To Investigate Bosch production failures and production line performance thereby increase production efficiency and product quality.

Mustafa Güneş Çetin

A Thesis Submitted in Partial Fulfilment

of the requirements for the

Degree of

Master of Science in Data Analytics

Diagram

Description automatically generated with medium confidence

December 2023

Supervisor: Kislay Raj

**CCT College Dublin**

**Assessment Cover Page**

|  |  |
| --- | --- |
| **Module Title:** | Capstone Project |
| **Assessment Title:** | To Investigate Bosch production failures and production line performance thereby increase production efficiency and product quality. |
| **Lecturer Name:** | Kislay Raj |
| **Student Full Name:** | Mustafa Güneş Çetin |
| **Student Number:** | 2022456 |
| **Assessment Due Date:** | 15/12/2023 |
| **Date of Submission:** | 15/12/2023 |

**Declaration**

|  |
| --- |
| By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution. |

[**1.** **Introduction** 1](#_Toc153563777)

[1.1. The Fourth Industrial Revolution and Artificial Intelligence 1](#_Toc153563778)

[1.2. Challenges and Opportunities in Smart Manufacturing with IoT and Computational Intelligence 1](#_Toc153563779)

[1.3. Industrial Internet of Things and Advanced Analytics for Process Surveillance 2](#_Toc153563780)

[1.4. Methods in Data Science for Real-Time Monitoring in Manufacturing 3](#_Toc153563781)

[1.5. General Project Aim 5](#_Toc153563782)

[1.6. Objectives 5](#_Toc153563783)

[1.7. Data Exploration 6](#_Toc153563784)

[1.7.1. Dataset 7](#_Toc153563785)

[1.7.2. Categorical features 8](#_Toc153563786)

[1.7.3. Numerical features 8](#_Toc153563787)

[1.7.4. Date features 8](#_Toc153563788)

[1.7.5. Exploratory Data Analysis 9](#_Toc153563789)

[2. Background and Related Work 15](#_Toc153563790)

[3. Methodology 20](#_Toc153563791)

[3.1. Feature Selection 20](#_Toc153563792)

[3.2. Preprocessing Step 21](#_Toc153563793)

[3.3. Random Forest Classifier 22](#_Toc153563794)

[3.4. AdaBoost Classifier and Bagging 23](#_Toc153563795)

[3.7. K-fold Cross Validation 27](#_Toc153563796)

[4. Experimental Results 28](#_Toc153563797)

[5. Conclusions 33](#_Toc153563798)

6. References 33

1. **Introduction**

## The Fourth Industrial Revolution and Artificial Intelligence

Artificial intelligence plays a crucial role in the fourth industrial revolution. (Li et al. 2017; Zhong et al. 2017). The focus of value generation, artifactual paradigms, enabling technologies, applied methodologies, and business strategies of design are continually enriched. This is a fact of the matter. (Horváth, 2021). With the acquisition of more appropriate knowledge and information and more applications, a person or system can become smarter (Rindermann and Ceci, 2009).

* 1. Challenges and Opportunities in Smart Manufacturing with IoT and Computational Intelligence

Smart manufacturing is being touted as the next indus- trial revolution (M. Bryner 2012).  
The objective of the smart manufacturing movement is to establish production operations that incorporate information. Operations undergo a transformation from being reactive to proactive, taking action before responding, shifting from compliance to performance, transitioning from tactical to strategic, and expanding from local to global. Tata Motors' smart factory implements the Nano model and possesses the capability to predict bottlenecks and malfunctions through the utilisation of automation technologies such as sensors, microprocessors, and motor controllers. Additionally, it possesses the capability to procure components from suppliers in real time (Bryner, 2012). It is widely believed by numerous researchers that the implementation of a computational intelligence-assisted design framework is imperative for the development of smart systems. The operational and behavioural self-adaptation requires a dedicated system intelligence (Ashby, 1947). In other words, the ability of the system to expand its knowledge base and improve its reasoning mechanisms when necessary will be a fundamental measure of the intelligence of systems. In a simplified manner, the level of intelligence exhibited by intellectualised engineering systems can be assessed based on their capacity to address a variety of complex real-world application problems. Additionally, their cognitive abilities can be evaluated based on their capacity to expand their knowledge base and problem-solving mechanisms. In the domain of system engineering, there exists a direct correlation between the inherent self-adaptive capabilities and potentialities (Sabatucci et al., 2018).

## Industrial Internet of Things and Advanced Analytics for Process Surveillance

Intelligent products and services are now widely available thanks to growth of the Internet of Things (IoT) (Atzori et al., 2010). The volume of data generated from IoT systems is a significant challenge to state-of-the-art big data solutions. Generating insights from the extensive amount of data requires the development of efficient data engineering systems and algorithms tailored to the specific type of sensor data being analysed (Maurya, 2016). In their study, (Bures et al.,2020) examined the manifestation of smart systems, which are characterised by a diverse and interconnected landscape comprising different applications of the IoT, Cyber-Physical Systems (CPSs), and smart sensing systems. In addition, the observers have witnessed a conventional implementation of a smart system application. This implementation consists of various autonomous components that inherently collaborate with each other. These components encompass hardware units that operate on specific networks, as well as software components that are associated with them. The achievement of smartness in this application is realised through the combined capabilities of sensing and operation, which are carried out both autonomously and collaboratively. Components proactively sense the environment and provide their knowledge to other components in order to enable them to make intelligent and informed decisions. A typical conceptualization of a smart system is that it changes its reasoning strategy and activates problem solving agents accordingly. It also learns new models to process a changing and growing set of input (sensor) data or knowledge base contents. The computing mechanisms are preprogrammed to perform this task, although the necessary adaptation and computational resources are determined during runtime. Systems adapt themselves within an anticipated envelope of changes. The concept of smartness does not imply that all decisions are made solely by the mechanisms of reasoning, but rather that humans are involved in the operational loops of the system (Schirner et al., 2013). In the realm of Industrial IoT, manufacturing processes and product lifecycles are under constant surveillance to detect any irregularities or failures (Xu et al., 2014). Utilising advanced analytics is crucial in extracting valuable insights and business value from the vast amount of data collected through IoT. Nevertheless, there are certain difficulties associated with IoT data analytics. The datasets collected using sensors are more complex and challenging to interpret compared to the structured datasets collected through human interaction on Internet websites (Maurya, 2016) In the realm of Industrial IoT, manufacturing processes and product lifecycles are under constant surveillance to detect any irregularities or failures (Xu et al., 2014). We consider data, information, and knowledge as separate and distinct tiers. Data are commonly perceived as a compilation of factual elements, whereas information encompasses the significance derived from interconnected data. Knowledge, on the other hand, represents the capacity to solve problems by integrating and abstracting information. Lastly, intellect denotes the aptitude to effectively apply acquired knowledge across diverse and ever-changing circumstances. Sensors collect signals from various sources and subsequently transform them into data. Information structures serve the purpose of capturing and encoding relationships among data, thereby revealing their inherent meaning. Advanced reasoning mechanisms, such as artificial neural networks, have the capability to uncover hidden relationships within vast data streams and transform them into patterns of knowledge. The field of context management involves acquiring and utilising meta-knowledge to effectively apply problem-solving knowledge (Horváth, 2020). The collection of data serves as the initial and fundamental step in the realm of industrial big data, assuming a crucial role in facilitating data-driven maintenance practises within the industrial sector. The establishment of a fine-grained data flow for subsequent statistical analysis and machine learning is challenging due to the integration of devices from various equipment suppliers, the diversity of communication protocols, varying degrees of device openness, and differing levels of intelligence among the devices (Wan, 2017). As the integration of information technologies continues to expand across various sectors of industry, there has been a growing emphasis on the incorporation of increasingly ssophisticated algorithms in the realm of active preventive maintenance. However, in the industrial setting, the need for durability and immediate processing becomes much more important. For example, the utilisation of deep learning has shown exceptional effectiveness in the fields of image identification and natural language processing. Empirical research has confirmed that deep learning algorithms may effectively utilise stored data to generate remarkable outcomes in various domains. However, caution must be exercised when considering the suitability of deep learning in the context of active preventive maintenance. Algorithms that may not possess a high level of sophistication, but when coupled with expert knowledge, have the potential to yield enhanced performance (Wan, 2017).

