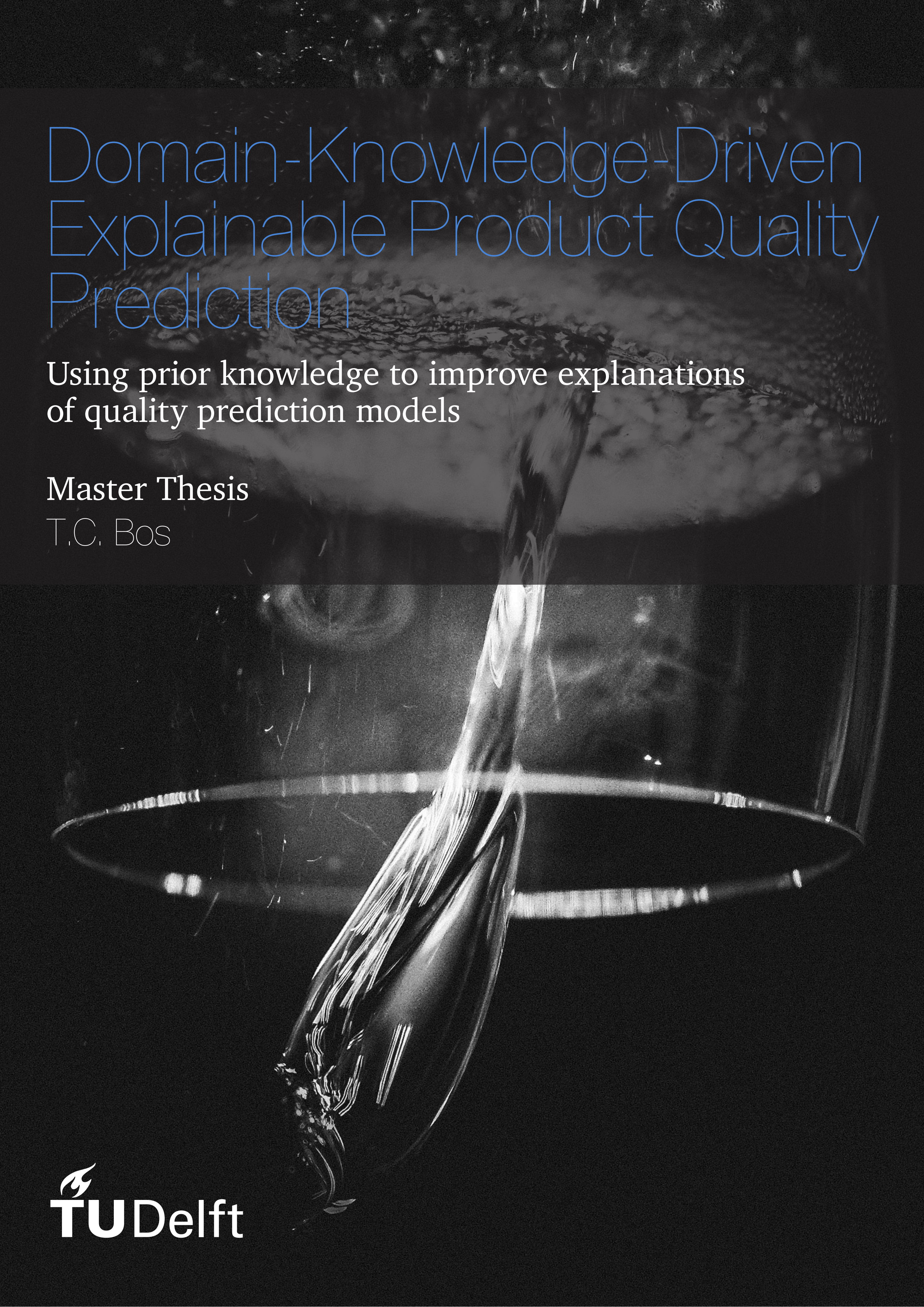
Domain-Knowledge-Driven Explainable Product Quality Prediction

Using prior knowledge to improve explanations of quality prediction models

Master Thesis

T.C. Bos



TU Delft

Domain-Knowledge-Driven Explainable Product Quality Prediction

Using prior knowledge to improve explanations of quality prediction models

by

T.C. Bos

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on June 15th, 2022.

Thesis Supervisor:	Mathijs de Weerdt
Daily Supervisor:	Emir Demirović
Co-Supervisors:	Guido Monchen and Ben Zwerink
Institution:	Delft University of Technology
Place:	Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft
Project Duration:	September, 2021 - June, 2022

Cover Image: *This photo is taken from a photographic exercise: 7 days, one photo a day in black and white to illustrate everyday life, acts of the day.* — Simon Buchou (https://unsplash.com/@simon_buchou)

TU Delft

Preface

This thesis was written to finalize the research done in my internship at Vrumona BV, where the goal was to improve bottle filling processes by applying data-driven technologies.

While I learned many aspects of machine learning during my Computer Science Master's programme, I did not have any prior experience with explainable artificial intelligence. This project was recommended to me by Emir Demirović, and it was during this project that I first started working with the concept of models giving out more information than just their predictions. This idea inspired me so much that I want to continue working on this subject after my studies, and I feel that this may be my most significant gain from this project.

I want to thank my supervisors, Emir Demirović and Mathijs de Weerdt, for guiding me through the process of a thesis project, and helping me see what I was doing well and where I could improve. I then want to thank my co-supervisors, Guido Monchen and Ben Zwerink, for helping me understand the problem from an industrial standpoint, and their hospitality when I was in Zeist. I also want to thank Marcel van de Kant and Jimmy Munnix for helping me with the logistics of the company and the discussions on the intricate details of the bottle filling process.

Finally, I want to express my gratitude to my family and friends, who have shown great support and helped me in doing my best during the last ten intense months.

Thomas Bos
Lelystad, June 2022

Summary

Explainable artificial intelligence has in recent years allowed us to investigate how many machine learning methods are creating its predictions. This is especially useful in scenarios where the goal is not to predict a variable, but to explain what influences that variable. However, the methods that have been created thus far do not focus on specific domains and only give scientific relations between variables. In the case of a production process, there exist a multitude of related variables, constrained in different ways, thus requiring a more sophisticated method than those which are universally applicable. The target of this research is a bottle filling machine at a large carbonated drink manufacturer, of which a great amount of data has been stored. The product quality, in this case the amount of CO_2 in the filled bottles, is subject to high amounts of fluctuation, leading to a lot of waste. To further contextualize the problem of bottle filling, an artificial dataset is created to serve as a basis for visualization and evaluation. Using literature on carbonated drink filling, multiple hypotheses were then formulated, independently from the thesis company, as a target for the evaluation process. To address the problem of determining what factors influence bottle CO_2, a methodology is proposed which aims to provide explainable quality prediction of production process output, while using available domain knowledge to increase the plausibility of explanations. The methodology makes use of counterfactual explanations for tree ensembles and process state forecasting to show how a process state can be changed to yield better product quality. Evaluation on the artificial and real datasets show that the methodology is able to effectively predict the amount of CO_2 in a bottle, while giving changes to the current process state which improve the product quality. It is also shown that the results produced by the methodology are in line with the formulated hypotheses. Bottle CO_2 is influenced greatly by temperature changes in the filler and these temperature changes are caused by flow of cooled lemonade through the process. As these flows are dependent on the filling speed, irregular filling is suspected to be the root cause for the bottle CO_2 fluctuations.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
CFE	CounterFactual Explanation
IML	Informed Machine Learning
LIME	Local Interpretable Model-agnostic Explanations
MIP	Mixed Integer Programming
MAE	Mean Absolute Error
ML	Machine Learning
MSc	Master of Science
MSE	Mean Squared Error
SAT	boolean SATisfiability
SHAP	SHapley Additive Values
SMT	Satisfiability Modulo Theory
VIF	Variance Inflation Factor
XAI	Explainable Artificial Intelligence

List of Figures

List of Tables

1 Introduction

With Machine Learning (ML) being at the forefront of the Industry 4.0 revolution [16], more companies will want to invest into optimizing their industrial pipelines using ML models, as they are able to make automated predictions using the large amount of data they are currently generating. These predictions can range from determining when machine parts will fail (Predictive Maintenance), to automated warehousing as done by Tesla[1]. A specific area in industrial settings where ML can be beneficial is in the modelling of production processes [69]. In some cases these processes are monitored by thousands of sensors measuring temperatures, pressures, flows, chemical contents, etc., creating complex multivariate time series problems [60].

