cross-entropy scores typically have better calibrated models, leading to more reliable predictions. The model demonstrates more reliable predictions in the first three intervals, where it also achieves higher ROC AUC scores, particularly in interval 1-150, as shown in Figure 5.13. After interval 1-150, the model records high loss scores and low ROC AUC scores, indicating poor model calibration and prediction accuracy. Consequently, it can be argued that in the later periods of the shoulder, the logistic regression model struggles to provide reliable confidence scores and balanced predictions.

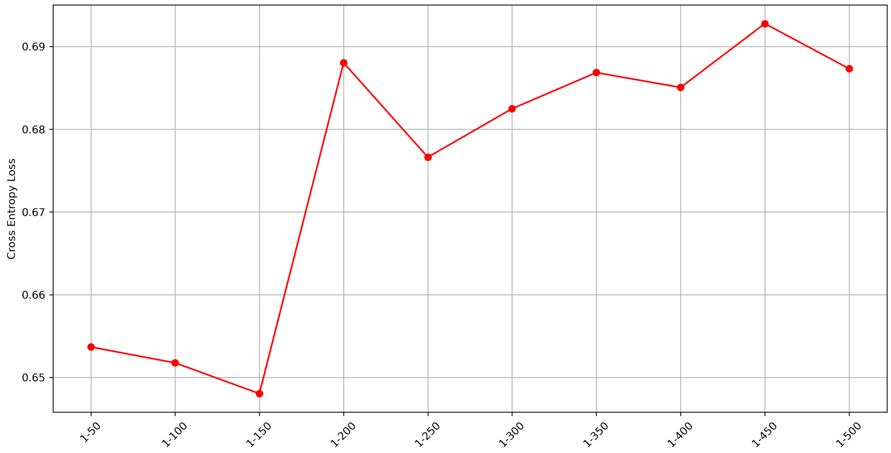


Figure 5.14: Cross Entropy Loss with logistic regression in the shoulder.

5.3.2 Random Forest

Figure 5.15 illustrates the ROC AUC score across intervals using a Random Forest model. The scores vary significantly, ranging from 0.55 to 0.65, which suggests that the model is almost randomly guessing the predictions. This poor performance could be explained by the uncertainty in the data, as random forest models require more data in the training and test sets to perform effectively. The large variation in the scores could be due to the fact that each interval contains a different number of datapoints. For instance, interval 1-50 only contains 50 datapoints, while interval 1-450 contains 450 datapoints. Therefore, later intervals could have higher certainty due to the increased amount of data.

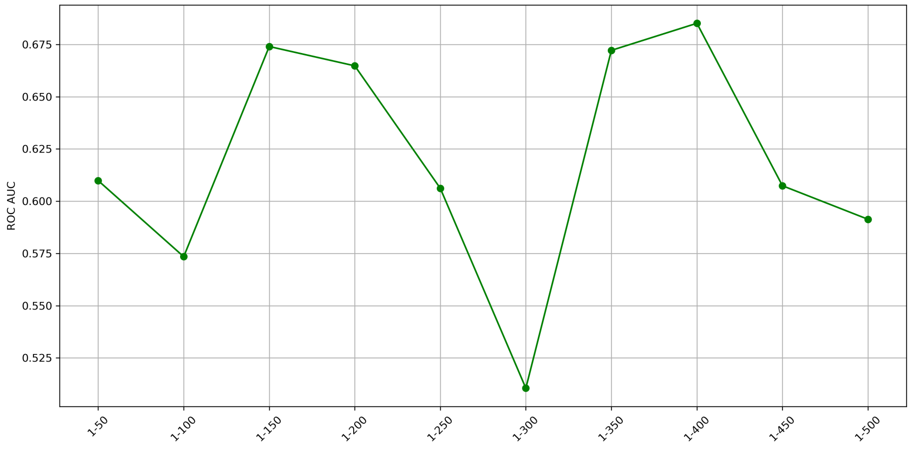


Figure 5.15: ROC AUC scores with random forest in the shoulder.

Figure 5.29 shows accuracy, precision, and recall scores for three intervals across the shoulder. It is evident that the Random Forest model has poor overall performance in the shoulder, with accuracy and precision rates for all three intervals around 50%. Only in intervals 1-150 and 1-450 does it show a relatively higher recall rate of 67% and 63%, respectively, which indicates a better ability to correctly predict ingots labeled with structural loss. The poor performance suggests that the random forest algorithm struggles to make accurate predictions with this type of data. This late in the Czochralski process, it is a possibility that the operator can also detect structural loss. Therefore, it may be more beneficial to rely on the operator's judgment for detecting structural loss rather than trusting the machine learning model in this context.

Table 5.29: Performance scores across three intervals in the shoulder with random forest.

	1-150	1-300	1-450
Accuracy	0.54	0.53	0.56
Precision	0.51	0.50	0.53
Recall	0.67	0.41	0.63

Table 5.34 presents the confusion matrix for interval 1-150. The results from the predictions made on the test set show a high recall rate with 18 True Positive cases and only 9 False Negative cases. However, a concerning aspect of this confusion matrix is the high number of False Positives, with 17 cases. This means that if the model were to be trusted, 17 ingots would be unnecessarily remelted.

Table 5.30: Confusion Matrix for interval (1-150) with random forest in the shoulder.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	13	17
Structure loss	9	18

Table 5.31 presents the confusion matrix for interval 1-300, indicating that the model has difficulty accurately predicting structural loss. With just 11 ingots correctly identified as having structural loss (True Positives), the model also exhibits a significant number of False Negatives (11). This suggests that the model's predictions should be treated with caution.

Table 5.31: Confusion Matrix for interval (1-300) with random forest in the shoulder.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	19	11
Structure loss	16	11

The confusion matrix for interval 1-450 in Table 5.32, also indicates that the model's predictions should be treated with caution. The predictions made for ingots labeled without structural loss are only 50% correct, meaning the model is essentially guessing randomly.

Table 5.32: Confusion Matrix for interval (1-450) with random forest in the shoulder.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	15	15
Structure loss	10	17

Figure 5.16 displays the cross-entropy loss scores across intervals for the random forest model, with an average loss score of approximately 0.69 and a standard deviation of 0.05, indicating moderate variability in model performance. Generally, the model provides reliable predictions until it reaches intervals 1-300 and 1-500, which exhibit marked peaks in loss scores. These intervals also show poor performance as evaluated in Figure 5.15. This pattern suggests that the model's predictions are unreliable and poor in these specific intervals.

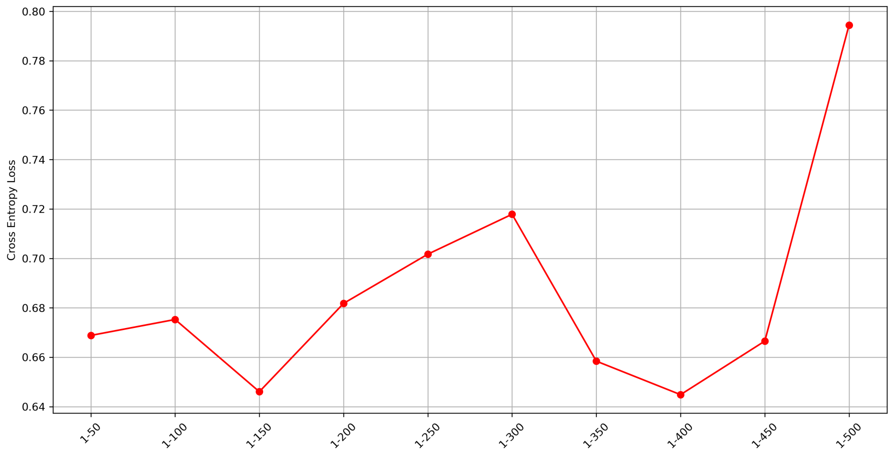


Figure 5.16: Cross Entropy Loss with random forest in the shoulder.

5.3.3 Neural Network

Figure 5.17 presents the ROC AUC scores across intervals for the neural network model. It is evident that the scores vary drastically. The only noteworthy performance is in interval 1-150, where the ROC AUC is just above 0.70. The feature importance analysis revealed that the ”Total heater power” feature had the greatest influence on the predictions. This suggests that the total heater power is higher for ingots with structural loss, where it should increase slightly to decrease the diameter at the start of the shoulder. As a result, there is an improvement in the performance of the predictions in interval 1-150.

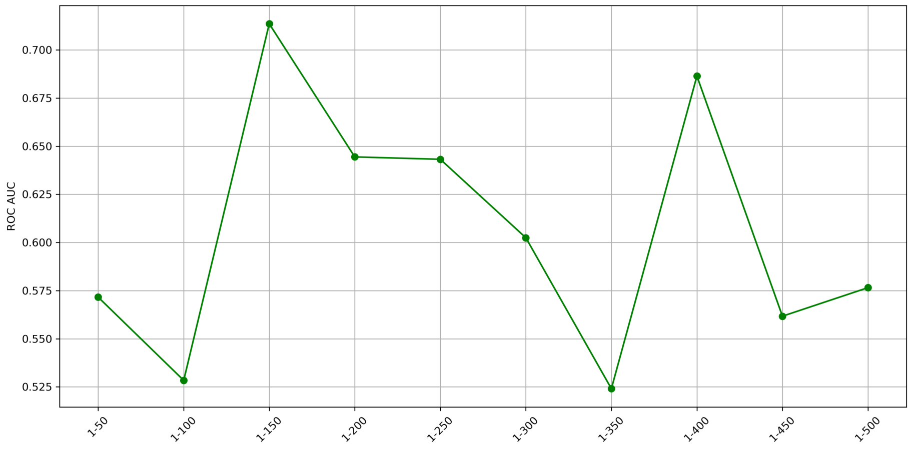


Figure 5.17: ROC AUC scores with neural network in the shoulder.