## Methods in Data Science for Real-Time Monitoring in Manufacturing

Manufacturing companies are exploring ways to adapt to industry 4.0 due to the growing need for advancements that can enhance their product yield (Lee et al., 2014). Quality control has long been a crucial component of production and an essential component. More advanced sensor technologies, such IoT and Radio Frequency Identification (Zaslavsky, et al., 2012), allow data to be collected at every level of the production process. Companies can collect crucial process data from different stages of their production line during the manufacturing process. By harnessing copious amounts of accessible data, firms can utilise advanced data analytics to effectively analyse and elucidate uncertainties. This enables them to make more perceptive and well-informed judgements regarding future product development. (Carbery et al., 2018). AI techniques have been utilised to enhance the early identification or prediction of faults in production lines within the field of Intelligent Manufacturing (Kusiak 2017; Tao et al. 2018). Manufacturing processes exhibit significant intricacy as they typically involve multiple sequential steps and necessitate various sorts of resources, including financial, human, and technological resources. The recent technological breakthroughs have provided several options to extract valuable information from manufacturing processes. This information can be used to optimise production and identify any factors that may result in wasteful allocation of resources. Analysing the data that describes the fabrication processes is highly tough because the products undergo numerous intricate operations. As a result, the data may exhibit characteristics such as missing information and a large number of features (Moldovan et al., 2019). Utilising standard statistical approaches to extract information from data that describes manufacturing processes is frequently insufficient for identifying procedures that may result in errors in the final outcomes (Moldovan et al., 2019).

Recent research efforts have delved into machine learning techniques for simulating manufacturing domains, focusing on predictive maintenance and defect detection (Susto, et al., 2015; Wuest et al., 2014; Kroll, et al., 2015; Wang, et al., 2017). The utilization of advanced machine learning and deep learning techniques for forecasting product quality has demonstrated high value, offering more precise predictions. However, challenges, such as selecting the most suitable prediction model and addressing missing information, need to be addressed, particularly in light of the intricate nature of manufacturing data (Moldovan et al., 2019).  
Given the importance of staying competitive and increasing productivity, it is clear that incorporating data science methods for real-time monitoring of manufacturing processes is a logical progression. An notable accomplishment occurred in 2012 when Intel managed to reduce its manufacturing expenses by $3 million by employing predictive analytics to prioritise inspections of their silicon chips (Ronen and Burns, 2013). As further examples, Raytheon, a defence manufacturing corporation, introduced the Manufacturing Execution System (MES) at its missile facility in Huntsville, Alabama (Hagerty, 2013). This system gathers and examines data from the factory shop floor, enabling precise determination of the optimal number of screw rotations required for a flawless critical component. Following this trend, Bosch competed in a Kaggle competition (Kaggle, 2016) and released its dataset, which consisted of anonymized records of measurements and tests made for each component along the assembly line. The company then challenged the Kaggle community to predict product part failures, which ultimately enabled Bosch to provide quality products to the end user at lower costs.

## General Project Aim

In this context, in this study highlights the importance of using advanced data analytics and machine learning techniques to understand, detect and solve problems encountered in Bosch's production line. The primary goal of the study is to identify the root causes of failures occurring in Bosch production lines and to increase production efficiency to finding solutions for these reasons.  
This study is important in the following key areas:

1. **Fast and Accurate Detection of Problems:** Using data analytics and machine learning models, problems on production lines can be quickly detected. This allows businesses to take a proactive rather than reactive approach.
2. **Determination of Root Causes:** The study addresses the root of the problems by identifying the root causes of failures in production lines. In this way, similar problems can be prevented from recurring.
3. **Data Based Decision Making:** Advanced data analytics and machine learning techniques enable businesses to make data-driven decisions. This optimizes business processes and increases efficiencies.

In conclusion, this study, using production data shared by Bosch on Kaggle, highlights the importance of data analytics and machine learning applications in industrial production. Successful adoption of these technologies enables Bosch to more effectively manage problems in production processes and increase production efficiency. Therefore, the study is expected to be an important reference source for companies seeking data-driven solutions in industrial production.

## Objectives

This study includes two main objectives set to understand and solve problems in Bosch's production line:

**RQ1: What are the primary causes of manufacturing failures on Bosch's production line, and how can these root causes be identified and addressed?**

The primary objective is to identify and examine malfunctions that have transpired in Bosch's manufacturing facilities and comprehend their underlying origins. To effectively address these factors, it is essential to get a significant outcome through the analysis. Within this framework, the objective is to ascertain the underlying causes of issues in the production line and devise solution plans through the utilisation of analyses and learning algorithms. This goal aims to focus on key problems to prevent efficiency losses in production processes and improve product quality.

**RQ2: How can advanced data analytics and machine learning techniques be leveraged to predict and mitigate manufacturing failures in real-time on the Bosch production line?**

The second goal is to use advanced data analytics and machine learning techniques to predict and prevent production errors in real time on Bosch's production line. Thanks to the models to be used, potential errors in the production process will be predicted in advance, paving the way for instant interventions to these errors and optimizing production processes. This goal aims to ensure that production lines operate more reliably and efficiently by strengthening predictive maintenance strategies.

The purposes are to achieve these two objectives in order to offer more efficient and strategic solutions to the issues in Bosch's manufacturing processes, resulting in cost reduction and enhanced customer satisfaction. Furthermore, this study seeks to offer a thorough comprehension of the potent utilisation of data analytics and machine learning in industrial production.

## Data Exploration

The dataset utilized in this study was provided by Bosch as a part of a data science competition hosted on Kaggle. It comprises measurements of manufactured parts as they progress through Bosch's production lines. Each data point is identified by a unique "Id" assigned to the manufactured part, and the binary target variable indicates whether the part fails quality control, with a value of 1 denoting failure.

### Dataset

Bosch has supplied a huge dataset (14.3 GB) containing three types of feature data: numerical, categorical, date stamps and the labels indicating the part as good or bad. The training data has 1184687 samples and the learned model will be used to predict on a test dataset containing 1183748 samples. There are 968 numerical features, 2140 categorical features and 1156 date features. Hence, one of the biggest challenges of this dataset is to process these features into something meaningful so they can be used to make a predictive model.

This extensive dataset includes numerous anonymized features specific to production lines and stations. For instance, features are denoted as L2 S95 F3456, representing the 3456th anonymous feature measured during product manufacturing on line 2 and station 95. The dataset is presented in three file types: Numerical (real-valued features), Categorical (discrete-valued features), and Date (timestamp for when each numeric or categorical feature was recorded).

The massive dataset was provided in the form of three types of ﬁles:

* Numerical: This type of ﬁle contains real-valued features.
* Categorical: This type of ﬁle records discrete-valued features.
* Date: This type of ﬁle records the timestamp for when each numeric or categorical feature was recorded.

There are two ﬁles per type: ﬁrst for the training dataset, and second for the test dataset. Hence, the ﬁnal input data ﬁles are as follows:

* train numeric.csv - the training set numeric features, which contains the ’Response’ variable.
* test numeric.csv - the test set numeric features, which contains the Ids for which the ’Response’ variable must be predicted.
* train categorical.csv - the training set categorical features
* test categorical.csv - the test set categorical features
* train date.csv - the training set date features
* test date.csv - the test set date features

### Categorical features

The categorical data has 2140 features, but on further evaluation, we ﬁnd that about 500 are multi value, 1490 single value and 150 are empty. Figure 1 The empty categorical features can be dropped as they contain no information. The single and multi-value categorical features can be converted to numerical by using the feature importance technique, where each class is represented by an integer.

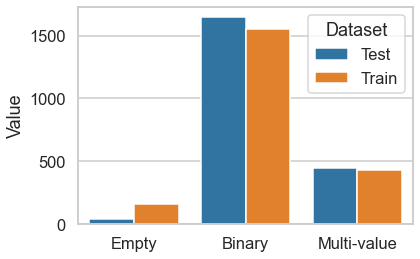


Figure 1 Categorical column types: Numbers of empty, binary and multiple categories in test and training sets

### Numerical features

The numerical feature names contain information about the stations, production line and a test number combination. The value for that feature is the corresponding measure- ment. For example, a feature named L3 S50 F4243 for a component indicates that the part went through production line 3, station 50, and the feature value corresponds to a test number 4243. This way, each product coming out of the manufacturing line can be segregated according to the production ﬂow. Observed there exist 51 stations distributed between 4 production lines.

### Date features

The date features names are labeled by production line, station id and date id. For example, L3 S50 D4242, would mean the product went through production line 3, station 50, and the feature value corresponds to date id 4242. There are a total of 1157 date features, with a lot of missing values. Same stations often have same date values.

The dataset is split into training and test sets, with separate files for numerical, categorical, and date features for both sets. The challenge posed by this dataset lies in its vastness in the training set alone, totaling 14.3GB of raw data.

### Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a significant step in gaining a comprehensive understanding of a dataset, identifying important features, and revealing natural patterns in the data. As the first stage of preparing inputs for AI-based models, EDA involves a series of visualization processes that include various graphs, examination of data relationships, and statistical analysis. This important step in data science aims to better understand the connections and patterns between features in the dataset. EDA not only facilitates the assessment of data quality by examining various aspects of the dataset but also helps detecting anomalies, thus creating a building block for subsequent analytical processes. This section presents all the features of the dataset and the relationships and patterns between features.

The data used in the study consists of the measurement values of the parts moving on the production line of the Bosh brand. Each piece has a unique value. Information on which part failed quality control on the data set can be obtained. Failure of part information in quality control is kept with a target value called 'Response'. Figure 2 shows the response value values at the general data set points.

ekran görüntüsü, dikdörtgen, kare, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 2 Target response values distribution

There are two different categories in total. While the "0" category (failed) constitutes the majority with 1,176,868 examples, the "1" category (passed) is represented in a smaller number with 6,879 data points. This distribution shows that the response labels in the dataset are unbalanced, as the "0" category has a significant majority. This type of distribution is a factor that should be considered when training and evaluating the model, as unbalanced data sets can cause the model to face various difficulties in learning responses. In the context of the characteristics of the data set and the analysis objectives, it is important to determine appropriate strategies by taking this distribution into account.

In order to investigate the insignificance in the data set, univariate characteristic analysis was performed. This analysis focuses on understanding the features in the data set by examining the distribution and statistical properties of each variable individually. Univariate analysis provides important information about general patterns in the data set by evaluating the variation, central tendency, and distribution pattern of each variable. This analysis was carried out to better understand how the unbalanced distribution, especially between the "0" and "1" categories, could affect the learning process of the model. Unvariate distribution values are visualized in Figure 3. This visualization will help determine strategies to be used in model training and performance evaluation.

origami, tasarım içeren bir resim

Açıklama otomatik olarak orta güvenilirlik düzeyiyle oluşturuldu

Figure 3 Unvariate distrubitions according to measurements and target value.

The values specified as “Variable” are named according to the station and feature numbers on the production lines. Variables represent measurements performed on the lines. To understand the amount of missing data in the features following each measurement, the “0” category was used as a negative class and the “1” category was used as a positive class.

ekran görüntüsü, dikdörtgen, yazılım, paralel içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 4 Proportion of non missing values.

As seen in Figure 4, a significant amount of missing values are noted in the data set. The frequency of occurrence of missing values shows a significant difference when compared between positive and negative examples. To evaluate this situation, a bar chart was created comparing the missing value rates between negative and positive samples. The missing value rate in negative samples is lower than in positive samples. The graph in the figure visually explains the tendency for missing values to occur, and the tendency for measurement values to be negative or positive varies. This type of analysis is an important step to identify missing data management strategies and optimize model performance.

Figure 5 focuses on the correlation values between negative and positive classes. Presented graphs contain the correlation map that shows the relations between measurements. According to the correlation map, relation intensity is higher in negative classes than in positive classes. In other words, the relationships between samples belonging to the negative class appear more robust and more clearly between measurements. This indicates that samples belonging to the negative class show a more consistent structure in terms of certain features and that these features are in a stronger relationship. Correlations between samples belonging to the positive class are lower or more diverse, indicating a less consistent or distinct relationship between measurements. This analysis is an important step in understanding feature differences between classes and how relationships between measurements may vary across classes. This information is taken into account in feature selection and modeling processes and used to determine more appropriate strategies.

ekran görüntüsü, renklilik, piksel, yaratıcılık içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 5 Correlation maps of negative and positive classes.

ekran görüntüsü, renklilik, kare, grafik içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 6 Difference of negative and positive classes correlation maps.

Figure 6 shows the differences between the correlation maps of the two classes. The matrix obtained by the difference of two matrices generally has sparse relationships, appearing clearly only in three specific feature combinations. The figure presents the focus of the differences in specific feature combinations between both matrices. Given the similarity among other features, apparent differences in these three specific combinations can provide important information, particularly about how these features vary or are related among particular classes. This observation is used to guide modeling strategies more effectively, especially considering that these specific combinations of features may affect the performance of the model or increase the separation between classes.

Network visualization operations were performed to better understand the relationship between each data point. Network visualization is used to graphically represent complex relationships between stations on a Bosch production line. Network visualization is an important tool for making sense of information between complex relationships. Network visualization has applications in many areas, such as understanding interactions between individuals in social networks, examining the relationships between genes and proteins in biological networks, or detecting possible threats in cyber security. It is especially valuable in understanding the relationships between stations on production lines. This technique identifies stations for each data frame and tracks transitions between these stations, creating a global dictionary of station pairs before optimizing systems and increasing efficiency. This visualization provides a valuable tool for understanding and optimizing the network structure in the production line by clearly showing the density, frequency and relationships of transitions between stations.

metin, çizgi, diyagram, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 7 Network visualization of all stations.

First, a data frame was created, and stations were determined according to the rules for the data point. A global dictionary of station pairs was then created by tracking transitions between features at stations for each data frame. In this way, the number of passes of each pair of stations was kept. The dictionary of station pairs is then sorted according to a specific sorting method, and a dictionary of nodes is created to determine the overall visibility of the stations. This node contains the number of connections stations have with each other. The visualization of node connections is given in Figure 7. This visualization visually describes the intensity, frequency, and relationships of transitions between stations. This process is especially important for understanding the transitions between stations on the production line and visualizing the network structure. The network structure provides a valuable tool for understanding the relationships and connections between stations.

# Background and Related Work

The term of productivity, which refers to the ratio of output to input, has been in existence for more than two centuries and has been utilized in diverse contexts and at different levels of aggregation within the economic system. Productivity is often regarded as a fundamental factor that influences economic production activities, and it is often considered the most crucial variable. (Singh, Motwani and Kumar, 2000) Productivity is an important component of any work organization, and any organization that is not operating at its highest level of productivity will not be able to endure over time. Productivity is often regarded as a fundamental factor that influences economic production activities, and it is often considered the most crucial variable. (Tangem, 2002). In the current fiercely competitive business landscape, producers must offer tailored and inventive products. In order to accomplish this, they necessitate inventive approaches for gauging performance. When evaluating manufacturing performance, manufacturers typically assess their own plants by comparing them to other plants in the industry. This comparison is based on various parameters including customer satisfaction, product quality, manufacturing order completion speed, productivity, product line diversity, and ability to manufacture new products with flexibility (Cordero et al., 2005). A performance metric serves the primary purpose of measuring the degree to which the activities or outputs of a process successfully accomplish predetermined objectives. This entails a comparison of real outcomes with a pre-established objective and an evaluation of the degree of any divergence from that objective. The desired level of performance is typically defined as a measurable benchmark, numerical number, or ratio (Ahmad et al., 2005). When operating a manufacturing company, it is important to select a suitable set of performance metrics. These metrics should be well-rounded to ensure that no single aspect of performance is prioritized at the expense of others. (Ahmad ve Dhafr, 2002)

In a study investigating the impact of new production requirements on production line efficiency and quality in a focused factory, identifies inter-subsystem effects within the factory, indicating that alterations in one part of the production system can influence the performance of interconnected subsystems. The research underscores the importance of analyzing manufacturing changes at a granular level and suggests that cause-and-effect relationships in complex systems may not always be clear Mukherjee et al. (2000). The methodological framework presented in the study has proven to be highly effective for analyzing and evaluating production line performance and efficiency. This comprehensive approach, which includes data collection, performance evaluation and results analysis, not only facilitates decision making but also suggests corrective actions to improve the overall production system. highlighted. Emphasizing real-time monitoring, especially in addressing challenges such as manual quality control tasks, the proposed methodology aims to support the evaluation of production line performance. This application not only reduces costs, but also increases productivity by connecting resources across factories, providing real-time data analysis and supporting informed decision-making. The adaptability of the framework allows seamless transfer of experimental data into a simulation environment, facilitating the evaluation of specific parameters and adjustments to production lines. (Fera et al., 2019)

In summary, as seen in these studies, data analysis-based studies emerge as a valuable tool for manufacturing companies, encouraging continuous improvement and adaptability in an ever-improving production performance and efficiency analysis environment.