Creating *hard models* of such processes, models based on the physical nature of the system, requires extensive knowledge about the many physical laws that apply to such processes and therefore this is very difficult to approach for non-experts [24]. Even if the knowledge is available, the cost of creating such intricate methods already asks for a more convenient solution. If this knowledge is not available, for instance due to retiring expert modellers [6], *soft models* are able model the process using the many data sources from processes which are usually already stored, such as process parameters, alarms, and sensors (e.g. temperature, pressure, flow) [19]. This is not without problems as in a production process, such as the bottle filling machine targeted by this research, setting measuring the quality of a sample may be destructive, making excessive sampling undesirable. This lack of samples prevents the use of very flexible ML (e.g. time series-based prediction or neural network) models which would overfit [77, 18].

Furthermore, in an industrial process setting soft models can quickly become a *black-box* model when they get more complex. Black-box models are *ubiquitous opaque decision systems* which do not show how its predictions are created. While such a model perfectly predicting the quality of a final product can be beneficial, it will never indicate that there is something inherently wrong with the process itself, as that cannot be conveyed through only an indicator for the quality of a product. Even worse, the model could be right for the wrong reasons, which will not become clear without further investigation. Ideally, the model can give explanations to how its predictions are constructed, in this case allowing its users to determine possible faults in the modelled production processes.

> All models are wrong, but some are useful. *G. E. P. Box [5]*

The field of Explainable Artificial Intelligence (XAI) aims to create ML models which are able to show how it is creating its predictions [17]. These models can then be used to uncover the hidden workings of the modelled process and allow for process optimization [69]. This will also allow companies to deal with situations where there is not enough knowledge transferring between retiring experts and new personnel, as new personnel can potentially learn the workings of machines through the model.

While being able to create a model from only data obtained from process monitoring is useful, it is not necessarily true that there is no prior knowledge of the process at all. In this case, we may want to be able to use this *domain knowledge* to improve the model, perhaps in a combined effort with data analysts and process engineers. Furthermore, there are circumstances where purely data-driven approaches do

[1] `https://www.mouser.com/blog/industry-40-supported-by-machine-learning`

not lead to satisfactory results in the first place. This can occur when not enough data is available, or when predictions do not adhere to logical constraints [58]. The field of Informed Machine Learning (IML) aims to incorporate prior knowledge about the problem into a ML model to improve its predictions and explanations [58].

1.1. Problem Statement

When one desires to create explainable ML models to a production process there is a problem which is highlighted by recent studies: there is a disconnect between the development of XAI methods and the *stakeholders' desiderata*, i.e., their goals, expectations, needs, and demands [34]. Instead of providing *scientific explanations*, more focus should be laid on giving *everyday explanations* which can be understood by the stakeholders outside of the field of ML [46, 34, 71]. In the case of this research the stakeholders should be the process engineers which have the knowledge about the process, enabling the methodology to harness their knowledge directly and provide explanations which they can directly understand.

Current staples in XAI such as SHapley Additive Values (SHAP) [41] and its model-specific variants [42] may be easily applied in projects, but are limited due to their very general nature in that they only provide exactly those *scientific relations* which on their own may not be useful enough to other stakeholders than model developers. An example where SHAP is applied to power grid data states that SHAP is able to show unexpected non-linear relationships between variables, but that it needs to be combined with domain knowledge to create real scientific discoveries. For explanations to be *persuasive*, i.e., be useful to the user, explanations need to be presented in a way that they can understand, and this is currently not researched enough [74, 46].

1.2. Domain Knowledge Driven XAI

The objective of this research is therefore to create a methodology that when applied to a production process uses the domain knowledge of that process to improve performance of the model explanations and, more importantly, give explanations which are aimed at the process engineers instead of the model developers.

1.3. Research Questions

Main Research Question

> How can a machine learning model be developed which predicts product quality, while giving production-process-specific explanations that domain-specific stakeholders can understand?

Sub-questions

- Can random forests be used to indicate what variables influence product quality?
- Can counterfactual explanations be used to suggest a better production process state?
- Can time series forecasting be used to forecast the influence of process parameter changes?
- Can domain knowledge be used to make counterfactual explanations more plausible with respect to production processes?

1.4. Reading Guide

After this introduction we will first be covering the context of the project in Chapter 2 by introducing the thesis company and the problem this research addresses. In Chapter 3 we revisit the three fields of process modeling, explainable artificial intelligence, and domain knowledge integration, we established in the introduction, and go over the recent literature done and the research gaps that need to be addressed. We will also cover the literature that is closest to this research and determine what it is missing. We then discuss the preliminaries in Chapter 4 by more formally introducing production processes, quality modeling, random forests, and counterfactual explanations. This then serves as a basis when the contributions by this research are introduced in Chapter 5, where we will also formulate some hypotheses regarding the target bottle filling machine. In Chapter 6 the results of applying the proposed methodology to the artificial and real datasets introduced in Chapter 2 are given and discussed and we will try to confirm the previously stated hypotheses. Finally, Chapter 7 concludes this research and gives directions for future work to based on the findings and limitations of this research, while also taking the time to give the thesis company directions on where to look for improvements to their process.

2 Context

We will now discuss the context in which this research has been executed. First we will introduce the thesis company, being a large soft drink manufacturer in the Netherlands, aiming to reduce its waste using better quality control. We will then discuss the aspects we need to consider when trying to improve production quality in a drink manufacturing setting, as well as the prior knowledge about such processes. Then an artificial dataset is introduced to illustrate the issue this research is aimed to resolve. Lastly, the available data is discussed together with some preliminary data exploration.

2.1. Soft Drink Supplier

The thesis company is a large soft drink supplier that produces and bottles carbonated soft drinks in-house. They are interested in how they can improve the efficiency and reduce the waste of their processes using state of the art data driven technologies. The initial target of this project is one individual bottle-filling process of which a simplified schematic is shown in Figure 2.1. This process consists of a stage where the lemonade is made, a carbonation stage where lemonade is mixed with CO_2, and the filling stage where the carbonated lemonade is filled into bottles.

2.2. The Problem of Bottle Filling

However, the filling of bottles is subject to many variables, mostly unknown to the thesis company before the start of this project, that induce fluctuations in the amount of CO_2 that ends up in the bottles. This is bad as the accepted amount of CO_2 lays within narrow bounds. Furthermore, determining the CO_2 of a bottle is a manual and destructive process and thus bottle samples are usually taken only hourly, which is too great of a window to stay on top of possible changes in bottle CO_2. The goal of this project

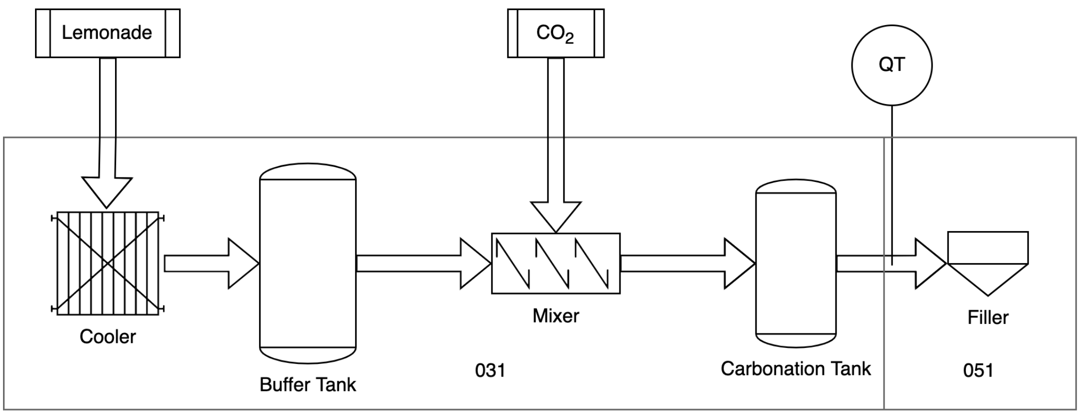


Figure 2.1: A simplified schematic of the bottle filling machine that supplied the data to test the methodology proposed in this research.

is then to create a methodology predicting the bottle CO_2 while aiding the process engineers by giving explanations that expose the inner workings of the bottle-filling process.