Table 5.33 presents the performance scores for three different intervals across the shoulder. Furthermore, interval 1-150 is identified as the interval with the most notable improvement in performance, with an accuracy rate of 68%, a precision rate of 63%, and a recall rate of 82%. The remaining two intervals yielded scores of approximately 50%, which is comparable to the performance of a random guess.

Table 5.33: Performance scores across three intervals in the shoulder with neural network.

	1-150	1-300	1-450
Accuracy	0.68	0.56	0.51
Precision	0.63	0.53	0.48
Recall	0.82	0.63	0.56

The confusion matrix for the three interval categories is presented below. The only matrix that demonstrates optimal performance is for interval 1-150 in Table 5.34. This matrix has 22 cases of true positive and 17 cases of true negative. The model only predicts 5 ingots as false negatives, while it predicts 13 ingots as false positives. This is a considerable number, particularly when one considers that the modeFor the confusion matrices presented in Table 5.35 and 5.36 it is observed that the performance for both intervals is very poor. This could indicate random guessing in this area of the shoulder.

Table 5.34: Confusion Matrix for interval (1-150) with neural network in the shoulder.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	17	13
Structure loss	5	22

Table 5.35: Confusion Matrix for interval (1-300) with neural network in the shoulder.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	15	15
Structure loss	10	17

Table 5.36: Confusion Matrix for interval (1-450) with neural network in the shoulder.

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	14	16
Structure loss	12	15

Figure 5.18 displays the cross-entropy loss scores across intervals for the neural network model, with an average loss score of approximately 2.82 and a standard deviation of 1.71. This indicates high variability and substantial uncertainty in the model's predictions. Overall, the model exhibits high loss scores, suggesting poor robustness and reliability in prediction.

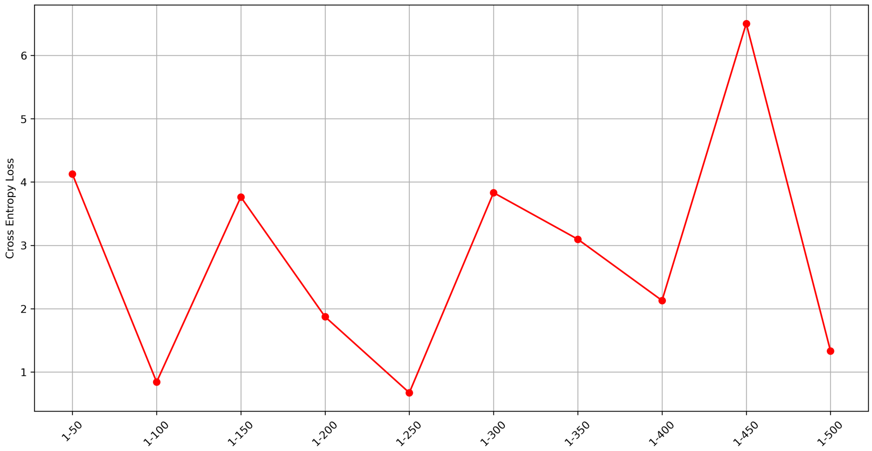


Figure 5.18: Cross Entropy Loss with neural network in the shoulder.

5.4 Body

A total of 152 ingots were included in the training datasets, while 66 ingots were included in the test datasets. All three machine learning models struggled to predict structural loss. For simplicity, only the results from the logistic regression model will be presented.

5.4.1 Logistic Regression

Figure 5.19 illustrates the ROC AUC scores across intervals in the body using logistic regression, where it can be observed that the model hovers around a value of 0.50 until it increases towards the end of the body.

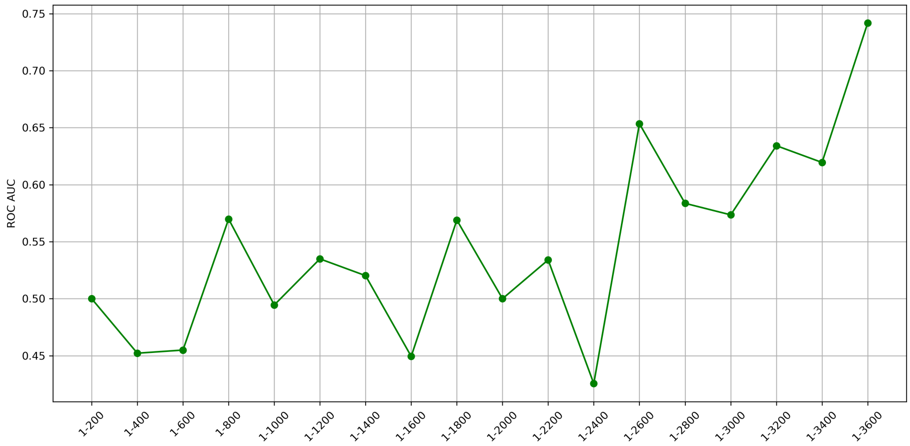


Figure 5.19: ROC AUC score for body with logistic regression.

Table 5.37 presents the performance scores for selected intervals, indicating poor performance in the body. It is evident that the model struggles to make accurate predictions for ingots with structural loss, as demonstrated by the significantly low recall rates across all three intervals. This issue is further highlighted in the confusion matrices presented in Tables 5.38, 5.39, and 5.40. The poor performance may be attributed to the lack of significant changes in the body, where the parameters operate under fixed settings. Similar results were also found with random forest and neural network models; therefore, their results will not be presented.

Table 5.37: Performance scores the body for logistic regression.

	1-1000	1-2000	1-3000
Accuracy	0.5	0.49	0.64
Precision	1	0	1
Recall	0.03	0	0.29

Table 5.38: Confusion Matrix on test datasets for interval (1-1000)

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	32	0
Structure loss	33	1

Table 5.39: Confusion Matrix on test datasets for interval (1-2000)

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	32	0
Structure loss	34	0

Table 5.40: Confusion Matrix on test datasets for interval (1-3000)

Actual \ Predicted	**Non-Structure loss**	**Structure loss**
Non Structure loss	32	0
Structure loss	24	10

5.5 Comparing Models

The objective of this thesis is to ascertain which machine learning model performs best of the four region of the ingot. The machine learning model that demonstrates the best performance will subsequently be used in the time-saving model to estimate the time savings associated with remelting. The models will be compared only for predictions of the neck and crown, due to the poor performance and large variations and uncertainties for all three models in the shoulder region. Due to the cross-validation conducted on the training sets, the scores will differ slightly from those presented in sections 5.1, 5.2 and 5.3. However, the overall trends will remain consistent.

5.5.1 Neck

Figure 5.20 illustrates the comparison of ROC AUC scores for all three models. All three models demonstrate a low score at the start of the neck, and then they perform more stably after interval 1-600, except for the neural network model, which shows large variations across the intervals. Logistic regression indicates the most stable performance and also the overall highest ROC AUC scores.

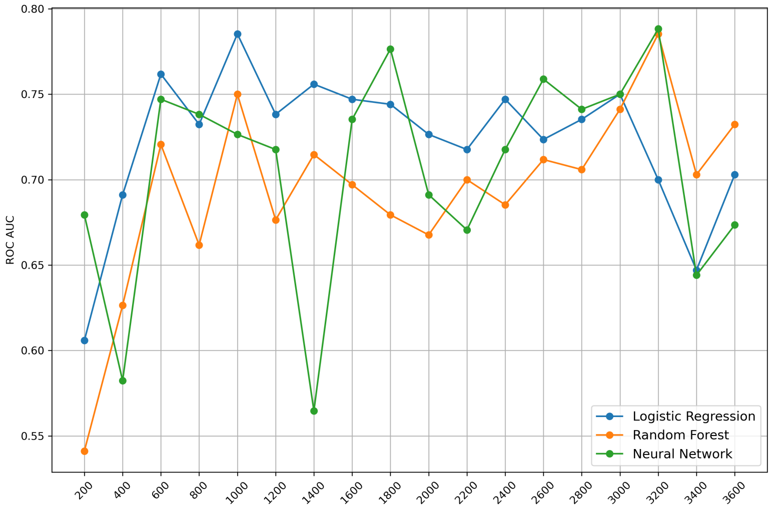


Figure 5.20: Comparison of the ROC AUC score for all models in the neck.

In order to demonstrate the diversity of the models, different intervals will be selected for closer analysis than those selected in section 5.1. For interval 1-1000 presented in Table 5.41, the neural network model demonstrates the best accuracy and precision rate of 68% and 77%. The recall rate is employed by the logistic regression model, which indicates that logistic regression has the greatest capacity to predict those actual labels with structure loss correctly, which makes fewer cases of False Negative but these can be detected in a later interval. However, as discussed in Section 3.2, to avoid the remelting of ingots that actually contained non-structure loss, which would be remelted and potentially lead to unnecessary costs, the precision score is given greater weight than the recall score.

Table 5.41: Performance scores for interval 1-1000 for all three models.

	Logistic Regression	Random Forest	Neural Network
Accuracy	0.57	0.65	0.68
Precision	0.52	0.61	0.62
Recall	0.82	0.65	0.77

From Table 5.42 it is evident that the neural network displays a poor performance, as it is unable to predict any of the actual labels correctly, despite the use of structure loss. For this interval, the random forest model exhibited the highest accuracy and precision, with values of 65% and 61%, respectively.