A new graphical representation and graph-structured neural network-based method is presented in order to address missing data in production data and predict product failures. The findings underline the potential of this innovative approach in the field of data analysis and especially in the use of graph-structured neural network-based methods. This method offers a new perspective on effectively handling missing data in production processes and predicting product failures. (S. Kang, 2020). The main aim of another study is to present a semi-supervised approach called Smart Engineering Analytics and Learning system, developed to predict failures in artificial lift systems in the oil industry. A key finding of the study is system's ability to deal with noisy, multiple multivariate time series, sparsely labeled, and mislabeled data. Additionally, the system can reduce operational expenses and increase overall performance by predicting failures, thus offering significant potential for the oil industry. (Liu et al., 2011) In another study, this study, which aimed to investigate the effectiveness of decentralized learning techniques in failure prediction on the production line, reached important findings (Ge et al., 2021). Main results of the research show that federated learning techniques exhibit similar performance to centralized learning methods and offer an alternative potential for defect prediction in production lines.

"Failure prediction" in production processes is a critical element on which modern industry depends. The three different studies mentioned above reveal the developments in this field and the innovative methods used in various sectors. These findings highlight that various methods and techniques applied in the field of "failure prediction" in the manufacturing industry play an important role in increasing operational efficiency and reducing costs. Therefore, it can be said that a successful "failure prediction" strategy is a critical element to ensure sustainability and reliability in industrial processes.

Accuracy is not the most suitable metric to optimise when dealing with imbalanced datasets in binary classification. As an illustration, in a dataset where the majority of datapoints are negative, it is quite straightforward to achieve a high accuracy of 99% by employing a simplistic model that consistently predicts 0. As a result, additional metrics have been created to evaluate the effectiveness of classifiers on imbalanced datasets:

Ratio of true positives (TP) to the total of true positives and false positives (FP).

Ratio of true positives (TP) to the total of true positives and false negatives (FN).

Harmonic mean of Precision (P) and Recall (R) metrics.

A correlation coefficient measuring the balance between sensitivity and specificity in the performance of a classifier.

The aforementioned indicators provide a more holistic comprehension of the classifier's performance on datasets that exhibit imbalances. This study focuses on optimizing MCC by specifically addressing the issue of class imbalance. Moreover, the approach has the capability to enhance other measures, including as precision and recall.

Prior studies have extensively investigated the direct optimization of the nonconvex metric of interest, rather than just prioritizing accuracy as most machine learning classification methods do. A substantial chunk of this study focuses on optimizing precision while maintaining a minimum degree of recall, or optimizing recall while ensuring precision remains above a specific threshold, or maximizing the mean average precision across queries in information retrieval. Optimising non-convex metrics specified by domain experts for binary classification on imbalanced datasets lacks a general framework.

When there is a large amount of data, we must take into account computational issues associated with the complexity of model learning. Pre-processing in the form of feature selection can be performed on high dimensional data to im- prove learning procedures through reducing both complexity by discarding irrelevant variables. Industries rely on the vast quantities of data gathered by embedded sensors, as well as data generated by the production line, and other sources. We require methods to extract and deduce knowledge from the data (Li et al., 2017).

It has a reputation for delivering competitive outcomes in numerous public data science competitions. Nevertheless, the statistical model prioritizes prediction accuracy and exhibits subpar performance when dealing with imbalanced datasets (Friedman, 2001). (Yue et al., 2007), (Tax et al., 2009), introduced a systematic approach to optimizing precision their research. Nevertheless, these investigations are constrained to linear hypothesis classifiers and necessitate substantial adjustments and computing burden to expand them to the nonlinear scenario and incorporate new metrics like MCC. Furthermore, the optimization approach continues to prioritize accuracy rather than the specific non-convex metric being considered. Therefore, this approach will have subpar performance for datasets with imbalanced class distributions, unlike our method which specifically focuses on optimizing the Matthews correlation coefficient (MCC). In another Work aims to maximizes recall (number of returned results) while maintaining a minimum level of precision (quality of returned results). (Buturovic et al., 2014) Furthermore, they do not specifically prioritize optimizing for MCC. Instead, they focus on optimizing for accuracy and thereafter select the threshold that yields the highest recall at a given precision. Although our method has a similar methodology to achieve maximum precision at a specific recall, it differs in that we directly optimize for the Matthews Correlation Coefficient (MCC). In a study that finds the residuals between predictions and real values to detect changes in the incoming data distribution and the predicted target, although it detects anomalies, it cannot predict them. Furthermore, it fails to establish a categorization model that is appropriate for the specific purpose of identifying uncommon internal manufacturing flaws. (Shan, 2010)

In this section, prior efforts on Bosch product quality prediction based on CL algorithms are introduces and discussed.

Bosch is a prominent global manufacturing company. It guarantees the production's excellent quality by closely monitoring its components during the manufacturing operations. By meticulously records data at every stage of the assembly lines, Bosch is able to employ sophisticated analysis to enhance the manufacturing processes.Therefore, Bosch has put out a dataset for competition on the Kaggle site that uses thousands of measurements and tests done on each part along the assembly line to predict internal failures.

A number of research have conducted analyses of the dataset and implemented methods for forecasting product quality based on CL algorithms, with time-series features being excluded from the analysis. (Carbery et al. 2019, 2018; Zhang et al. 2016; Khoza and Grobler 2019; Kotenko et al. 2019; Mangal and Kumar 2016; Hebert 2016; Maurya 2016) or studies have been conducted that include time series features in their studies. (Huang et al. 2019b; Moldovan et al. 2019; Liu et al. 2020d), The research was carried out using various learning methods such as logistic regression, gradient boosting machine, random forest, gradient boosted trees, Naive Bayes, Bayesian network, K-nearest neighbors, support vector machines, multilayer perceptron classifier, majority voting, decision tree, and statistical process control. A study was done to systematically analyze the features of the Bosch dataset and using Bayesian Network to make predictions about product quality. (Carbery et al. 2018). In another study, which weakened data heterogeneity through clustering and then applied Random Forest, Boosting, Logistic Regression, Naive Bayes, Decision Tree to each cluster, it was concluded that Random Forest performed better than other algorithms (Zhang et al., 2016). Additionally, in another study comparing Random Forest, Naive Bayes, Support Vector Machine and Statistical Process Control, it was observed that Random Forest performed better (Zhang et al., 2016), (Khoza and Grobler, 2019). Employed Support Vector Machine, K-Nearest Neighbors, Logistic Regression, Perceptron, Decision Tree, and Mean Value as classification algorithms. Their findings demonstrated that Support Vector Machine achieved comparatively superior accuracy in prediction. (Kotenko et al., 2019) Mangal and Kumar (2016) con- ducted a visual analysis of the three types of data: categorical, numeric, and time-series features on the Bosch dataset, and used LR, Extra Trees Classifier, RF, XGBoost for quality prediction. (Liu et al., 2020d) introduced a comprehensive quality prediction framework that encompasses all stages of the process and effectively captures the temporal relationships between distinct characteristics. The study conducted by (Moldovan et al., 2019) shown that the LSTM RNN model had superior performance compared to other machine learning models. In a separate study, a visual analysis was performed on the Bosch dataset to examine three types of data: categorical, numeric, and time-series features. LR, Extra Trees Classifier, RF, and XGBoost were utilized to predict the quality (Hebert, 2016). Once and for all, the study compared Federated Learning and Centralized Learning methods for the fault prediction problem and reveals that Federated Support Vector Machine (FedSVM) and Federated Random Forest (FedRF) can replace Support Vector Machine and Random Forest, respectively, for fault prediction in the Bosch production line. (Ge et al., 2021)

# Methodology

The methodology section of the research creates the basic building blocks of the study and explains how the method was carried out and how the data was processed. In this section, the applied preprocessing steps and machine learning methods are explained in depth.

## Feature Selection

Feature selection in the dataset consisting of measurements of parts on the production line and observed defects based on these measurements is an important data preprocessing step to optimize model training time. At the same time, it is aimed to obtain high performance models with more effective inputs by focusing on important features. In this study, different strategies are used for feature selection. In order to perform feature selection, missing data points in the dataset are checked. Features with less than 70% missing values are identified. In this way, data points with a high density of missing points in the samples are eliminated. The first step of feature selection is completed by eliminating according to the rate of missing data. In this step, a cleaner data set is obtained.

In the next step, the variable "feature\_reducing" is set according to certain conditions. If this variable is set to "PCA", dimensionality reduction is performed using the Truncated Singular Value Decomposition for Sparse Matrices (TruncatedSVD) approach. This approach reduces the dimensionality of the feature matrices and optimizes memory usage. In addition, this approach contributes to a faster training process by avoiding the high dimensionality problem. New data frames are created with the selected data and the inputs of the models are obtained. As another selection method, feature\_reducing` variable is set as "fi" (feature importance). This approach uses a Random Forest Classifier to rank the importance of the features. This ranking can determine the contribution of each feature to the performance of the model. As a result of this approach, the most important features are selected and a new data frame is created. These strategies provide an approach to improve the overall performance of the model, resulting in more robust and optimized results.