2.2.1. What is Known About Bottle Filling

There are a number of facts known about the problem. First of all, the calculation of the amount of CO_2 in a bottle is a function of its temperature and pressure. Higher temperature and lower pressure yield lower CO_2 measurements, while lower temperature and higher pressure yield higher CO_2 measurements [66]. Besides CO_2 setpoints, the amount CO_2 in bottles should therefore be sensitive to changes in temperature and pressure as well which is important to keep in mind.

2.2.2. Unique To The Target Process

When considering the bottle filling machine at the company there are a number of aspects that may be of influence. First of all, the lemonade is only cooled before the carbonation process, meaning that the lemonade in the process will generally increase in temperature after that point. The faster lemonade flows through the process, the less of an effect this aspect will have, but it seems logical to assume that when the bottle filler has been idling for a long time the temperature increase can be a problem. Furthermore, no measurement of CO_2 levels is done directly before bottles are filled, making it more difficult to determine exactly where problems lie. Unfortunately, the company has no process models and has only performed parameter optimization through physical testing.

2.2.3. Illustration Using a Toy Example

In order to illustrate the problem of lemonade increasing in temperature when stationary in the process, we will use a manually constructed artificial dataset. The structure of this hypothetical bottle filling machine is **incoming lemonade $\rightarrow$ carbonation tank $\rightarrow$ filler tank**, essentially the carbonation tank and filler from Figure 2.1. Thus, it is assumed that the carbonation has already happened. For a given time step t a value (e.g. temperature) $x_{i,t}$ of section i (carbonation tank or filler tank) is determined using the previous value $x_{i,t-1}$ and the previous value $x_{j,i-1}$ for the section j preceding it.

It is assumed that because the carbonation tank has a larger volume than the filler tank, it will take longer to warm up and cool down to room temperature. The CO_2 in a section is determined by the following (non-factual) function inspired by the relations between CO_2, temperature, and pressure as described in [66]:

$$\text{bottleCO2}(T, p, m, T_{ideal}, p_{ideal}) = \max(0, m \cdot (1 - 0.1 \left(\frac{T}{T_{ideal}}\right)^3 + 0.2 \left(\frac{p}{p_{ideal}}\right)^{\frac{1}{3}})), \tag{2.1}$$

where T is the temperature, p the pressure, m the amount of CO_2, and T_{ideal} and p_{ideal} are the ideal temperature and pressure at which the amount of CO_2 should be maximized. A section of the resulting data can be viewed in Figure 2.2. Say we now have a trained a black-box ML model creating the predictions in Figure 2.3. Although the predictions of the model can be used to roughly estimate the current amount of CO_2 in the produced bottles, we have gained no knowledge about the process we are studying. Ideally we have a model which, for example, explains what causes the small fluctuations in the quality predictions around 300 time steps, or why the prediction suddenly drops around 440 time steps, or how we get the quality predictions up again when we find ourselves in a situation such as at 550 time steps. An example of a more transparent model is the very simple hand-made *decision tree* model seen in Figure 2.4. For each prediction, the exact reasoning of the model can be easily followed, in contrast to black-box models where that reasoning is not that easily known. In this case, the high filler and room temperatures are the cause for the low prediction. Ideally, this model is automatically generated so that the model decisions can be extracted, giving insight into the workings of the process. However, it is also important to keep in mind that we do not have this amount of samples in the real dataset due the destructive nature of bottle sampling.

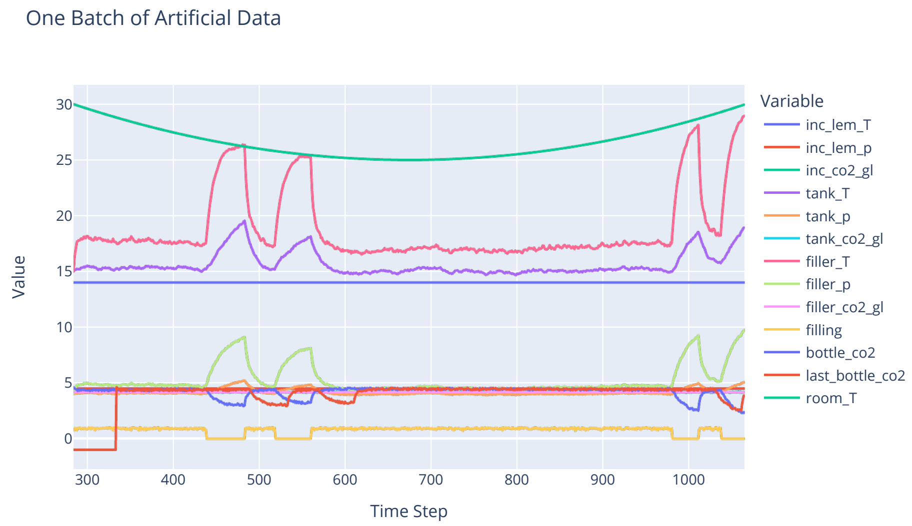


Figure 2.2: Time-series data of the manually constructed toy dataset viewing the temperatures (T) and pressures (p) of the incoming lemonade, carbonation tank, and filler tank, room temperature, bottle CO_2, and whether filling is done.

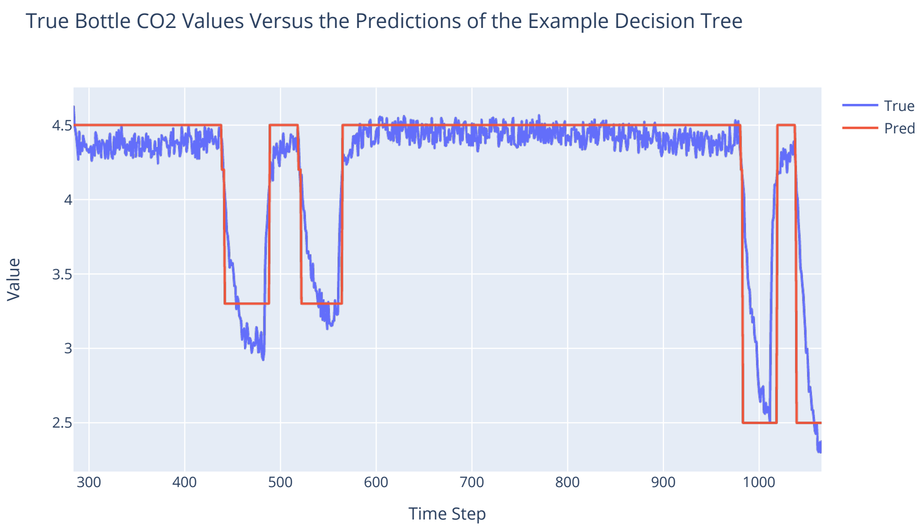
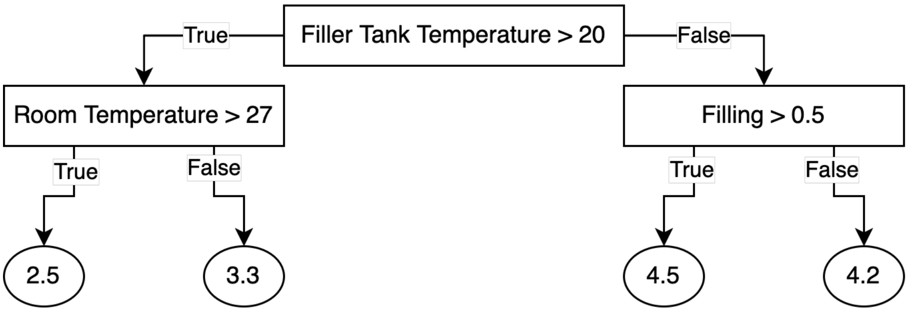


Figure 2.3: Example predictions created using the hand made model from Figure 2.4.




```
def simple_predictor(x):
    if x['filler_T'] > 20:
        if x['room_T'] > 27:
            return 2.5
        else:
            return 3.3
    else:
        if x['filling'] > 0.5:
            return 4.5
        else:
            return 4.2
```

Figure 2.4: Example decision tree model (top), with Python syntax (bottom), for predicting the bottle CO_2 on the artificial dataset.