Table 5.42: Performance scores for interval 1-2000 for all three models.

	Logistic Regression	Random Forest	Neural Network
Accuracy	0.60	0.65	0.54
Precision	0.54	0.61	0
Recall	0.88	0.65	0

In the later stages of the neck, the random forest model displays the highest performance, as evidenced by Table 5.43. This may be attributed to the ensemble learning of random forest, whereby multiple individual models are combined to create a more robust and accurate model. Given the limited data set on which the model is based, the ensemble learning approach may be able to compensate for any model weaknesses.

Table 5.43: Performance scores for interval 1-3000 for all three models.

	Logistic Regression	Random Forest	Neural Network
Accuracy	0.62	0.73	0.68
Precision	0.57	0.68	0.63
Recall	0.77	0.77	0.71

5.5.2 Crown

Figure 5.21 presents the comparison of the ROC AUC scores for all three models in the crown. The models shows a similarly trend, with high performance in early stages of the crown and the performance declines steadily towards the end of the crown. Logistic regression has the highest peak of 0.9 in interval 1-1000, but further into the crown it looks like the random forest model overall has the highest ROC AUC score after interval 1-2600.

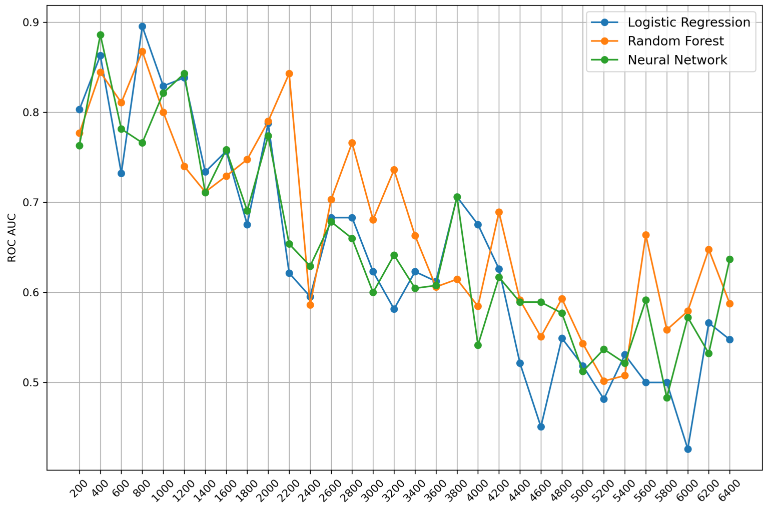


Figure 5.21: Comparison of the ROC AUC score for all models in the crown.

To demonstrate the diversity of the models, different intervals will be analyzed more closely than those selected in Section 5.2. Table 5.44 presents the performance scores for interval 1-800, where it is evident that the neural network exhibits the weakest performance among the models. Logistic regression, on the other hand, shows slightly better precision, resulting in fewer cases of False Positives. Consequently, ingots actually without structure loss would not be incorrectly subjected to remelting.

Table 5.44: Performance scores for interval 1-800 for all three models.

	Logistic Regression	Random Forest	Neural Network
Accuracy	0.75	0.75	0.67
Precision	0.88	0.80	0.65
Recall	0.56	0.64	0.68

For interval 1-2800 it is clear that the random forest model utilize the best performance, as shown in Table 5.45. Random forest contains the highest scores for all three performance scores.

Table 5.45: Performance scores for interval 1-2800 for all three models.

	Logistic Regression	Random Forest	Neural Network
Accuracy	0.67	0.73	0.69
Precision	0.68	0.72	0.67
Recall	0.60	0.72	0.72

Table 5.46 presents that the random forest model also works better in this interval, only recall score is higher for both logistic regression and neural network. Here both showcase very similarly scores.

Table 5.46: Performance scores for interval 1-4800 for all three models.

	Logistic Regression	Random Forest	Neural Network
Accuracy	0.53	0.61	0.55
Precision	0.52	0.63	0.54
Recall	0.60	0.48	0.60

In general, the random forest model demonstrated better performance on the unseen data used in the crown. It is challenging to ascertain the precise reasons for the slightly lower performance of the logistic and neural networks compared with the random forest. However, it is notable that both models utilise the sigmoid function, which calculates the predicted probability of the outcome. The sigmoid function is a widely used mathematical function due to its combination of non-linear and differentiable properties. The article [52] states that the use of the sigmoid function may precipitate a vanishing gradient problem, which arises when the output is 0 or 1, and the gradient of these inputs is close to zero. This can result in an unstable training process and a reduction in the weight of the network, thereby impeding its capacity to learn. It can be argued that the combination of a small amount of data with this approach may be the reason for the better performance of ensemble learning with random forest on this kind of data.

5.6 Time-Saving Model

The time-saving model, using the random forest model, is based on its overall performance presented in sections 5.2 and 5.5.2. The accuracy threshold was set at 70%, achieved during four intervals. For intervals that meet this threshold are all represented the early stages of the crown, consistent with results in section 5.2.2. The remelting time was calculated by Equation 4.1 where the volume of the ingot at where the model indicated remelting is multiplied with the remelting rate. For the same ingot the remelting time is calculated by the volume at where the operator detected structure loss, multiplied with the same remelting rate constant. This analysis enables to compare the time saved and time lost across interval above the accuracy threshold.

Based on calculations using the method described in Section 4.6, an estimate of the average remelting time for intervals exceeding the accuracy threshold was determined to be 10 minutes. Time savings were calculated by taking the difference between when the model achieved an accuracy above 70% and the remelting time detected by the operator, including the period of crystal growth from when the model showed high performance to when structural loss was detected by the operator. It was found that with one correct prediction, it is possible to save approximately 60 minutes.

A model with a minimum accuracy of 70% suggests incorrect predictions (false positives and false negatives) 30% of the time. For simplicity, it is assumed that the time taken to correct false negatives is minimal, as errors might be rectified in subsequent intervals. Therefore, the focus is on the time lost due to false positives, where remelting was predicted but not actually needed. The calculation of time lost due to unnecessary remelting operations was performed by determining the average time taken for a remelting process and then multiplying this by the number of false positives. The calculation of time saved was done by multiplying the count of true positives by the time saved per correct prediction (60min) to get the total time saved for each interval. It is important to note that this analysis was performed on a test set containing 51 ingots.

Table 5.47 presents the accuracy of the model for different intervals along with the number of false positives and the corresponding time lost due to these false positives. The accuracy of the model ranges from 0.73 to 0.88 across different intervals. The time lost due to false positives varies between 30 and 60 minutes per interval. This time lost is significant as it represents the operational inefficiencies caused by incorrect model predictions.

Table 5.47: Time-saving model results showing the interval, accuracy, false positives, and time lost for each interval.

Interval	Accuracy	False Positives	Time Lost (hours)
(1, 400)	0.88	3	0.5
(1, 600)	0.78	4	0.67
(1, 800)	0.75	4	0.67
(1, 1000)	0.76	3	0.5
(1, 1800)	0.73	6	1
(1, 2000)	0.73	6	1

In table 5.48 displays the number of true positives and the corresponding time saved due to correct predictions by the model. The time saved is substantial, ranging from 16 to 22 hours per interval. This highlights the potential benefits of the model in identifying the need for remelting accurately, thereby saving significant operational time.

Table 5.48: Time-saving model results showing the interval, accuracy, true positives, and time saved for each interval.

Interval	Accuracy	True Positives	Time Saved (hours)
(1, 400)	0.88	22	22.0
(1, 600)	0.78	18	18.0
(1, 800)	0.75	16	16.0
(1, 1000)	0.76	16	16.0
(1, 1800)	0.73	17	17.0
(1, 2000)	0.73	17	17.0

Table 5.49 provides the net time saved by subtracting the time lost due to false positives from the time saved due to true positives. The net time saved

ranges from 15.3 to 21.5 hours per interval. This indicates that despite the time lost due to false positives, the model still provides a significant net benefit in terms of time saved. At the most it is possible to save 21.5 hours while

Table 5.49: Time-saving model results showing the interval, accuracy, and net time saved for each interval.

Interval	**Accuracy**	**Net Time Saved (hours)**
(1, 400)	0.88	21.5
(1, 600)	0.78	17.3
(1, 800)	0.75	15.3
(1, 1000)	0.76	15.5
(1, 1800)	0.73	16.0
(1, 2000)	0.73	16.0

The results indicate that the time-saving model effectively predicts the need for remelting with a reasonable degree of accuracy. Although false positives lead to some time loss, the overall net time saved is substantial. For example, in the interval 1-400, the model achieves an accuracy of 88%, resulting in a net time saving of 21.5 hours. This demonstrates that the model can significantly reduce operational inefficiencies by making timely and accurate predictions in the early stages of the crown. Even for intervals with accuracy's close to the threshold, a large net time saving of 16 hours is observed, suggesting that a threshold of 70% yields excellent results and could potentially be lowered to include more intervals. This analysis indicates that it is possible to saved around 16-21.5 hours when testing the model on 51 ingots. Thereby, increasing number of ingots could increase the time saved to maximize operational benefits.

The estimated times for remelting (10 minutes) and time saved per correct prediction (60 minutes) are based on assumptions. These estimates may vary significantly in real-world scenarios, affecting the accuracy of the calculated time saved and lost. The model does not consider the stabilization time required after a prediction is made. In reality, there may be additional delays and operational steps that impact the overall time saved.