## Preprocessing Step

Preprocessing methods play an important role in various machine learning methods to eliminate imbalances in the dataset and handle missing data points. Within the scope of the study, three techniques were used to organize unbalanced classes. First, adaptation to the minority class was achieved by undersampling. With this approach, data points in the majority class are randomly diluted. In this way, the number of data samples belonging to the majority class was adapted to the minority class and the number of data was equalized. In oversampling, synthetic values are created and the minority class is adapted to the majority class. This approach is the exact opposite of the undersampling approach. The number of data of the minority class has been increased by the amount of the majority class. In this way, balanced data was created. As the third technique, a hybrid approach was used. This hybrid approach combines oversampling and undersampling steps. This method applies oversampling to the minority class and a certain amount of undersampling to the majority class. In this way, it equalizes the samples of both data points and creates a balanced data set. Thanks to these pre-processing techniques, the class imbalance problem can be prevented and the generalization ability of machine learning models is increased.

Furthermore, the preprocessing methods applied in this study are effective in addressing missing data points and contribute to the overall robustness and reliability of the machine learning models. The undersampling technique reduces the samples in the majority class and promotes a fairer distribution of data samples over classes. Contrarily, the oversampling method generates synthetic values for the minority class, not only eliminating class imbalance but also increasing the representativeness of the dataset.

The hybrid approach emerges as a subtle strategy that strikes a balance between accommodating the minority class and ensuring that the majority class is proportionately represented by combining elements of both oversampling and undersampling. This combination optimizes the model's ability to distinguish patterns and make accurate predictions, resulting in an improved and balanced dataset.

* 1. Random Forest Classifier

Random Forest is a machine learning algorithm developed based on decision trees. This algorithm can recognize complex relationships and patterns in input data by combining decision trees as shown in Figure 8. Each decision tree in the Random Forest algorithm workflow is trained on a randomly selected subset of data points, and the training results are incorporated into the overall result. Random Forest algorithm takes a result from each of the decision trees and tends to match the results found by the majority to the final results. In this way, it can produce more reliable and accurate results. It also improves generalization ability by considering the decision of multiple trees.

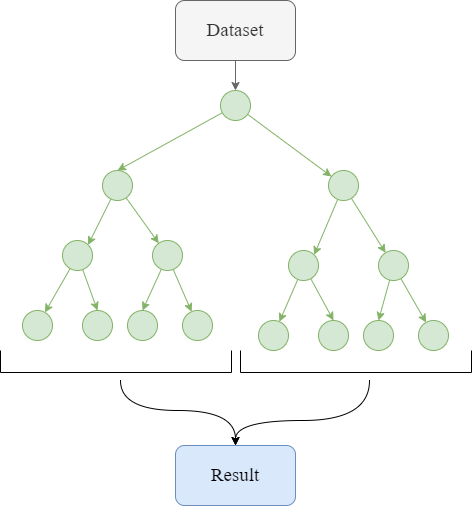


Figure 8 Random Forest classifier diagram.

Random Forest algorithm can successfully perform both classification tasks on categorical data and regression tasks on numerical data. In the classification function, the output of each decision tree provides a probability estimate of a class. By combining these predictions, the likely class for the data point is determined. In the regression function, the average of the values predicted by the trees as output is taken and the output of the algorithm is created.

Since the Random Forest algorithm combines multiple decision trees, results such as overfitting prevention and high accuracy can be achieved. In addition, this algorithm can also be successful on large data sets. Automatic feature selection also provides time-dependent and accuracy-dependent performance in classification and regression functions.

The Random Forest algorithm basically deals with a technique called bagging. Bagging combines the results of multiple learning algorithms trained on samples generated from different data points. With this combination, it aims to increase the performance of the machine learning model. Using this technique, Random Forest reduces the dependency between decision trees and creates more reliable models. Additionally, in the algorithm, the feature set on which each decision tree will be based is randomly selected, ensuring that the trees focus on different features. In this way, the algorithm can provide learning from a wider pool of information. With this feature, overfitting problem can be prevented.

* 1. AdaBoost Classifier and Bagging

Detection of errors that may occur in parts on production lines is immensely significant in quality control processes in the industrial field. For detection, ensemble learning methods such as AdaBoost Classifier and Bagging are highly preferred in improving error detection processes. Many distinctive features make it widely preferred. AdaBoost is based on a focused learning approach. In this way, it can recognize errors much more effectively from complex data sets created from measurements on production lines. AdaBoost starts with one weak learner as shown in Figure 9. It then presents strong learners by sending the output of each weak learner to the next learner. Each learner tends to avoid the mistakes of the previous learner. Thanks to this workflow, AdaBoost, when used to detect parts on the production line, performs updates based on past errors and can recognize future errors more effectively.

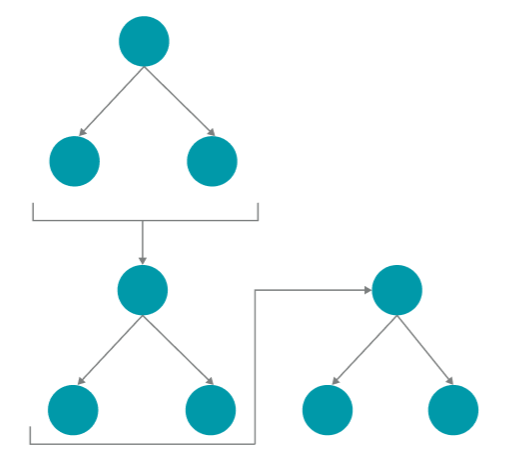


Figure 9 AdaBoost classifier workflow.

AdaBoost uses weighted data attention to focus on complex features and features that are sparsely distributed over the dataset. It can handle a wide variety of varying conditions on the production line. In other words, AdaBoost can perform well on industrial data sets and unstable data sets containing various errors. AdaBoost algorithm has high adaptability. It can adapt to rapidly changing dynamic conditions on the production line. With its update ability, it can quickly react and learn to changes in the production process. AdaBoost method can produce more reliable and effective results in detecting errors on parts on the production line. This ensemble learning techniques also make it possible to optimize quality control in production processes.

* 1. Bagging

The Bagging approach, whose workflow is shown in Figure 10, stands out as an important ensemble learning technique to optimize and improve fault detection processes on the production line. Bagging stands for Bootstrap Aggregating and is an approach that combines models trained on subsets created from different data points. By using subsets of data, it makes it possible to train on different samples. In this way, with this approach, the model learns on a more diverse pool of information and its fault detection capability increases. It can cope with complex features in parts on the production line. Bagging provides robustness against noise in the dataset that can make learning difficult. Training on different subsets of data can reduce random noise. As a result, it provides more reliable and accurate fault detection. On the other hand, this feature can minimize errors due to inaccurate information on measurements performed on parts in the production line.

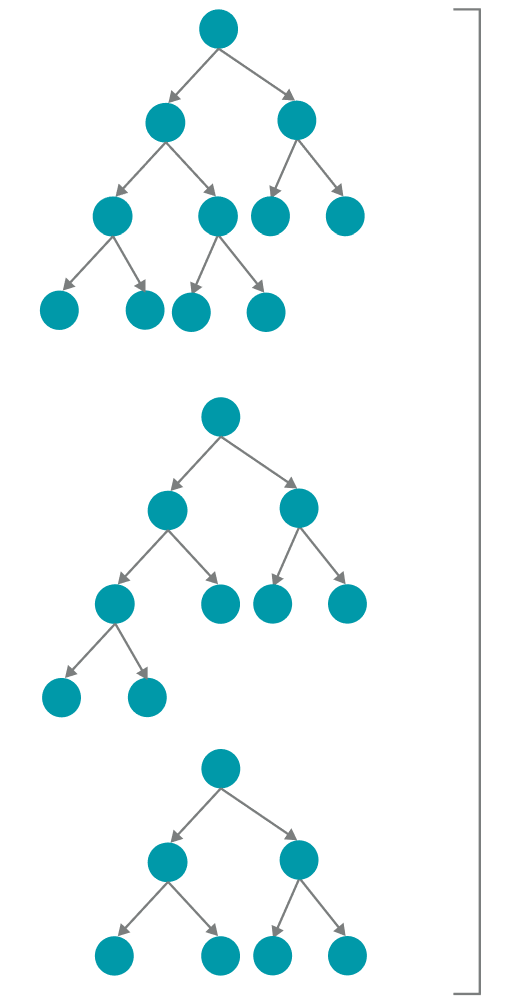


Figure 10 Baggin approach workflow.