2.3. Data

The data available consists of time-series data of many temperature, pressure, and flow sensors, as well as when valves open and close, and process set points. In this report the names of the parameters and sensors are used as they appear in the dataset. Table 2.1 gives clarification on what each of the names represents. Important in this work is the distinction between process parameters and sensors. We consider as process parameters all measurements of flow (`FT` or `FC`) and the temperature measurement directly following the cooling of lemonade (`TT_031_130_033`), as these values can be adjusted. We then have hourly bottle samples as target data. All data is separated in batches as the bottle filling machine is usually only producing for at most two days before it changes recipe or stops for cleaning. In this project we will be focusing on time series data with a resolution of 10 seconds, where each batch is defined as the slice of data from 15 minutes before the first bottle sample to 15 minutes after the last bottle sample. Appendix A contains a number of plots of raw values of one batch. In this section we will perform some data analysis to determine whether the data shows any strong relations, or if it may be easily separable.

Sensor	Measurement
F*	Flow measurement sensor. SETPOINT indicates the process variable set during production.
PT*	Pressure sensor.
T*	Temperature sensor.
V*	Valve state. Either closed (0) or open (1).

Table 2.1: Meaning for each of the parameter and sensor name types.

Name	Description
xx_031_xxx_xxx	Carbonation section of the process
xx_051_xxx_xxx	Filling section of the process
FT_031_140_037	Incoming CO_2 flow to the mixer
FT_031_130_001	Lemonade flow to the cooler
FT_031_160_041	Lemonade flow to the mixer
FT_031_135_041	Lemonade flow to the filler
FC_031_140_010	Lemonade flow to carbonation tank
bottle_co2	Target. Volume percentage CO2 in a bottle.
last_bottle_co2	Target. Previous bottle sample in the same batch (-1 if none yet).
QT_031_140_083	Multi-purpose inline CO_2, temperature, and pressure measurement between the carbonation tank and filler.

Table 2.2: Descriptions for some important names in the dataset.

2.3.1. Demonstration of Noisiness in the Bottle Sample Data

To further clarify why predicting the bottle CO_2 is not trivial, figures are created which plot the bottle sample CO_2 values against the inline CO_2, temperature, and pressure measurements in Figure 2.5. According to the fact that bottle CO_2 is a calculated as function of those aspects of the lemonade, this means that we could expect to see a pattern in these plots. However, this is not the case as none of the plots show any clear relations between the variables. This means that we need to look further to determine why the bottle CO_2 fluctuates.

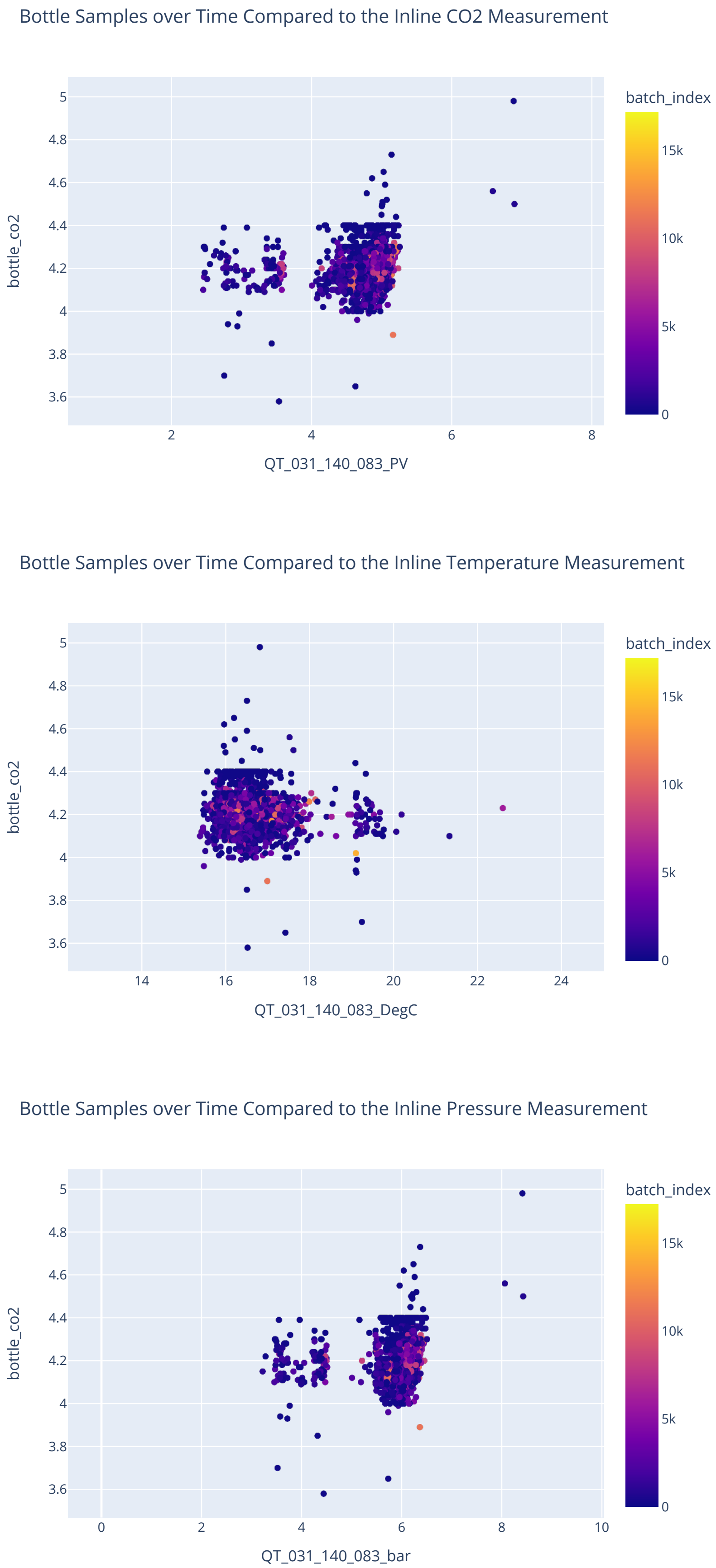


Figure 2.5: Bottle CO_2 plotted against the three inline measurements. The points are coloured with the batch index which is equal to $\frac{\text{seconds since batch start}}{10}$. The fact that there are no clear patterns indicates that the prediction of bottle CO_2 is subject to other factors yet to be determined.

2.3.2. Correlation Analysis

Due to the structure of the bottle filling machine we can expect that some sensors are highly correlated. To determine whether the data has any strong correlations between its variables, a correlation matrix was generated which can be seen in Figure 2.6. Immediately clear is the positive correlation between all the temperature sensors and the negative correlation between the pressure sensors and the temperature sensors. There is also very strong correlation between the flow into the reservoir tank (`FC_031_140_010`) and the incoming lemonade flow (`FC_031_160_041`) which can be explained by the fact that the sensors closely follow each other. Also interesting is the fact that incoming flow of CO_2 (`FC_031_140_037`) positively correlates with all temperature sensors. A possible explanation is that due to the negative relation between temperature and bottle CO_2, operators will notice the drop in bottle CO_2 due to higher temperatures and compensate with higher CO_2 setpoints. An interesting observation can be made is that all three inline measurements seem to correlate very similarly to other sensors, even though they are supposed to be measuring different aspects of the lemonade. This may indicate that this sensor may not be working as we would expect. For instance, the inline temperature seems to correlate negatively with all other temperature measurements which does not seem logical. Finally, the level of the buffer tank (`LT_031_160_016`) is negatively correlated with most of the temperatures, which can be explained by the idea that when this tank is being emptied, cold lemonade flow is being run through the process, lowering temperatures in the pipes and tanks.