5.7 Summary of results

In summary, some of the objectives for this thesis have been accomplished, while others have not. This will be summarized in the following list:

- **Neck Region:** All three models—logistic regression, random forest, and neural networks—demonstrated improved stability in later intervals. The 'Seed speed heater power ratio' was identified as the most important feature influencing the predictions. Among these, the random forest model exhibited the highest predictive performance in the neck region, characterized by better precision and fewer false positives compared to the other models.

- **Crown Region:** The models showed good results in the early stages but a decline in accuracy towards the end of the crown was seen. Key features influencing predictions were the 'Energy efficiency indicator' and 'Diameter increase rate.' Among them, the random forest model consistently demonstrated superior performance, maintaining higher ROC AUC scores throughout the crown.

- **Shoulder Region:** All models struggled to predicting structural loss in the shoulder region. None of the models demonstrated notable better performance, indicating the need for further refinement and additional features specific to this region.

- **Body Region:** Similar to the shoulder, the body region posed difficulties for all models. The lack of significant parameter changes in this region likely contributed to the poor performance of the models.

- **Time-Saving Model:** Using data from the random forest model, the time-saving model attained accuracy above the 70% threshold in four intervals. By analyzing False Positives and True Positives extracted from these intervals, it was possible to calculate the time lost due to incorrect predictions and the time saved due to correct predictions. Despite losing 30-60 minutes per interval to False Positives, the net time saved was substantial, ranging from 16 to 21.5 hours per interval when testing 51 ingots in the model.

Chapter 6

Conclusions

This thesis has explored three machine learning models, logistic regression, random forest, and neural networks, to predict structural loss in different regions of the ingot during the Czochralski process. The regions analyzed—neck, crown, shoulder, and body—each presented unique challenges and yielded distinct performance outcomes. It was found that including datasets where structural loss was detected closer to the investigated interval, enhanced by a post-interval, improved the accuracy of the predictions.

The neck and crown regions demonstrated the most promising results. In particular, the random forest model consistently outperformed the other models, exhibiting high accuracy, precision, and recall scores. This suggests that the ensemble learning approach of random forest effectively captures the complexities and variabilities inherent in the early stages of the Czochralski process.

In the neck region, the random forest model achieved the highest accuracy and precision, especially in the later intervals, indicating its robustness in predicting structural loss as the Czochralski process progresses. Common for all models in the neck was that the feature ”Seed heater power ratio” was one of the most influential parameter on the predictions. This suggest that the balance between seed speed and heater power is critical to control, and also to measure accurately to be able to determine structure lloss early. The logistic regression model, showed stability and decent performance in the early intervals, did not match the overall efficacy of the random forest model. The neural network model, despite its potential, exhibited considerable variations and inconsistencies, highlighting the need for further refinement and possibly larger training datasets.

The crown region further validated the superiority of the random forest model, which maintained higher ROC AUC scores and better overall performance metrics compared to the other models, particularly in the early to mid-stages of the crown. While the logistic regression model showed strong initial performance, it struggled to maintain accuracy and precision deeper into the crown. The neural network model again displayed variability, indicating that further optimization is necessary. The diameter growth and its dependency on heater power changes were significant factors influencing predictions, suggest the shape of the crown is important specially in the early stages.

In the shoulder and body regions, all three models struggled to provide reliable predictions. The high variability and uncertainty in these areas suggest that the current modeling approaches may be insufficient, or that additional for these regions, more specific features and parameters need to be incorporated to improve predictive accuracy.

The time-saving model using the random forest algorithm has shown significant practical benefits. By maintaining an accuracy threshold of 70%, four intervals achieved this, all representing early crown stages. Despite some time lost to false positives, testing the model on 51 ingots for the four intervals resulted in net time savings ranging between 16 and 21.5 hours.

The research carried out in this thesis highlights the potential of machine learning models, to enhance the Czochralski process by enabling the early detection of structural loss. These findings could lead to significant time savings, and consequently saving money and energy, by discover the need for remelting early in the Czochralski process.

6.1 Future Work

There are several suggestions for future work. Firstly, a larger amount of data for ingots would increase both the training and test sets. It would be interesting to see how the model performs with larger datasets, as this could also reduce some of the uncertainties associated with the somewhat limited number of datasets used in this work, if the predictions would have the same trends. With larger datasets it would also be possible to decrease the post-interval and see how the model would be behave.

Different calculations of features could also be further investigated, to see if the performance of the model improves. A options could also to include more parameters from the Czochralski silicon process, such as weight of the ingot, crucible rotation and pressure in the crucible, to investigate the influence they would have on the predictions.

It would not be possible to treat the data as time-series with logistic regression, random forest and MLPClassifer models from Scikit-learn library, where the risk of overfitting would be high. Therefore, it would also be interesting to see how the model would behave if the data was treated as time-series. This would make it possible to get a better understanding of the patters in the data. To achieve this it would be necessary to use other machine learning models, for instance Long short-term memory (LSTM).

Bibliography

[1] NorSun AS. Norsun, 2024. URL `https://www.norsun.no/`.

[2] BYJU'S. P-n junction. `https://byjus.com/physics/p-n-junction/`, 2024. Accessed: 2024-05-14.

[3] Guilherme Gaspar, Antoine Autruffe, and Mário Pó. *Silicon Growth Technologies for PV Applications*. 05 2017. ISBN 978-953-51-3159-5. doi: 10.5772/intechopen.68351.

[4] S. Meroli. Czochralski process vs float zone: two growth techniques for mono-crystalline silicon. URL `https://meroli.web.cern.ch/Lecture_silicon_floatzone_czochralski.html`. [Online; accessed October 4, 2023].

[5] MatLab. Machine learning in matlab. `https://www.mathworks.com/help/stats/machine-learning-in-matlab.html`.

[6] Niklas Donges. Random forest: A complete guide for machine learning. `https://builtin.com/data-science/random-forest-algorithm`.

[7] Ergün Akgün and Metin Demir. Modeling course achievements of elementary education teacher candidates with artificial neural networks. *International Journal of Assessment Tools in Education*, 5, 01 2018. doi: 10.21449/ijate.444073.

[8] geeks for geeks. Artificial neural networks and its applications. `https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/`.

[9] scikit-learn Developers. Cross-validation: evaluating estimator performance. `https://scikit-learn.org/stable/modules/cross_validation.html`, Accessed: 2024-05-05.

[10] International Energy Agency. The world needs more diverse solar panel supply chains to ensure a secure transition to net zero emissions, 2022. URL https://www.iea.org/news/the-world-needs-more-diverse-solar-panel-supply-chains-to-ensure-a-secure-transition-to-net-zero-emissions.

[11] Ben McWilliams, Simone Tagliapietra, and Cecilia Trasi. Smarter european union industrial policy for solar panels, 2024. URL https://www.bruegel.org/policy-brief/smarter-european-union-industrial-policy-solar-panels.

[12] Olli Anttila. Czochralski growth of silicon crystals. In Markku Tilli, Mervi Paulasto-Kröckel, Teruaki Motooka, Veikko Lindroos, Veli-Matti Airaksinen, Sami Franssila, and Ari Lehto, editors, *Handbook of Silicon Based MEMS Materials and Technologies*, Micro and Nano Technologies, pages 19–60. Elsevier, 2009. ISBN 978-0-12-817786-0. doi: https://doi.org/10.1016/B978-0-12-817786-0.00002-5. URL https://www.sciencedirect.com/science/article/pii/B9780128177860000025.

[13] Rania Hendawi and Marisa Di Sabatino. Analyzing structure loss in czochralski silicon growth: Root causes investigation through surface examination. *Journal of Crystal Growth*, 629:127564, 2024. ISSN 0022-0248. doi: https://doi.org/10.1016/j.jcrysgro.2023.127564. URL https://www.sciencedirect.com/science/article/pii/S0022024823004906.

[14] C.B.Honsberg and S.G.Bowden. Photovoltaics education website. https://www.pveducation.org/.

[15] J. Van Zeghbroeck. *Principles of Semiconductor Devices*. 2011. URL https://books.google.no/books?id=hw3YtwEACAAJ.

[16] Nuggehalli Ravindra and V.K. Srivastava. Temperature dependence of the energy gap in semiconductors. *Journal of Physics and Chemistry of Solids*, 40:791–793, 12 1979. doi: 10.1016/0022-3697(79)90162-8.

[17] Philipp Laube. Fundamentals: Doping: n- and p-semiconductors. https://www.halbleiter.org/en/fundamentals/doping/.

[18] The Editors of Encyclopaedia Britannica. Metastable state. https://www.britannica.com/science/metastable-state.

[19] Mark C. Staub and Christopher Y. Li. Towards shape-translational symmetry incommensurate polymer crystals. *Polymer*, 195:122407, 2020.

ISSN 0032-3861. doi: https://doi.org/10.1016/j.polymer.2020.122407. URL https://www.sciencedirect.com/science/article/pii/S0032386120302445.

[20] N. Thejo Kalyani, S.J. Dhoble, B. Vengadaesvaran, and Abdul Kariem Arof. Chapter 20 - sustainability, recycling, and lifetime issues of energy materials. In S.J. Dhoble, N.Thejo Kalyani, B. Vengadaesvaran, and Abdul Kariem Arof, editors, *Energy Materials*, pages 581–601. Elsevier, 2021. ISBN 978-0-12-823710-6. doi: https://doi.org/10.1016/B978-0-12-823710-6.00015-7. URL https://www.sciencedirect.com/science/article/pii/B9780128237106000157.