Bagging is based on parallel processing of models. Thanks to parallel processing, it performs well in terms of speed on large data sets. This approach, in which models are trained independently in parallel, speeds up the training process. Bagging approach has the ability to reduce the average variance of the models. In this way, it can work with models with high variance. Thanks to this advantage, it can run complex models in parallel without any problems. On the other hand, the flexibility of the Bagging approach to learn by combining various algorithms can enable the selection and optimization of the most appropriate model for the detection of defects that may occur in parts on the production line. The Bagging approach enables the combination of different algorithms such as decision trees, support vector machines and K-NN. In this way, the Bagging approach can adapt to various data structures and different learning situations. Thanks to its various advantages, the Bagging approach can create a powerful model for the detection of defects on parts in production lines. By combining and optimizing models, it can achieve higher and more reliable results. It can make positive contributions to quality control processes.

* 1. Hyper-Parameter Tuning

Light Gradient Boosting Machines (LightGBM) is a powerful machine learning approach. LightGBM can perform well on large datasets with many data points. It has a tree-based approach and can work effectively on both regression and classification tasks. As shown in Figure 11, LightGBM uses a leaf-wise growth approach. This approach adds new nodes at each learning step. This allows for a faster training process and memory management. With its ability to be used in categorical features and its ability to learn various features, it can facilitate data preprocessing stages. However, in order for these models to be compatible with the data, the correct hyperparameter setting is required.

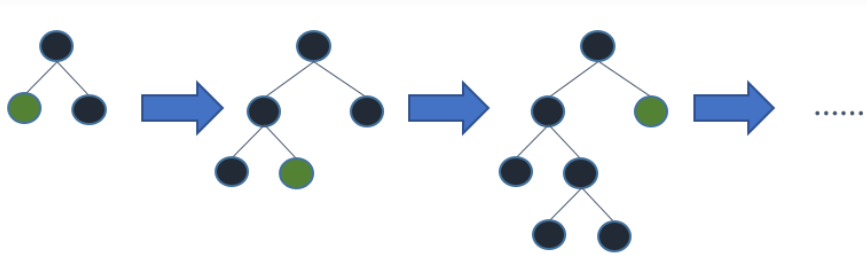


Figure 11 Leaf-wise tree growth.

For hyperparameter tuning, Optuna can provide many advantages. Optuna is an open source library that can be used on models like LightGBM. In the training process of complex models such as LightGBM, it is very difficult to find the right combination of hyperparameters. With Optuna, this process can be automated. Optuna is able to search the domains of specific hyperparameters. The big data obtained on the production line is complex and diverse. There can be many differences between data points. Realizing the hyperparameter tuning of the LightGBM model can make the best use of big data. At the same time, Optuna is able to automate the hyperparameter tuning process, enabling time management.

Optuna's working strategy is based on "Bayesian Optimization". Bayesian Optimization is a probability model that is used to determine the optimum point of all parameters of a function. The pseudo code of Optuna's working strategy is as follows:

# Defining the Hyperparameter Space: def objective(trial):

# Defining a specific range for hyperparameters

param1 = trial.suggest\_float('param1', 0.0, 1.0)

param2 = trial.suggest\_int('param2', 1, 100)

3. metric = train\_and\_evaluate\_model(param1, param2)

4. Reporting the metric value to Optuna: trial.report(metric, step=iteration)

5. Implement bayesian optimization strategy

6. if trial.should\_prune():

7. raise optuna.TrialPruned()

8. return metric

9. Run Optuna: study = optuna.create\_study(direction='maximize')

10. study.optimize(objective, n\_trials=100)

11. Obtaining the optimal hyperparameter combination

best\_params = study.best\_params

best\_value = study.best\_value

In Optuna, first, the hypiparameter area (trial) to be optimized is determined. It is determined in which value ranges the hypeparameter field will be effective. The objective function to be optimized is defined. The objective function has the ability to measure how much the model performance changes over a given combination of hyperparameters and produces a metric as a result of these measurements. The metric returned from the objective function definition provides information about the accuracy and error rate of the model. With the information generated, optuna is run and a certain number of trials are performed. These trials include a combination of different hyperparameters. With the Bayesian Optimization approach, a probability model is created from the past trials. This probability model aims to select the most optimal one among the combinations that have not yet been tried. In this way, optimal hyperparameters are obtained more quickly and efficiently. Finally, Optuna generates the hyperparameter combinations.

## K-fold Cross Validation

K-Fold Cross Validation is an approach used to measure the performance of machine learning models. This method divides the dataset into a certain number of parts as shown in Figure 12. By performing model training on each partition, it allows the performance of the model in different subsets to be evaluated. K-Fold Cross Validation involves iterations where each partition is used as test data. In each iteration, the test set is separated from a certain part of the dataset and the remaining data points are used for model training. With this approach, the reliability of the model can be determined by performing training and testing on various parts of the dataset.

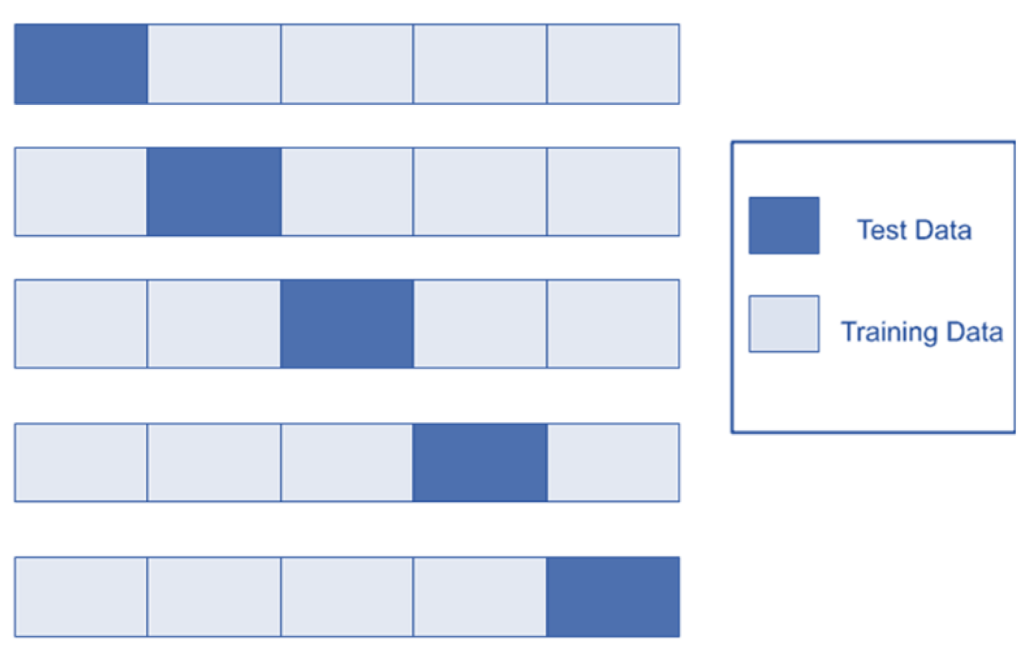


Figure K-fold Cross Validation.

Thanks to the K-Fold Cross Validation method, the data set can be used more effectively. At the same time, the generalization ability of the model, that is, its behavior on different data sets can be examined. These advantages of the K-Fold Cross Validation approach make it easier to identify the overfitting and underfitting problems faced by the model. It can also contribute to model selection, hyperparameter tuning and different evaluation processes.

1. Experimental Results

This section describes the results of machine learning models run to detect defects in parts on the Bosch production line. These results are very important for identifying potential defects and improving quality control at every stage of the production process. These models aim to detect the presence of defects by taking as input the information of parts such as station and measurement. The results provide useful analyses for the early detection of potential defects at each stage of the production line. In addition, machine learning models can automate the defect detection process, thus optimizing the production process.

The fault detection system for parts on the production line is not only realized in terms of quality. It can also function on important issues such as cost. Thanks to the continuous updating of machine learning models, systems that quickly adapt to changes in production conditions can be created.