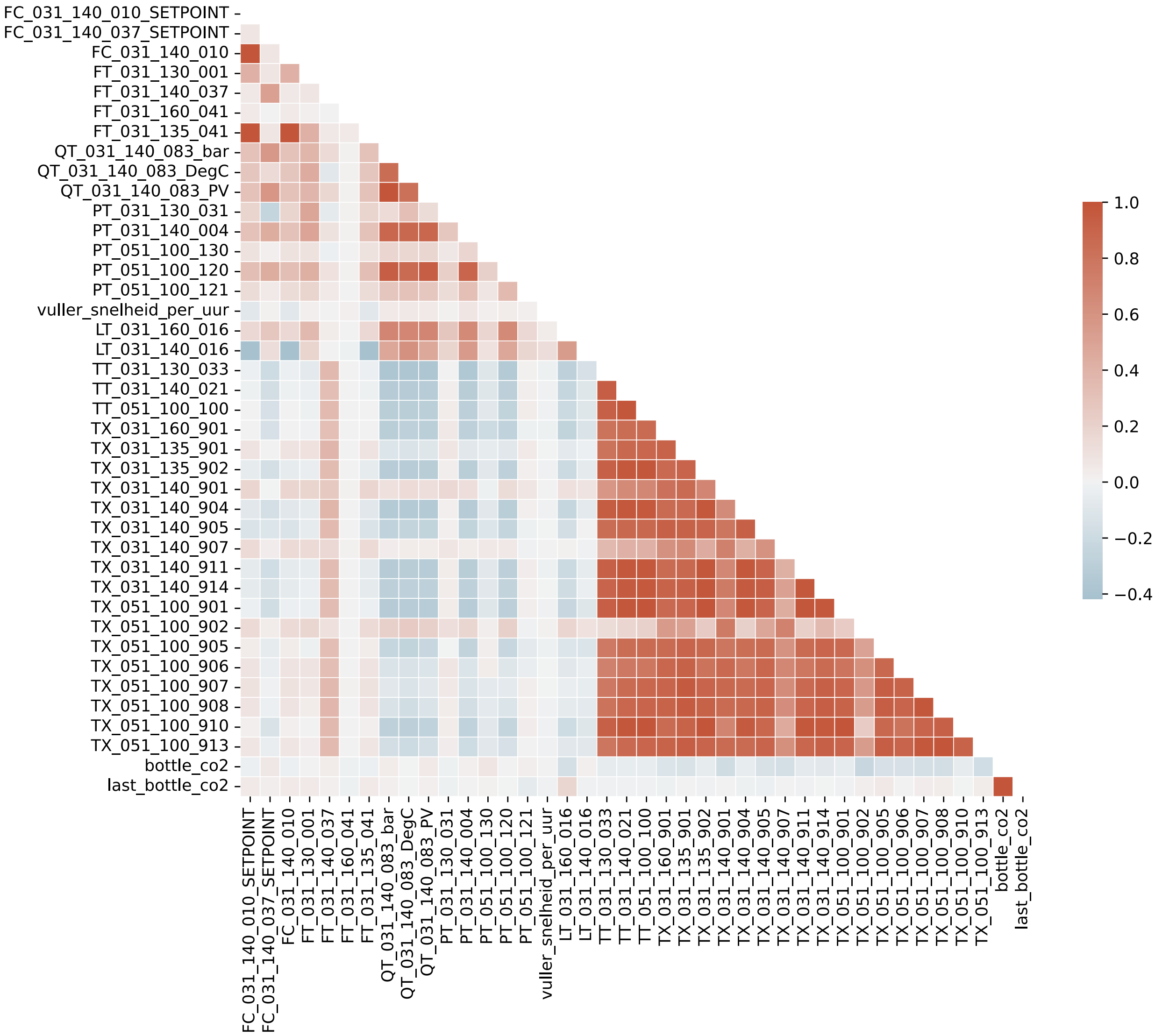


Figure 2.6: Correlation matrix for all of the variables in the dataset.

2.3.3. Regression Analysis

In order to determine whether the prediction of bottle CO_2 can be trivially predicted or that prediction requires more complex models we perform linear regression analysis using the statsmodels library [61]. From the correlation matrix in Figure 2.6 it was clear that some features are highly correlated. To reduce this *multicollinearity* in the dataset, features with a Variance Inflation Factor (VIF) greater than 10 are removed one at a time until all features have a VIF less or equal to 10 [45]. The results of fitting a linear regression model to the remaining features can be seen in Table 2.3 and the effects of each feature can be seen in Figure 2.4. With a low R-squared of 0.265 it is apparent that more complicated non-linear models should be considered to capture more of the variance in the bottle samples. The significant effects are (in order): flow into the carbonation tank and cooler, the pressure coming out of the cooler and to the filler, the buffer tank level, and multiple temperatures on lemonade pipes and CO_2 flow for filler tank pressurization. The fact that the buffer tank level is heavily negatively correlated with the bottle CO_2 is an interesting observation. As we discussed before, filling speed lowers the level of this tank, which in turn lowers temperatures across the bottle filling machine, which should then in theory increase the bottle CO_2 and it seems that the linear model agrees with this theory. In Chapter 5 we will go over these points again and formulate hypotheses to be tested by the proposed methodology.

Dep. Variable:	bottle_co2	**R-squared:**	0.265
Model:	OLS	**Adj. R-squared:**	0.256
Method:	Least Squares	**F-statistic:**	29.65
		Prob (F-statistic):	2.77e-118
		Log-Likelihood:	-2634.6
No. Observations:	2083	**AIC:**	5321.
Df Residuals:	2057	**BIC:**	5468.
Df Model:	25		
Covariance Type:	nonrobust		

Table 2.3: OLS Regression Results on the complete dataset with features subject to collinearity removed.

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	6.898e-14	0.019	3.65e-12	1.000	-0.037	0.037
FC_031_140_010	0.1009	0.044	2.316	**0.021**	0.015	0.186
FT_031_130_001	-0.0526	0.022	-2.413	**0.016**	-0.095	-0.010
FT_031_140_037	0.0388	0.058	0.673	0.501	-0.074	0.152
FT_031_160_041	0.0154	0.019	0.814	0.416	-0.022	0.053
QT_031_140_083_bar	0.4266	0.053	8.075	**0.000**	0.323	0.530
PT_031_130_031	0.1796	0.024	7.345	**0.000**	0.132	0.228
PT_031_140_004	0.0016	0.023	0.070	0.944	-0.044	0.048
PT_051_100_130	0.1456	0.029	5.090	**0.000**	0.090	0.202
PT_051_100_120	-0.0652	0.021	-3.162	**0.002**	-0.106	-0.025
PT_051_100_121	0.0143	0.021	0.674	0.501	-0.027	0.056
vuller_snelheid_per_uur	-0.0097	0.019	-0.499	0.618	-0.048	0.028
LT_031_160_016	-0.2120	0.025	-8.356	**0.000**	-0.262	-0.162
LT_031_140_016	0.0509	0.040	1.276	0.202	-0.027	0.129
TT_031_130_033	-0.0430	0.033	-1.318	0.188	-0.107	0.021
TT_031_140_021	0.1083	0.055	1.960	0.050	-7.02e-05	0.217
TT_051_100_100	0.0564	0.039	1.441	0.150	-0.020	0.133
TX_031_160_901	0.0899	0.052	1.745	0.081	-0.011	0.191
TX_031_135_901	-0.1258	0.054	-2.334	**0.020**	-0.231	-0.020
TX_031_140_904	0.1237	0.043	2.850	**0.004**	0.039	0.209
TX_031_140_907	0.1153	0.039	2.944	**0.003**	0.038	0.192
TX_031_140_911	-0.1013	0.037	-2.723	**0.007**	-0.174	-0.028
TX_051_100_905	0.0873	0.040	2.184	**0.029**	0.009	0.166
TX_051_100_906	-0.0254	0.041	-0.619	0.536	-0.106	0.055
TX_051_100_910	0.0832	0.056	1.476	0.140	-0.027	0.194
TX_051_100_913	-0.5096	0.058	-8.755	**0.000**	-0.624	-0.395

Table 2.4: The effects of each feature on the regression model. Significant ($P < 0.05$) effects have been made bold.

3
Literature Review

In this literature review we will be investigating relevant literature with respect to the three fields that this research aims to bridge: production process quality prediction, explainable artificial intelligence, and domain knowledge integration. Finally, we will investigate the literature that comes closest to the goals we set out in this project.

3.1. Production Process Quality Prediction

Predicting the produced quality of a production process can be done in three ways: modelling using a priori physical and mathematical knowledge about the process, modelling using observed using data obtained from the process (e.g. sensor data), and hybrid modelling which combines both methods [76]. While there is great progress in the field of simulating multiphysics problems, creating these models is often a very complex process [54, 68] and may even be impossible in the case of missing, gappy or noisy boundary conditions [30]. Data-driven approaches eliminate the need for this complex knowledge and allow for the extraction of the needed process information directly from the recorded process data [76].