[21] Libo Wang, Jinpei Liu, Yanzheng Li, Ganghui Wei, Qiong Li, Zining Fan, Hao Liu, Yue An, Chenxi Liu, Junshuai Li, Yujun Fu, and Qiming Liu. Dislocations in crystalline silicon solar cells. *Advanced Energy and Sustainability Research*, 5, 12 2023. doi: 10.1002/aesr.202300240.

[22] Heinrich Häberlin. *Photovoltaics: system design and practice*. John Wiley & Sons, 2012.

[23] Chiara Candelise and Nicoletta Marigo. What is behind the recent dramatic reductions in photovoltaic prices? the role of china. *ECONOMIA E POLITICA INDUSTRIALE*, 40:5–41, 09 2013. doi: 10.3280/POLI2013-003001.

[24] N. Usami. 4 - types of silicon–germanium (sige) bulk crystal growth methods and their applications. In Yasuhiro Shiraki and Noritaka Usami, editors, *Silicon–Germanium (SiGe) Nanostructures*, Woodhead Publishing Series in Electronic and Optical Materials, pages 72–82. Woodhead Publishing, 2011. ISBN 978-1-84569-689-4. doi: https://doi.org/10.1533/9780857091420.2.72. URL https://www.sciencedirect.com/science/article/pii/B9781845696894500043.

[25] Hans J Scheel. Historical aspects of crystal growth technology. *Journal of Crystal Growth*, 211(1):1–12, 2000. ISSN 0022-0248. doi: https://doi.org/10.1016/S0022-0248(99)00780-0. URL https://www.sciencedirect.com/science/article/pii/S0022024899007800.

[26] Halima Zahra Bukhari. Modeling and control of the czochralski crystal growth process. 2021. URL https://hdl.handle.net/11250/2773699.

[27] V. Prakash, A. Agarwal, and E K Mussada. Processing methods of silicon to its ingot: a review. *Silicon*, 11(11):1617–1634, 2019. doi: https://doi.org/10.1007/s12633-018-9983-0. URL `https://link.springer.com/article/10.1007/s12633-018-9983-0citeas`.

[28] Ptable. Periodic table ptable properties. https://ptable.com/Properties/Series.

[29] Xuegong Yu and Deren Yang. Growth of crystalline silicon for solar cells: Czochralski si. pages 129–174, 2019. doi: 10.1007/978-3-662-56472-1_12. URL `https://doi.org/10.1007/978-3-662-56472-1_12`.

[30] Lei Jiang, Da Teng, and Yue Zhao. A soft measurement method for the tail diameter in the growing process of czochralski silicon single crystals. *Applied Sciences*, 14(4), 2024. ISSN 2076-3417. doi: 10.3390/app14041569. URL `https://www.mdpi.com/2076-3417/14/4/1569`.

[31] G. Coletti. Impurities in silicon and their impact on solar cell performance. 2011.

[32] Øyvind S. Sortland, Moez Jomâa, Mohammed M'Hamdi, Eivind J. Øvrelid, and Marisa Di Sabatino. Statistical analysis of structure loss in czochralski silicon growth. *AIP Conference Proceedings*, 2147(1): 100002, 8 2019. ISSN 0094-243X. doi: 10.1063/1.5123875. URL `https://doi.org/10.1063/1.5123875`.

[33] Keigo Hoshikawa and Xinming Huang. Oxygen transportation during czochralski silicon crystal growth. *Materials Science and Engineering: B*, 72(2):73–79, 2000. ISSN 0921-5107. doi: https://doi.org/10.1016/S0921-5107(99)00494-8. URL `https://www.sciencedirect.com/science/article/pii/S0921510799004948`.

[34] G. Müller, A. Mühe, R. Backofen, E. Tomzig, and W.v. Ammon. Study of oxygen transport in czochralski growth of silicon. *Microelectronic Engineering*, 45(2):135–147, 1999. ISSN 0167-9317. doi: https://doi.org/10.1016/S0167-9317(99)00115-X. URL `https://www.sciencedirect.com/science/article/pii/S016793179900115X`.

[35] Robert Doering Yoshio Nishi. *Handbook of Semiconductor Manufacturing Technology, Second Edition*. CRC Press, 2 edition, 2007. ISBN 1574446754,9781574446753. URL `http://gen.lib.rus.ec/book/index.php?md5=e052daf0825891a3b037a453916b4b2b`.

[36] Michael W Berry, Azlinah Mohamed, and Bee Wah Yap. *Supervised and unsupervised learning for data science*. Springer, 2019.

[37] Xiaonan Zou, Yong Hu, Zhewen Tian, and Kaiyuan Shen. Logistic regression model optimization and case analysis. In *2019 IEEE 7th international conference on computer science and network technology (ICCSNT)*, pages 135–139. IEEE, 2019.

[38] Jr. Frank E. Harrell. Regression modeling strategies. *Springer Series in Statistics*, pages 219–221, 2016. ISSN 0172-7397. doi: https://doi.org/10.1007/978-3-319-19425-7.

[39] Scott A Czepiel. Maximum likelihood estimation of logistic regression models: theory and implementation. *Available at czep. net/stat/mlelr. pdf*, 83, 2002.

[40] scikit-learn developers. Logisticregression - scikit-learn 1.0.2 documentation, 2023. URL `https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html`. Accessed: 2024-04-30.

[41] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[42] Tony Thomas, Athira P. Vijayaraghavan, and S. Emmanuel. Applications of decision trees. 2019. doi: 10.1007/978-981-15-1706-8_9.

[43] Sotiris B Kotsiantis. Decision trees: a recent overview. *Artificial Intelligence Review*, 39:261–283, 2013.

[44] Gaowei Xu, Min Liu, Zhuofu Jiang, Dirk Söffker, and Weiming Shen. Bearing fault diagnosis method based on deep convolutional neural network and random forest ensemble learning. *Sensors*, 19(5):1088, 2019.

[45] Leo Breiman. Bagging predictors. *Machine learning*, 24:123–140, 1996.

[46] Alexey Tsymbal, Mykola Pechenizkiy, and Pádraig Cunningham. Dynamic integration with random forests. In *Machine Learning: ECML 2006: 17th European Conference on Machine Learning Berlin, Germany, September 18-22, 2006 Proceedings 17*, pages 801–808. Springer, 2006.

[47] scikit-learn developers. Randomforestclassifier - scikit-learn 1.0.2 documentation, 2023. URL `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`. Accessed: 2024-04-30.

[48] S. Agatonovic-Kustrin and R. Beresford. Basic concepts of artificial neural network (ann) modeling and its application in pharmaceutical research. *Journal of pharmaceutical and biomedical analysis*, 22 5: 717–27, 2000. doi: 10.1016/S0731-7085(99)00272-1.

[49] M. Alsmadi, K. Omar, S. Noah, and Ibrahim Almarashdah. Performance comparison of multi-layer perceptron (back propagation, delta rule and perceptron) algorithms in neural networks. *2009 IEEE International Advance Computing Conference*, pages 296–299, 2009. doi: 10.1109/IADCC.2009.4809024.

[50] Roger Grosse. Lecture 5: Multilayer perceptrons. *inf. téc*, 2019.

[51] Sridhar Narayan. The generalized sigmoid activation function: Competitive supervised learning. *Information Sciences*, 99(1):69–82, 1997. ISSN 0020-0255. doi: https://doi.org/10.1016/S0020-0255(96)00200-9. URL `https://www.sciencedirect.com/science/article/pii/S0020025596002009`.

[52] Zheng Hu, Jiaojiao ZHANG, and Yun Ge. Handling vanishing gradient problem using artificial derivative. *IEEE Access*, PP:1–1, 01 2021. doi: 10.1109/ACCESS.2021.3054915.

[53] scikit-learn developers. Mlpclassifier - scikit-learn 1.0.2 documentation, 2023. URL `https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html`. Accessed: 2024-04-30.

[54] P. Nystrup, Erik Lindström, and H. Madsen. Hyperparameter optimization for portfolio selection. 2:40 – 54, 2020. doi: 10.3905/jfds.2020.1.035.

[55] Li Yang and Abdallah Shami. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415: 295–316, 2020. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2020.07.061. URL `https://www.sciencedirect.com/science/article/pii/S0925231220311693`.

[56] Isaac Kofi Nti, Owusu Nyarko-Boateng, Justice Aning, et al. Performance of machine learning algorithms with different k values in k-fold cross-validation. *International Journal of Information Technology and Computer Science*, 13(6):61–71, 2021.

[57] Mohammad Hossin and Md Nasir Sulaiman. A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, 5(2):1, 2015.

[58] Charles E. Metz. Basic principles of roc analysis. *Seminars in Nuclear Medicine*, 8(4):283–298, 1978. ISSN 0001-2998. doi: https://doi.org/10.1016/S0001-2998(78)80014-2. URL `https://www.sciencedirect.com/science/article/pii/S0001299878800142`.

[59] Andrew P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7): 1145–1159, 1997. ISSN 0031-3203. doi: https://doi.org/10.1016/S0031-3203(96)00142-2. URL `https://www.sciencedirect.com/science/article/pii/S0031320396001422`.

[60] Yaoshiang Ho and Samuel Wookey. The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling. *IEEE Access*, 8:4806–4813, 2020. doi: 10.1109/ACCESS.2019.2962617.

[61] Scikit learn Developers. *SimpleImputer*, 2023. URL `https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html`. Accessed: 2024-06-02.

[62] Wahiba Yaïci, Evgueniy Entchev, Michela Longo, Morris Brenna, and Federica Foiadelli. Artificial neural network modelling for performance prediction of solar energy system. In *2015 International Conference on Renewable Energy Research and Applications (ICRERA)*, pages 1147–1151, 2015. doi: 10.1109/ICRERA.2015.7418589.