In the study, the data passed through preprocessing steps are divided into 70% train and 30% test set. The performance of the Random Forest model, which takes this data as input, is measured using various metrics. These metrics are shown in Table 1. Matthews Correlation Coefficient (MCC) metric shows a performance of 0.4525 in the test set. We conclude that the classification performance of the model is moderate. The Random Forest algorithm shows a high performance in the train and test set. This performance is an indication that the model correctly distinguishes the error classes. The accuracy metric shows a similar performance on both data sets, train and test. This indicates that the Random Forest model shows balanced results on both training and test sets. Balanced results indicate that the overall accuracy performance of the model is high. The Random Forest model shows a very high performance on the test set. This is an indication that the model detects errors well on data that it has not seen before. The model is able to provide a good measure of the probability that the samples predicted as positive are actually positive. With the Recall metric, a good Random Forest model shows balanced performance on both train and test sets. With the F1 Score metric, the model also shows balanced results between the precision and recall metrics. These results show that the Random Forest algorithm performs a balanced and reliable detection overall.

Table Random Forest algorithm results on train and test set.

|  |  |  |
| --- | --- | --- |
| **Metric** | **Train Score** | **Test Score** |
| MCC | 0.4818 | 0.4525 |
| ROC-AUC | 0.7337 | 0.7203 |
| Accuracy | 0.7337 | 0.7202 |
| Precision | 0.8085 | 0.7857 |
| Recall | 0.6126 | 0.6058 |
| F1 Score | 0.6970 | 0.6841 |

The results of the AdaBoost algorithm are obtained using various performance metrics as shown in Table 2. According to the results obtained on the training and test sets, it is observed that the AdaBoost algorithm performs better on the training set. With the MCC metric, a performance of 0.835 is achieved on the training set. This metric result indicates that the model shows a very high performance on the training set. It is also concluded that the model has a strong classification ability. When the results measured using the ROC-AUC metric are analyzed, it is seen that the model successfully distinguishes the error classes. However, the model generally performs better on the train set. On the test set, the performance is lower compared to the metrics measured on the train set. The imbalance in the ROC-AUC metrics indicates that the generalization ability of the model is low. This is also observed in the Accuracy metric. While the model shows a very high accuracy performance on the train set, it shows a lower performance on the test set. As a result, the AdaBoost model focuses more on the data in the training set and its performance on the test set is limited. Despite the poor performance of the AdaBoost algorithm on the test set, the results on the test set indicate that the model can be improved for specific fault detection problems. This implies that the model should be subjected to optimization processes against specific errors.

Table AdaBoost algorithm results on train and test set.

|  |  |  |
| --- | --- | --- |
| **Metric** | **Train Score** | **Test Score** |
| MCC | 0.8350 | 0.5495 |
| ROC-AUC | 0.9166 | 0.7745 |
| Accuracy | 0.9166 | 0.7745 |
| Precision | 0.9462 | 0.7861 |
| Recall | 0.8834 | 0.7542 |
| F1 Score | 0.9137 | 0.7698 |

The performance metric values of the results obtained using the Bagging approach are shown in Table 3. The MCC metric shows a performance of 0.4898 on the training set and 0.4763 on the test set. MCC is important as a metric that evaluates the classification performance of the model. The detection performance of the model on the test set is lower than the training set. The low performance results measured by the MCC metric indicate that the model has difficulties in the classification function and cannot adapt to the patterns in the test set. On the other hand, in the light of these results, the Bagging approach has limitations in correctly identifying positive and negative classes. This affects the generalization ability of the model. The ROC-AUC metric for how well the model discriminates the classes is 0.6944 in the test set. According to this metric, the model performs well overall, but has more difficulty in distinguishing classes in the test set than in the train set. The values measured by the accuracy metric are close to each other in the training and test sets. This shows that the model has a balanced performance and can obtain reliable results. The model trained with the Baggin approach shows a very high accuracy metric performance. This high precision performance is an indication that the model is better able to discriminate positive predictions. The Recall metric is calculated as 0.4071 in the training set and 0.4055 in the test set. The F1 Score performance of the model shows a balanced performance in general. While the Bagging model performs well on the training set, it generally performs poorly on the test set. This indicates that the model focuses more on the training data. The obtained results emphasize the need to include optimization processes to increase the generalization ability of the model.

Table Bagging approach results on train and test set.

|  |  |  |
| --- | --- | --- |
| **Metric** | **Train Score** | **Test Score** |
| MCC | 0.4898 | 0.4763 |
| ROC-AUC | 0.6989 | 0.6944 |
| Accuracy | 0.6989 | 0.6944 |
| Precision | 0.9776 | 0.9603 |
| Recall | 0.4071 | 0.4055 |
| F1 Score | 0.5748 | 0.5702 |

Figure 13 shows the ROC curves on the train based on true positives and false negatives. AdaBoost performance on the test set is quite high. Bagging and Random Forest algorithms show an improvable performance.

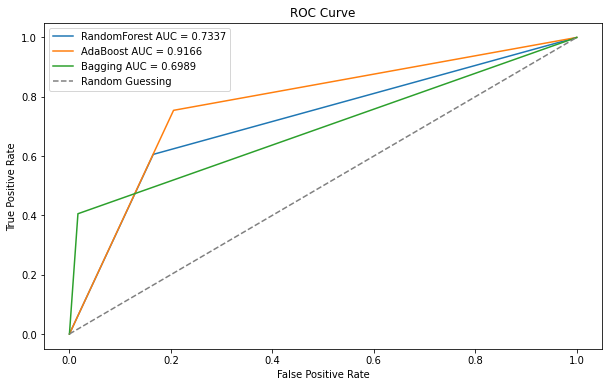


Figure ROC Curve graphic of train set.

Figure 14 presents the ROC curves based on true positives and false negatives on the test set. These graphs show that AdaBoost performance is higher compared to the Bagging approach and the Random Forest algorithm. ROC curves are considered as an important indicator to evaluate the sensitivity and specificity of the model. The higher performance of the AdaBoost algorithm on these results indicates that the model distinguishes between positive and negative classes more effectively. On the other hand, the results indicate that the model has a more robust classification ability overall.

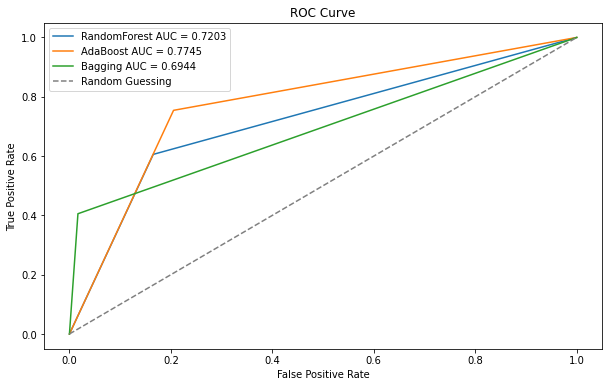


Figure ROC Curve graphic of test set.

The best score and hyperparameter values obtained according to the results of the model optimization performed on the LightGBM algorithm are shown in Table 4. The parameter "n\_estimators" indicates the number of trees selected by Optuna for the algorithm. The maximum depth of the trees is held by the parameter "max\_depth". The proportion of features used in the algorithm for each split is stored in the "max\_features" hyperparameter. This fine tuning process aims to better adapt the model to the training data and increase its generalization ability. The best score value obtained shows that the LightGBM model performs successfully on the problem of fault detection in the production line. The results obtained with hyperparameter tuning contribute to the best performance of the model.