In [68], a methodology using the popular Partial Least Squares (PLS) regression on sensor data is proposed for predicting quality of diesel distillation. It was shown that their method can obtain similar performance to a Neural Network (NN) based model while being more transparent due to its ability to determine variables which contribute to better quality. However, the methodology or hyper-parameters used to construct the NN were not given. Another widely used regression method is Multiple Linear Regression (MLR) which relates the input variables to the target variable in a linear polynomial fashion, but it fails if the input data is poorly designed, for instance, due to multicollinearity [68]. Therefore extensions such as Principal Component Regression (PCR), which focuses more on the variance of the input variables, and Ridge Regression (RR), which focuses more on regression coefficients, can be used [44].

However, a problem with these models is that they do not consider the dynamic properties of processes, which led to the creation of Dynamic PCA (DPCA) which considers time-lagged features to incorporate process dynamics into its target predictions [36].

More recently, work has been done to deal with the fact that quality variables are hard to measure, for example, due to lab analysis time and cost, leading to large amounts of unlabelled data. Supervised methods which expect fully labelled data can therefore not be used on the entire dataset. Manifold Regularization (MR) was created to allow supervised methods to be used on partially labelled datasets, turning them into semi-supervised methods by assuming that if data points are close to each other, their respective labels should be similar as well. Semi-supervised Probabilistic Principal Component Regression (SSPPCR) [15], Co-training Partial Least Square (Co-PLS) [2], Semi-Supervised Hierarchical Extreme Learning Machine (SS-HELM) [75], and SS-PdeepFM [54] were then developed to apply MR in a product quality setting.

The Dynamic Probabilistic Latent Variable Model (DPLVM) was created to model dynamic processes by extending Dynamic Bayesian Networks (DNBs), used to model stochastic nature of the process, with Linear Dynamic Systems (LDSs) which describe dynamic relationships between process variables [14].

3.2. Explainable Artificial Intelligence

When discussing explainability or interpretability of models it is important to first determine the difference between them. In [57] explainable models create details or reasons to explain its behaviour, while interpretable models are transparent in their functioning by design. Explainability is needed when the task at hand requires models which are too complex to be interpretable [57]. Both interpretable and explainable models can be very beneficial for many reasons. Firstly, interpretable models are useful to model builders as they give greater insight into the training process, allowing the tuning of the model to better fit the task, and secondly, model explanations allow end-users to guide their future actions [38]. An example in the context of this project can be that a model indicates certain sensors as being very important to its decisions, giving technical personnel reason to further investigate. When the goal is to create a model which gives explainable decisions there are two approaches: post-hoc interpretability and inherent interpretability. Post-hoc interpretable methods can be used to obtain explanations from a previously trained existing (black-box) model, while inherently interpretable models are able to give explanations due to their interpretable design and functionality [38].

3.2.1. Post-hoc Explainability

Post-hoc methods aim to extract information from already learned models and do not precisely depend on how the models work, i.e., they consider the model to be black-box. These methods are therefore useful if the model's complexity limits the ability of to trace the logic behind predictions [11].

LIME One of the first model explanation frameworks is Local Interpretable Model-agnostic Explanations (LIME) [55] which is able to explain a classifier or regressor by explaining a subset of individual instances. LIME builds on the idea that a model can be locally approximated using an explainable linear model (e.g. K-LASSO in their work) and thus it is model-agnostic. An explanation then consists of the feature importances for each feature used in the prediction of a single model input [10]. A trustworthiness metric is then given as the amount of predefined 'untrustworthy' features present in an explanation and it was shown that LIME performs better than some existing explainable models [55]. Although LIME has limitations with respect to models which are hard to linearly approximate locally, it paved the way for further advances in explainable artificial intelligence. Furthermore, more work is needed to make the explanations LIME gives more explainable, as understanding them can be a frustrating process, especially considering that the explanations are prone to misinterpretation [10]. Regardless of these limitations, LIME has been successfully used in an industrial setting to explain the prediction of remaining useful life of test beds [64].

SHAP A very popular example-based post-hoc method is SHAP [41]. This method calculates, independently of the structure model, the Shapley values [70] of each subset of features to determine feature importance to the output of that model. Furthermore, SHAP was created such that it satisfies missingness, consistency, and local accuracy [41] and it is the only feature attribution method to do so [63]. After its inception, SHAP has been applied to many time series forecasting problems [49, 33, 28, 31] where the built in visualization and explanation tools of the SHAP framework[1] were used to explain predictions and show relations between features. However, a downside of the SHAP method is that for one prediction it has to evaluate the model on every subset of features, which can be costly [33]. Furthermore, feature importances and attributions may be useful to the model developer, but if the goal of the explanations is to inform domain experts more steps are needed to fit these explanations to their needs and understanding.

3.2.2. Inherent Explainability

While post-hoc methods aim to extract explanations from a model by observing its predictions, there are also methods that build upon the fact that the model itself is transparent. The advantage of these methods is that there is a guarantee that the explanations represent the actual decision making process of the model.

Explanations for Tree-based Models Decision trees, which will be covered in more depth in Chapter 4, are an interpretable, yet flexible and robust machine learning method which are constructed by repeat-

[1] `https://github.com/slundberg/shap`

edly splitting data into more homogeneous groups [9]. An example decision tree can be seen in Figure 2.4. Due to this graphic nature of these trees, small trees can be easily interpreted. Larger trees can still be analysed using SHAP and its model-specific implementation called TreeSHAP, which is specifically made for decision tree methods such as XGBoost or scikit-learn's tree-based models [42]. TreeSHAP uses the actual structure of the decision trees to determine how explanations are made, in contrast to SHAP which only considers the model predictions themselves. Currently, TreeSHAP supports feature contribution, feature importance, feature dependence, and interaction plots for a number of tree-based libraries.

In [26] MaxSAT solvers are used to generate Prime Implicant (PI) explanations for tree ensemble predictions $f(\mathbf{x})$, which represent the minimal subset $\mathcal{X} \subseteq \mathcal{F}$ such that for a different $\mathbf{x}'$ the prediction $f(\mathbf{x}')$ will be the same as $f(\mathbf{x})$ long as they share the values for each feature in $\mathcal{X}$. A similar SAT approach is proposed in [27] where tree ensemble predictions $f(\mathbf{x})$ are instead explained using 'contrastive explanations' which, for feature set $\mathcal{F}$, are defined as the minimal subset $\mathcal{Y} \subseteq \mathcal{F}$ such that allowing $\mathcal{Y}$ to take an arbitrary value changes the prediction. It was proven in [25] that PI-explanations and contrastive explanations are minimal hitting sets of each other. A downside of this exact approach is that in scenarios where there are a lot of classes, such as in a regression setting, the PI-explanation $\mathcal{X}$ may be too large to be useful as very small changes in $\mathbf{x}$ may already result in another class. In similar fashion, the contrastive explanation $\mathcal{Y}$ may be too small to really give any insight into the decision-making process of the tree.

Counterfactual Explanations When the goal is to explain undesired predictions, counterfactual explanations can be used, which explains a prediction $f(\hat{\mathbf{x}})$ by giving an alternative, perturbed model input $\mathbf{x} = \hat{\mathbf{x}} + \epsilon$ such that $f(\mathbf{x})$ has a more desired outcome [65]. In recent years, work has been done to make use of the transparent structure of the decision tree and ensembles of trees when creating counterfactual explanations. In [29] counterfactuals of decision trees are generated by first converting the tree into a characteristic formula and then using a satisfiability solver to find model input that satisfies that formula under the desired constraints. In [51] a methodology was then created which aims to increase the efficiency of generating counterfactual explanations by making the search complexity independent of the size of the feature space, resulting in much better scalability than previous methods.

Temporal Saliency Maps The idea of saliency maps on images[2] has also been applied to time series. Heatmaps can be created to show the relevance of features, either enriching a line plot of a time series [59, 60, 1, 50], or showing the feature importance of all features with temporal information [52, 20]. It is important to know, however, that it is challenging to interpret these visualizations, requiring domain or expert knowledge [59].