Appendix A

Feature Importance

Here, the feature importance for each model is included for neck, crown, shoulder and body. The importance of features will vary for each model, due to each model has different assumptions about the data. Logistic regression assumes linearity, while random forest and neural network can model interactions between features, making some features appear more important due to their combined effect with others.

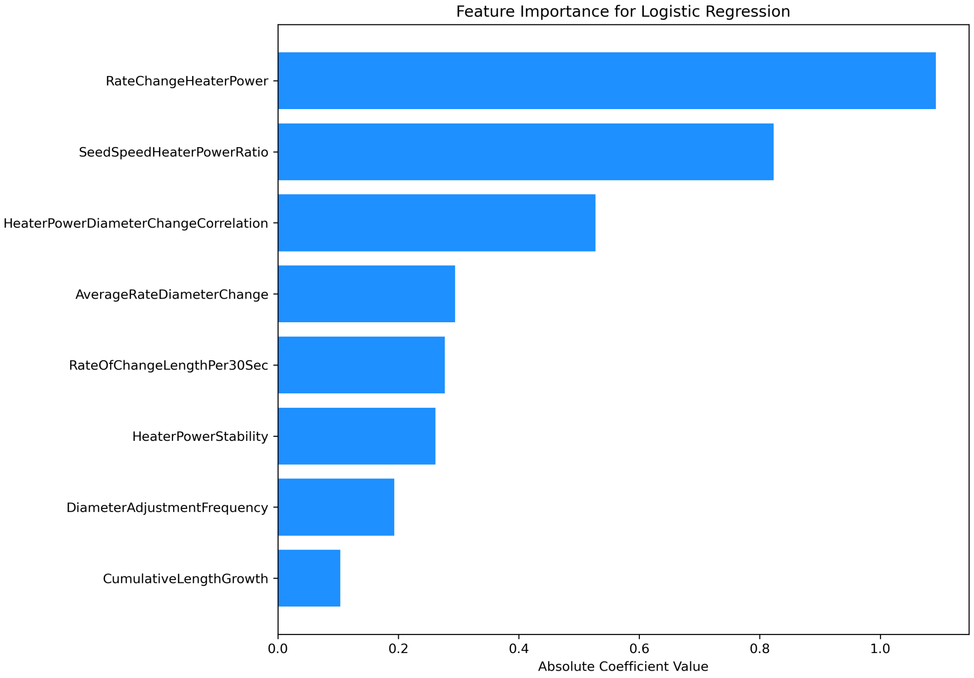


Figure A.1: Feature Importance for Logistic Regression (Neck)

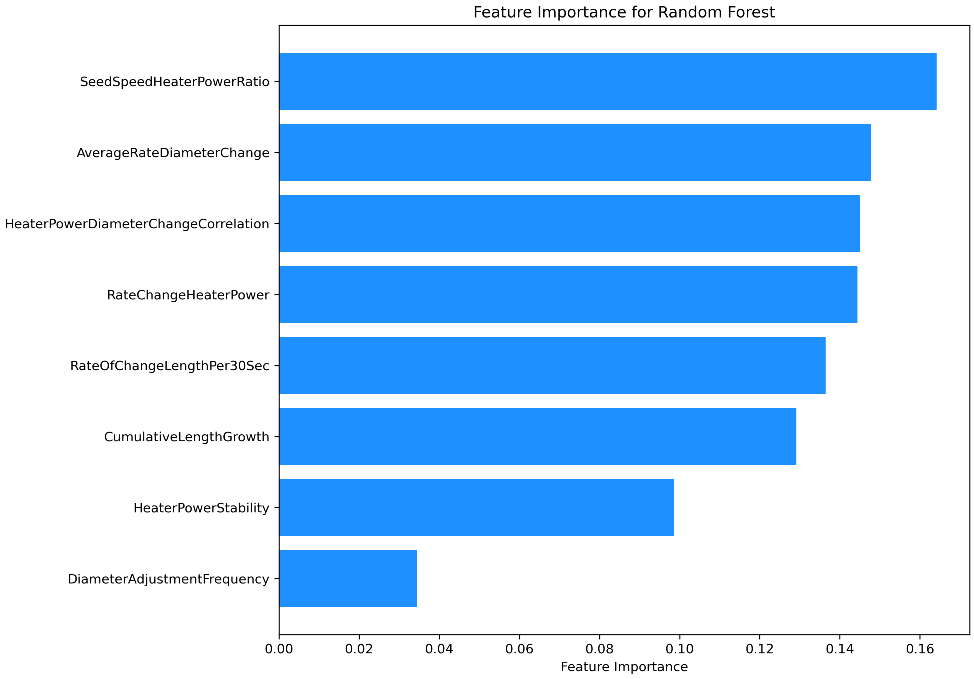


Figure A.2: Feature Importance for Random Forest (Neck)

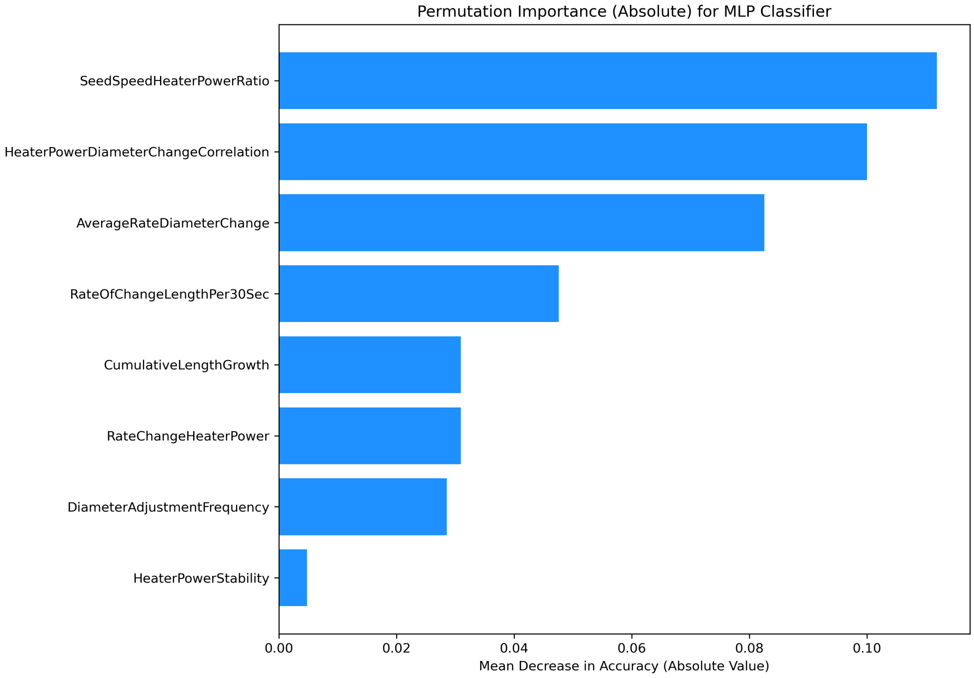


Figure A.3: Permutation Importance (Absolute) for MLP Classifier (Neck)

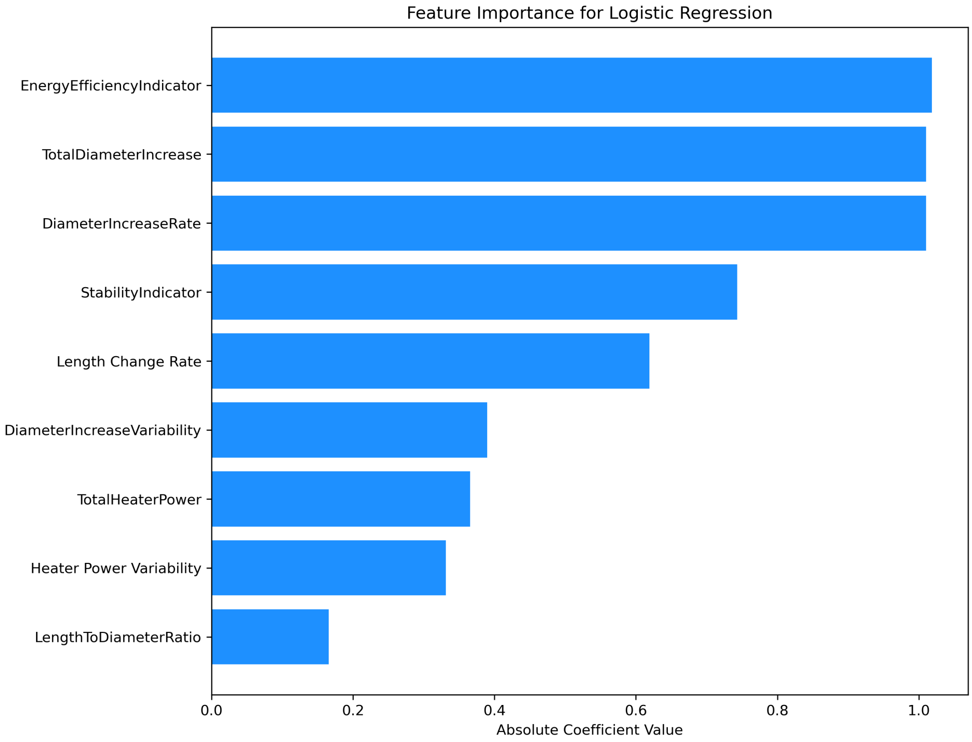


Figure A.4: Feature Importance for Logistic Regression (Crown)