Table LightGBM best score and hyper-parameters results.

|  |  |
| --- | --- |
| **Variable** | **Value** |
| Best score | 0. 5251 |
| n\_estimators | 318 |
| max\_depth | 8 |
| max\_features | 0.7023110476090613 |

1. Conclusions
2. References
3. Horváth, I. (2021). Connectors of smart design and smart systems. AI EDAM, 35(2), 132-150. doi:10.1017/S0890060421000068
4. Rindermann, H & Ceci, SJ 2009, ‘Educational policy and country outcomes in international cognitive competence studies 2, *Perspectives on Psychological Science, vol.* 4, no. 6, pp. 551–568.
5. M, Bryner 2012, ‘Smart manufacturing: The next revolution’, *Chemical Engineering Progress*, vol. 108, no. 10, pp. 4–12.
6. Ashby, WR 1947, ‘Principles of the self-organizing dynamic system’, *Journal of General Psychology* 37, pp. 125–128.
7. Sabatucci, L, Seidita, V & Cossentino, M 2018 ‘The four types of selfadaptive systems: a metamodel’, Proceeding of the International Conference on Intelligent Interactive Multimedia Systems and Services. Cham: Springer, pp. 440–450.
8. Bures, T, Gerostathopoulos, I, Hnetynka, P, Plasil, F, Krijt, F, Vinarek, J & Kofron, J 2020 ‘A language and framework for dynamic component ensembles in smart systems’, *International Journal on Software Tools for Technology Transfer*, vol. 22, no. 4, pp. 1–13.
9. Schirner, G, Erdogmus, D, Chowdhury, K & Padir, T 2013, ‘The future of human-in-the-loop cyber-physical systems’, *Computer*, vol. 46, no. 1, pp. 36–45.
10. Horváth I (2020a) Position statement: how are smart design and smart systems shaking hands? Repository of the Smart Design of Smart Systems Workshop of the SDPS 2020 On-Line Conference, 28 November 2020, pp. 1–38
11. J, Wan, S, Tang, D, Li, S, Wang, C, Liu, H, Abbas & AV, Vasilakos 2017, ‘A Manufacturing Big Data Solution for Active Preventive Maintenance’, *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2039-2047. doi: 10.1109/TII.2017.2670505.
12. Zhong, RY, Xu, X, Klotz, E, & Newman, ST 2017, ‘Intelligent manufacturing in the context of industry 4.0: a review’, *Engineering*, vol. *3*, no. 5, pp. 616–630.
13. Li, B, Hou, B, Wen-tao, Y, Xiao-bing, L, & Yang, C 2017, ‘Applications of artificial intelligence in intelligent manufacturing: A review’, *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 1, pp. 86–96.
14. M. Bryner, “Smart manufacturing: The next revolution,” *Chemical Engineering Progress*, vol. 108, no. 10, pp. 4–12, 2012.
15. Atzori, Luigi & Iera, Antonio & Morabito, Giacomo. (2010). The Internet of Things: A Survey. Computer Networks. 2787-2805. 10.1016/j.comnet.2010.05.010.
16. A. Maurya, "Bayesian optimization for predicting rare internal failures in manufacturing processes," 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, USA, 2016, pp. 2036-2045, doi: 10.1109/BigData.2016.7840827.
17. Li Da Xu, Wu He, and Shancang Li. Internet of Things in Industries: A Survey. *IEEE Transactions on Industrial Informatics*, 10(4):2233–2243, 2014.
18. J. Lee, H. Kao, S. Yang, ”Service innovation and and smart analytics for industry 4.0 and big data environment,” *Procedia Cirp*, vol. 16, pp. 3-8, 2014.
19. C. M. Carbery, R. Woods and A. H. Marshall, "A Bayesian network based learning system for modelling faults in large-scale manufacturing," 2018 IEEE International Conference on Industrial Technology (ICIT), Lyon, France, 2018, pp. 1357-1362, doi: 10.1109/ICIT.2018.8352377.
20. G. Susto, et al.,” Machine learning for predictive maintenance: A mul- tiple classiﬁer approach,” *IEEE Transactions on Industrial Informatics*, vol. 11(3), pp. 812-820, 2015.
21. T. Wuest, C. Irgens, K-D, Thoben,” An approach to monitoring quality in manufacturing using supervised machine learning on produce state data”, *Journal of Intelligent Manufacturing*, vol. 25(5), pp. 1167-1180, 2014.
22. B. Kroll, et al., ”System modelling based on machine learn- ing for anomaly detection and predictive maintenance in industrial plants”, *IEEE Conf. on Emerging Technology and Factory Automation*, Barcelona, Spain, 12 Jan. 2015, pp. 1-7.
23. G. Wang, et al., ”A novel Bayesian network-based fault prognostic method for semiconductor manufacturing process,” in *IEEE Interna- tional Conference on Industrial Technology*, 2017, pp. 1450-1454.
24. Zaslavsky, C. Perera, and D. Georgakopoulos, “Sensing as a Service and Big Data,” *Proc. Int. Conf. Adv. Cloud Comput.*, pp. 21–29, 2012.
25. Moldovan, I. Anghel, T. Cioara and I. Salomie, "Time Series Features Extraction Versus LSTM for Manufacturing Processes Performance Prediction," 2019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD), Timisoara, Romania, 2019, pp. 1-10, doi: 10.1109/SPED.2019.8906653.
26. J. Hagerty, “How Many Turns in a Screw? Big Data Knows,” *The Wall Street Journal*, may 2013.
27. A. C. Einat Ronen, Kenneth Burns, “Utilizing Predictive Sales Analytics with Big Data,” Intel, Tech. Rep., 2013. [Online]. Available: <http://www.intel.com/content/www/us/en/it-management/intel-it-best-> practices/using-big-data-predictive-analytics-to-optimize-sales- paper.html
28. Kaggle, “Bosch Production Line Performance,” 2016. [Online]. Avail- able: [https://www.kaggle.com/c/bosch-production-line-performance](http://www.kaggle.com/c/bosch-production-line-performance)
29. Kang, Seokho. (2021). Product failure prediction with missing data using graph neural networks. Neural Computing and Applications. 33. 1-10. 10.1007/s00521-020-05486-2.
30. Liu, Yintao & Yao, Ke-Thia & Liu, Shuping & Raghavendra, Cauligi & Balogun, Oluwafemi & Olabinjo, Lanre. (2011). Semi-supervised Failure Prediction for Oil Production Wells. 434-441. 10.1109/ICDMW.2011.151.
31. Ge, Ning & Li, Guanghao & Zhang, Li & Liu, Yi. (2021). Failure prediction in production line based on federated learning: an empirical study. Journal of Intelligent Manufacturing. 33. 1-18. 10.1007/s10845-021-01775-2.
32. Mukherjee, Ashok. (2000). The Impact of new manufacturing requirements on production line productivity and quality at a focused factory. Journal of Operations Management. 18. 10.1016/S0272-6963(99)00023-6.
33. Fera, Marcello & Greco, Alessandro & Caterino, Mario & Gerbino, Salvatore & Caputo, F. & Macchiaroli, Roberto &
34. DʼAmato, Egidio. (2019). Towards Digital Twin Implementation for Assessing Production Line Performance and Balancing. Sensors. 20. 97. 10.3390/s20010097.
35. Chen, Chee-Cheng. (2008). An objective-oriented and product-line-based manufacturing performance measurement. International Journal of Production Economics. 112. 380-390. 10.1016/j.ijpe.2007.03.016.
36. Cordero, R., Walsh, S.T., Kirchhoff, B.A., 2005. Motivating performance in innovative manufacturing plants. The Journal  
    of High Technology Management Research 16, 89–99.
37. Ahmad, M.M., Dhafr, N., Benson, R., Burgess, B., 2005. Model for establishing theoretical targets at the shop floor level in speciality chemicals manufacturing organizations. Robotics and Computer-Integrated Manufacturing 21, 391–400.
38. Ahmad, M.M., Dhafr, N., 2002. Establishing and improving manufacturing performance measures. Robotics and Computer- Integrated Manufacturing 18, 171–176.
39. H. Singh, J. Motwani, A. Kumar, “A review and analysis of the state of the art research on productivity measurement” Industrial Management and Data Systems, vol. 100, pp 234-41, 2000.
40. Tangen, S (2002) “A theoretical foundation for productivity measurement and improvement of automatic assembly systems” *Licentiate* *Thesis*, Stockholm: Royal Institute of Technology, pp 19-30.
41. Kusiak, A. (2017). Smart manufacturing must embrace big data. *Nature*, *544*(7648), 23–25.
42. Jerome H Friedman. Greedy function approximation: a gradient boosting machine. Annals of statistics, pages 1189–1232, 2001.
43. Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. A support vector method for optimizing average precision. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pages 271–278. ACM, 2007.
44. David MJ Tax, Marco Loog, and Robert PW Duin. Optimal mean- precision classiﬁer. In International Workshop on Multiple Classiﬁer Systems, pages 72–81. Springer, 2009.
45. Ljubomir Buturovic, Mike Wong, Grace W Tang, Russ B Altman, and Dragutin Petkovic. High Precision Prediction of Functional Sites in Protein Structures. PloS one, 9(3):e91240, 2014.
46. Toon Calders and Szymon Jaroszewicz. Efﬁcient AUC optimization for classiﬁcation. In European Conference on Principles of Data Mining and Knowledge Discovery, pages 42–53. Springer, 2007.
47. J.Z. Shan. Detecting change in data, November 16 2010. US Patent 7,836,111.
48. Li, C., Mahadevan, S., Ling, Y., Wang, L., and Choze, S. A dynamic bayesian network approach for digital twin. In 19th AIAA Non-Deterministic Approaches Conference (2017), p. 1566.