Layer-Wise Relevance Propagation Another possible explanation technique is Layer-wise Relevance Propagation (LRP) which propagates predictions backwards through a neural network, highlighting which neurons were relevant in activating neurons deeper into the network [47]. With respect to time series, RLP has been applied to Long-Short Term Memory networks (LSTMs) [47, 72] and Convolutional Neural Networks (CNNs) [7].

3.2.3. XAI in Industrial Settings

Relatively little research exists which covers the use of XAI in industrial scenarios. In [63] XAI is applied in a manufacturing process setting by using SHAP to extract from ML models explanations to which machines attribute to better wafer quality. From these feature attributions an improvement plan was created for resource optimization. In Section 3.4 two more examples of XAI used in industrial settings will be given.

3.3. Domain Knowledge Integration

Many domain knowledge integration methods exist, but they are not always suited for our specific use case. Some are physics based, which is not possible due to an assumed lack of physics knowledge, while others have never been applied to production processes.

[2]https://analyticsindiamag.com/what-are-saliency-maps-in-deep-learning/

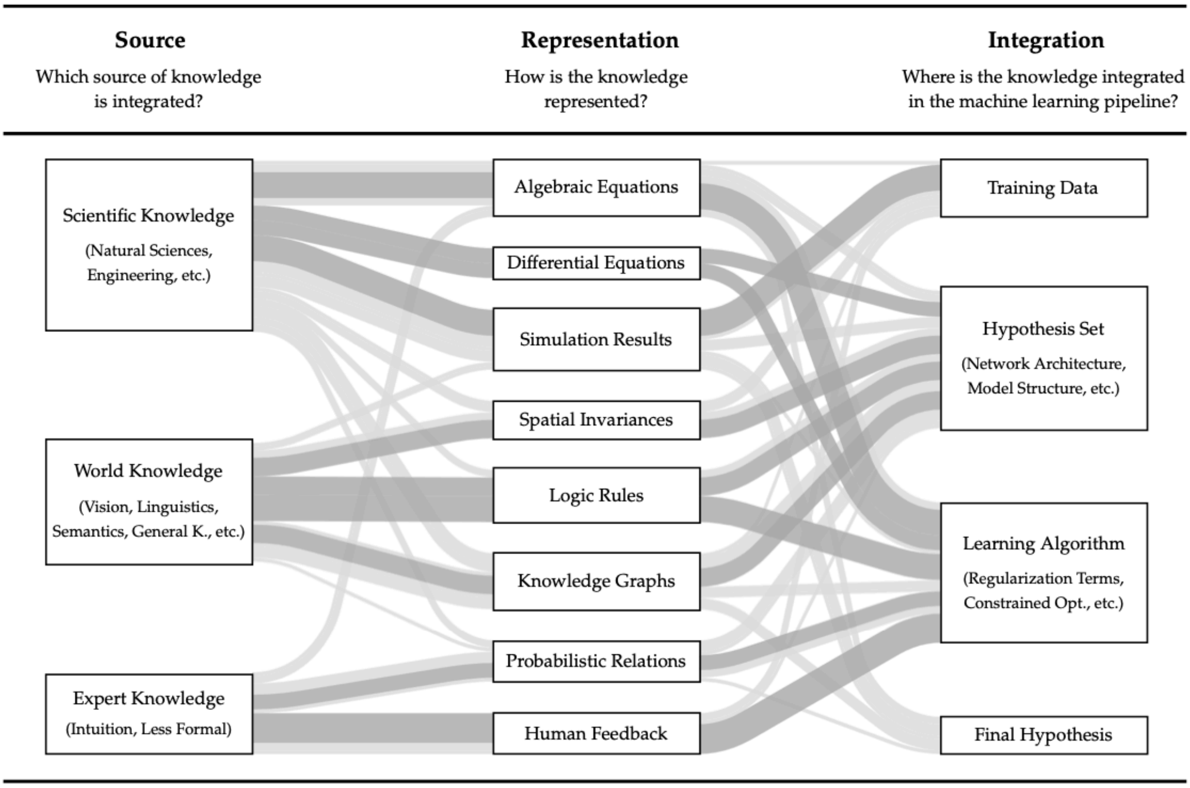


Figure 3.1: Taxonomy of Informed Machine Learning by Rueden et al. [58]. The width of the links represents the amount of papers using the linked approaches. Highlighted paths (darker grey) represent central approaches of IML.

Informed Machine Learning Using purely data-driven approaches when applying machine learning can lead to problems in scenarios where the available data is not enough to create sufficiently generalized models, or scenarios where models need to adhere to constraints dictated by (natural) laws and guidelines [58]. The act of integrating prior knowledge into the training process of a ML model to address these issues is called Informed Machine Learning (IML). In [58] IML approaches are classified by determining the source, representation, and integration of knowledge. This led to a 'Taxonomy of Informed Machine Learning', which can be seen in Figure 3.1, that summarizes the findings. A popular method of teaching prior knowledge to a model is by augmenting the loss function with terms that penalize a model when it produces output which is not in line with that prior knowledge [58, 56, 8].

Physics-Informed Machine Learning A sub-field of IML is Physics-informed Machine Learning which busies itself with modelling and forecasting of the dynamics of multiphysics and multiscale systems [30]. Most ML approaches currently are unable to extract interpretable information and knowledge from the vast amounts of data present in scientific domains, or are physically inconsistent. As prior knowledge (e.g. observational, empirical, physical, or mathematical) about the problem at hand is usually available in scientific settings, the ability to use this in models is needed [30]. One recent popular method to incorporate prior knowledge in the form of Nonlinear Partial Differential Equations is the Physics-Informed Neural Network [53] which uses regularization that penalizes a neural network if partial differential equations are not satisfied. In [43] a method aimed at learning physics-constrained LSTMs is proposed for the forecasting of photovoltaic power generation. The LSTM is extended with a data filtering module in front of the input layer, a clipping module before the output layer, and the loss function is extended with a loss penalty module that penalizes the model when physical bounds are not satisfied. The data filtering module is aimed at eliminating unreasonable forecasts (e.g. positive power generation at midnight), the clipping module is used to restrict the output of the model to eliminate physically unreasonable forecasts (e.g. negative power generation), and the modified loss function is used to penalize the model when upper and lower bounds on the expected output are exceeded [43].

Monotonic Constraints Another approach not covered in [58] is proposed by [78]. The goal is to implement prior knowledge in the form of monotonistic constraints by using lattices [21] in a model. A lattice is an interpolated look-up table which can be constrained to be monotonic and can be trained in standard empirical risk minimization frameworks to create deep lattice models [78]. Monotonicity can also be implemented through semi-infinite optimization as shown in the approach in [32] where expert knowledge is used to improve a model for a press hardening process with limited data. However, the given approach is limited to regression problems.

Improving Counterfactual Explanations When generating counterfactual explanations of tree-based models, one wants to guarantee the feasibility and plausibility of the resulting counterfactuals [51]. When generating counterfactual explanations of a model through optimization or satisfiability (as discussed in Section 3.2.2), domain knowledge constraints can be added which enforce known facts to hold. In [51] a methodology is proposed which allows the constraining of fixed and monotonic features, known linear relations, known logical implications, and resource constraints. They enforce plausibility through the assumption that a counterfactual explanation should be typical of 90% of the cases of the target class. Even though the domain knowledge integration constraints are promising, the focus of this paper lies more on the performance gains between the existing and proposed method, and less on the now available use cases.

3.4. Related Work

Having discussed the work in the different areas this research builds on, we will now review the limited amount of work that resembles the goals set out for this project, i.e. methodologies aimed at providing domain-knowledge driven explanations of production processes.

Domain-Knowledge-Based Feature Construction in Ironmakinging In [37] a methodology is proposed which uses genetic algorithm-based feature construction to predict the quality of an industrial production process. The genetic algorithm tries to find a function, using a pre-determined set of operators, which best maps the process variables to the quality of the final product. Domain knowledge in the form of additional operators can then be used to guide the genetic algorithm into finding more accurate solutions. With respect to the ironmaking process the method was applied to, an example of domain knowledge as given by the paper was that the 'shift' operator was added as the authors expected the effect of adding certain minerals to the process would show a delayed effect on the target variable. The explainability of this methodology follows from the fact that after training the most common components (relations between variables) observed in the different models will explain what influences the process quality. The methodology seems very promising, but the domain knowledge integration is only briefly discussed and limited to the adding of additional operators. It may be very useful to add the integration of domain knowledge specific to certain variables in the target problem, as this may reduce the search space or increase the plausibility.