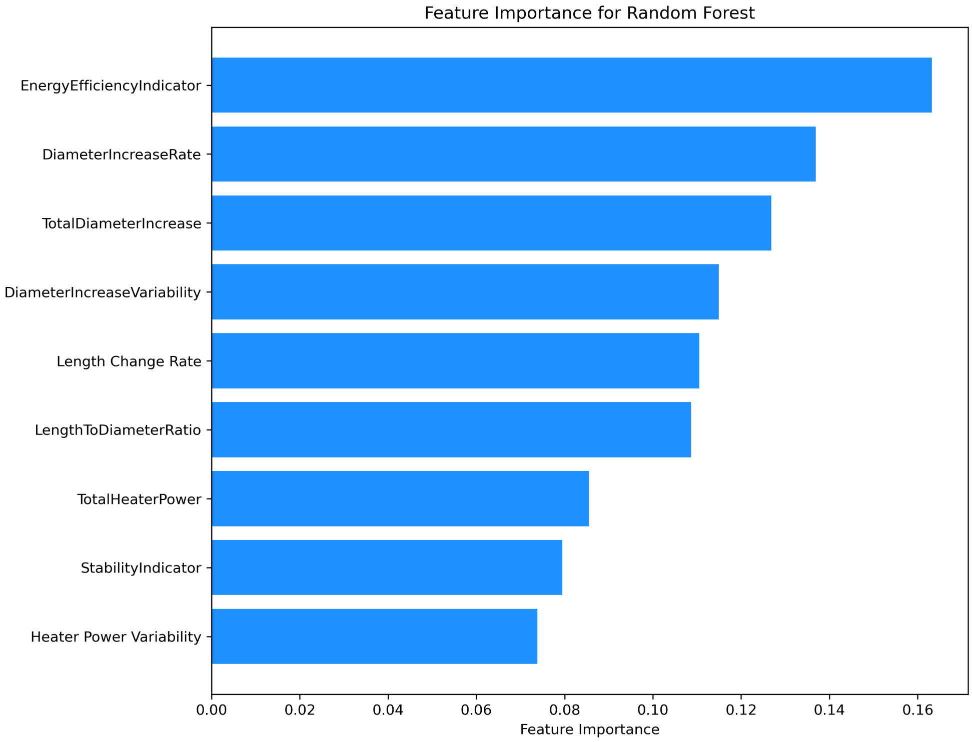


Figure A.5: Feature Importance for Random Forest (Crown)

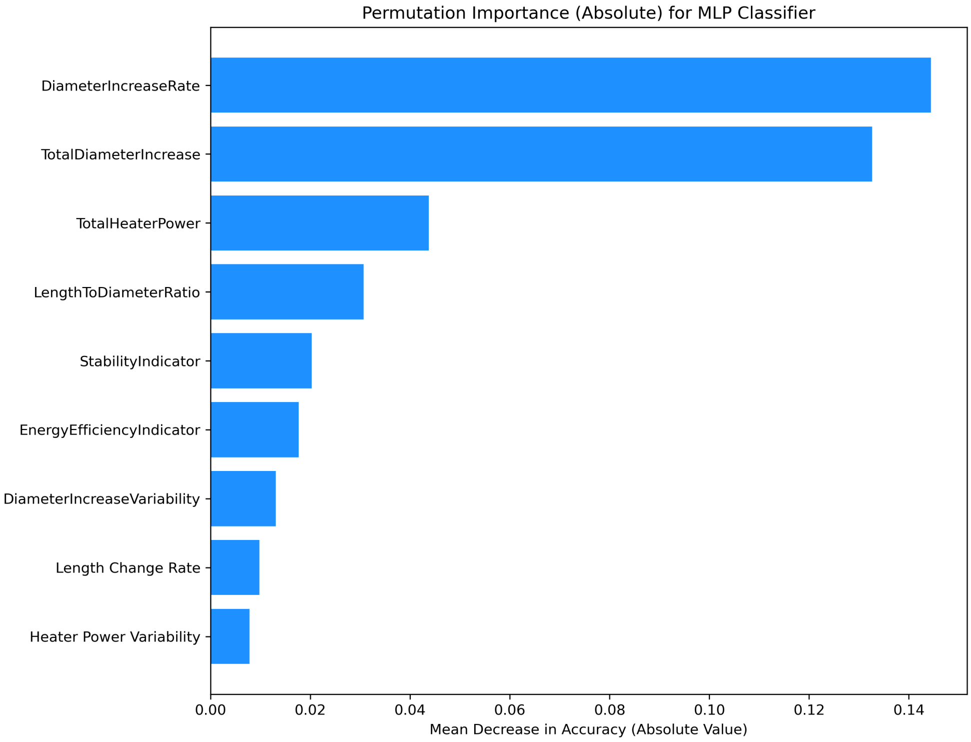


Figure A.6: Permutation Importance (Absolute) for MLP Classifier (Crown)

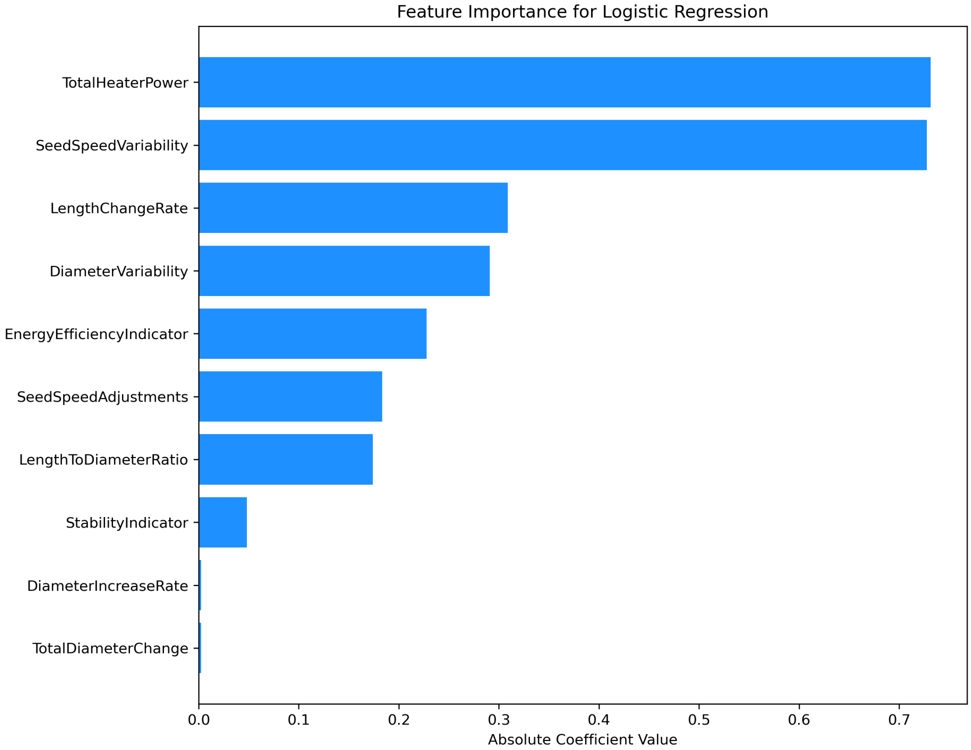


Figure A.7: Feature Importance for Logistic Regression (Shoulder)

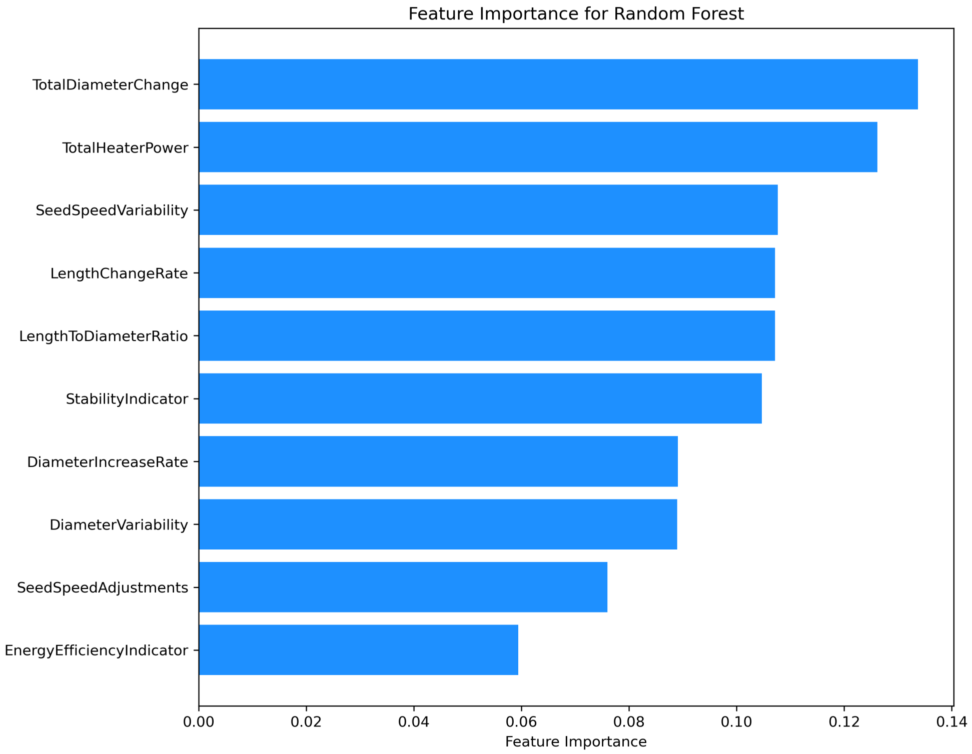


Figure A.8: Feature Importance for Random Forest (Shoulder)

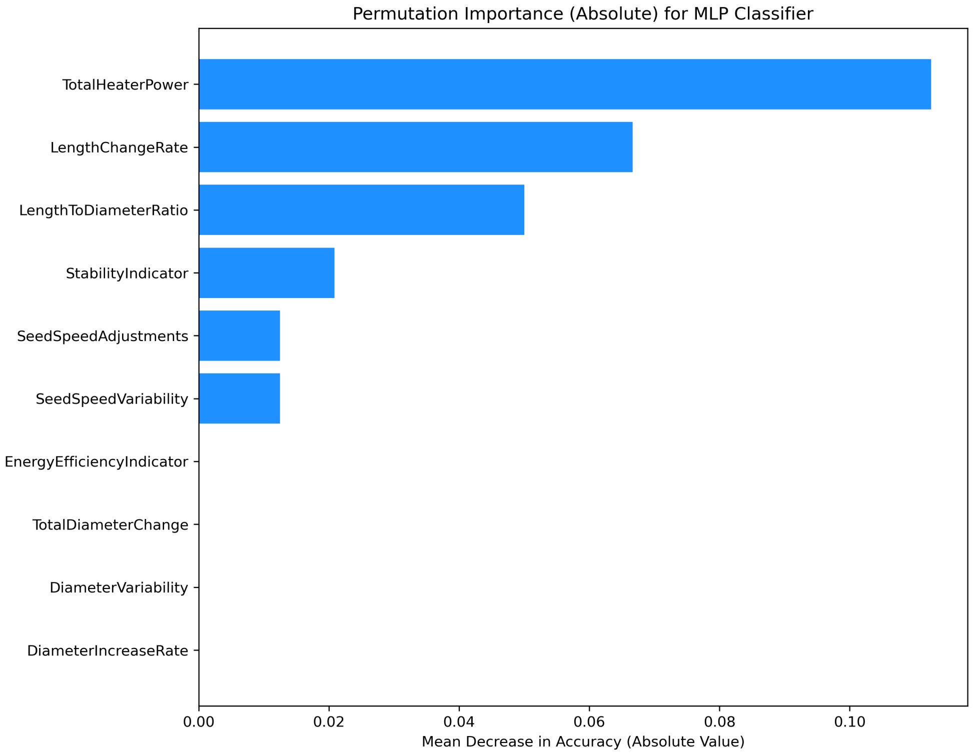


Figure A.9: Permutation Importance (Absolute) for MLP Classifier (Shoulder)

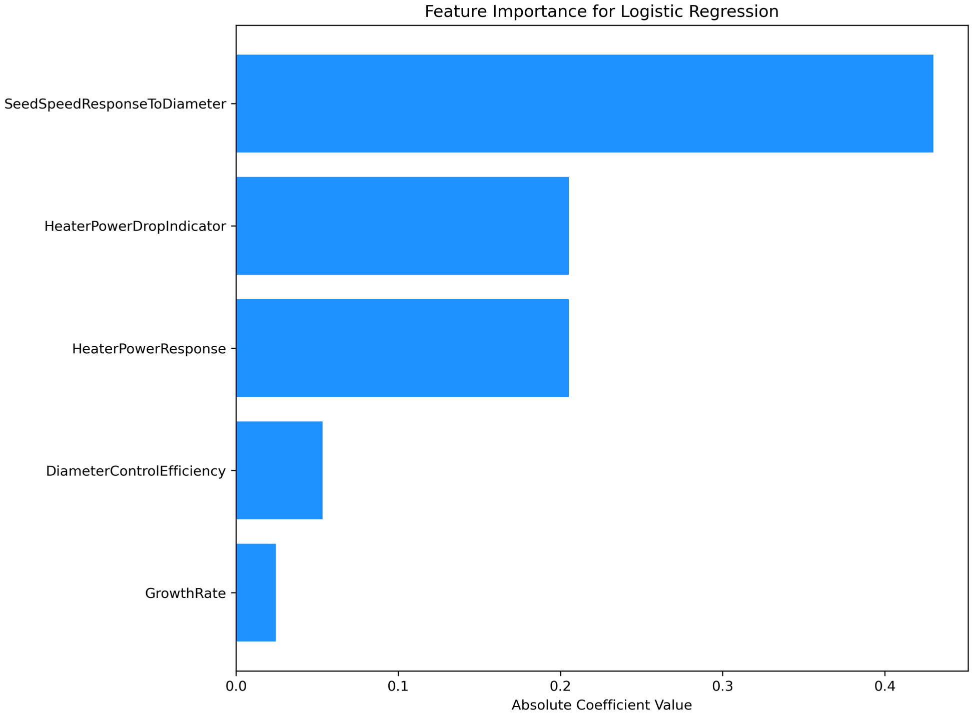


Figure A.10: Feature Importance for Logistic Regression (Body)

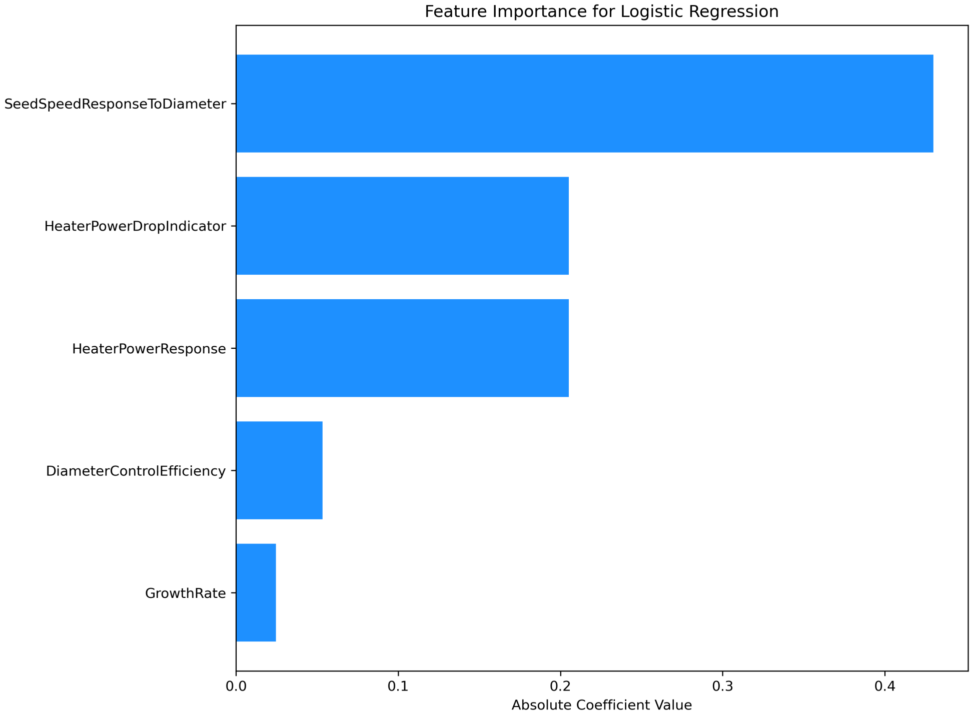


Figure A.11: Feature Importance for Random Forest (Body)

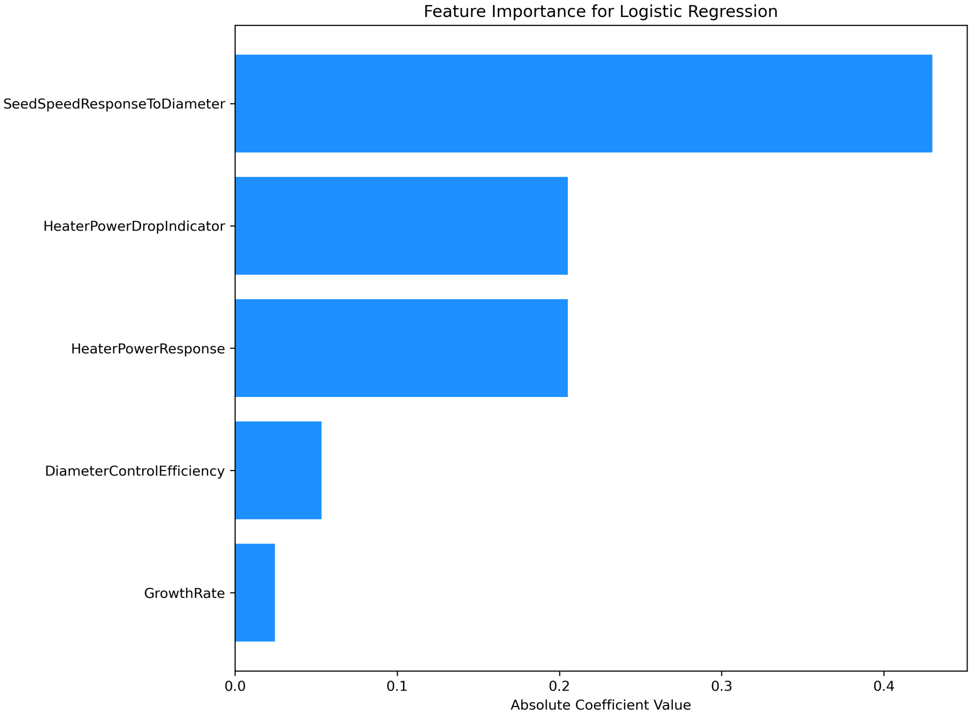


Figure A.12: Permutation Importance (Absolute) for MLP Classifier (Body)