Remaining Useful Life Estimation using XAI and Domain Knowledge In [64] a methodology is proposed for the explainable and domain knowledge driven prediction of the Remaining Useful Life (RUL) of a fatigue test bed. Domain knowledge, along with multiple feature selection techniques, was used to determine important features on which a random forest was trained. ELI5 and LIME were then used for global and local explanations of the RUL predictions. A clustering method was finally used to divide the data into different groups, with the goal of investigating the difference in feature relevance for the different groups. The methodology seems to accurately predict the RUL, but the feature selection techniques do not seem to improve the performance of the already most accurate tree-based models. This is to be expected as tree-based models have inherent feature selection (as will be discussed in Chapter 4). This means that the domain knowledge used in this paper does not really account for the performance of the methodology. Using ELI5 and LIME for explanations also seems to explain the important features of the model, but it the paper does not cover how this information can be used to, for instance, improve the RUL of the test bed.

3.5. Summary

In this literature review we divided the work supporting this research into three categories: production process modeling, explainable artificial intelligence, and domain knowledge integration. With respect to production process modeling we can conclude that many methods exist that allow for the accurate modelling of processes, and some even focus on the problem of unbalanced datasets, such as in the case of this research where we assume that there is a relatively low amount of labelled data samples due to varying sampling restrictions. However, these methods are not explainable and usually do not offer more than a pure estimation of the target variable. Recent explainable methods are able to show which features are important to the prediction of a target variable, but they have only been sparingly applied to production processes. In the few cases that cover production processes such as in this research the methodology usually comes down to applying out-of-the-box methods such as SHAP or LIME. Domain knowledge integration has been applied much more to the field of process modeling, but is usually implemented with physics based knowledge or the natural laws applying to the process, which are mostly unknown in the setting of this research. Other methods allow for the generation of counterfactual explanations with constraints, but these have not yet been applied to production processes and further research should prove whether these methods are able to generate plausible explanations in a production process setting.

4

Preliminaries

Before we discuss the contributions of this research in the next chapter, we need to cover the preliminaries. These are the methods on which this research builds which are taken directly from previous works, and they are divided into three categories: production processes, quality prediction, and model explanations. First, a formal definition of production processes is introduced. Then quality prediction is covered, together with the random forest model which is the model of choice in this research for predicting the quality of the production process. Lastly, we discuss tree-based explanations, such as counterfactual explanations for tree-based methods, and how we can generate those counterfactual explanations using optimization.

4.1. Production Process

Consider again the schematic in Figure 2.1. We formally define a production process to be a process producing some product whose quality y_t is defined as the quality of a sample produced at time step t. In the specific context of this project the quality indicator would be the amount of CO_2 in a filled carbonated drink bottle. Besides the quality of the product the process is assumed to be monitored by n_s sensors **S** where $s_{i,t}$ represents the value of sensor i at time step t. Lastly, we consider n_p process parameters **P** where $p_{i,t}$ represents the value of parameter i at time step t. These process parameters are assumed to be set manually to influence the quality of the product. We then denote the *process state* to be $\mathbf{x}_t = [\mathbf{p_{,t}}\ \mathbf{s_{,t}}]$ at time step t.

4.2. Quality Prediction

In Section 3 ML methods for quality prediction were discussed. As production processes can be very complex due to the number of sensors, there is a requirement for flexible models such as deep learning or ensemble methods. However, for this project it is very important that models are transparent and explainable and thus deep learning methods are not suited due to their opaque nature. Tree-based methods are a much better fit due to their interpretable structure which can be used to create model explanations, as we will cover in later sections, while retaining the flexibility of complex models.

4.2.1. Tree-based Models

Consider training data $\mathcal{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)$, targets $\mathbf{y} = (y_1, ..., y_n)$, and possible classes $\mathcal{C} = (c_1, ..., c_n)$. The decision tree modelling the training data can then be constructed using the CART algorithm which has the following pseudocode [40]:

1. Start at the root node with $\mathcal{X}_{node} = \mathcal{X}$.
2. Using exhaustive search, find the feature f and value v that splits X_{node} and minimizes the sum of node impurities for each resulting split. Create a child node for each data split, setting $\mathcal{X}_{node}$ to that split.
3. Repeat the previous step for each child node until the stopping criterion is met (e.g. minimum size of $\mathcal{X}_{node}$, maximum depth).

4. Use *minimum cost-complexity pruning* to reduce the resulting tree to the desired size [67]. This involves removing nodes from the tree such that the training set error of the original tree is least affected.

The decision tree is constructed by recursively determining the best way to split the data based on a *node impurity criterion*. The impurity of a node is minimal when it perfectly splits data, i.e. each leaf node contains samples of one class only. Two examples of this are the *Gini Impurity Index*,

$$g(p) = 1 - \sum_{j=1}^{J} p_j^2, \tag{4.1}$$

and the *entropy function*,

$$h(p) = -\sum_{j=1}^{J} p_j \log p_j. \tag{4.2}$$

However, as the latter is dependent on the computation of logarithms, which are computationally expensive, the former is usually chosen. When the optimal split is found, the process is repeated for each of the two generated splits until either a node has an impurity of 0 or the maximum depth is reached. The resulting leaf nodes are then assigned a value determined by a majority vote of the samples reaching that leaf. In the end a decision tree is obtained which is able to capture complex relationships within data and, if sufficiently deep, has a relatively low bias [23].

In this research regression trees are used, which only differ from classification trees in the sense that the targets **y** and classes C are nominal instead of ordinal [67]. We denote this regression tree as a function $f : \mathbb{R}^n \to \mathcal{R}$.

4.2.2. Random Forests

The central idea of *random forests*, which is a type of tree ensembles, is to average the predictions of a set of many noisy decision trees F such that the variance decreases while the bias stays the same [23]. A random forest prediction then comes down to

$$f(\mathbf{x}) = \frac{1}{|F|} \sum_{f_i \in F} f_i(\mathbf{x}). \tag{4.3}$$

Additionally, the individual trees are trained using a procedure called *bootstrapping* which trains each tree on a subset of the dataset $\mathcal{X}$, created by choosing n times from n points with replacement [3]. Aggregating predictors trained using bootstrapping is called *bagging* (bootstrap-aggregating) [3].

4.2.3. Random Forest for Quality Prediction

In order to then use the random forest to predict product quality y_t at time step t we use as input the sensor data $\mathbf{S}_{i,t}$ and process parameters $\mathbf{P}_{i,t}$. This then gives the training data $\mathcal{X} = [\mathbf{P}\ \mathbf{S}]$.

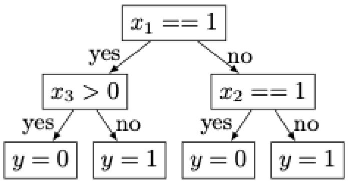
4.3. Model Explanations

Given that we are using decision tree-based models, there are multiple ways to generate explanations. We will first discuss explanations that follow from the interpretability of the model, and then we will discuss *counterfactual explanations* which locally explain a prediction by giving alternative model input which yields a more desirable model prediction.

4.3.1. Decision Tree Interpretability

As the functioning of a decision tree is transparent, we can give explanations which are directly related to the way the predictions are generated. An obvious local explanation given the structure of a decision tree is the *decision path* for a given input, which gives the nodes along which the input passed when the decision tree generated the result. If we take the example decision tree from Figure 4.1 and pass it the input $\mathbf{x} = (1, 1, 1)$, this would result in the following decision path:

$$(x_1 = 1) \to (x_3 > 0) \to y = 0. \tag{4.4}$$



(a) Graphical representation

```
if x1 == 1
  y = 0 if x3 > 0 else 1
else
  y = 0 if x2 == 1 else 1
return y
```

(b) Program (in Python)

$$\begin{aligned}\phi_f(\boldsymbol{x}, y) = &(x_1 = 1 \wedge x_3 > 0 \wedge y = 0)\\ &\vee (x_1 = 1 \wedge x_3 \leq 0 \wedge y = 1)\\ &\vee (x_1 = 0 \wedge x_2 = 1 \wedge y = 0)\\ &\vee (x_1 = 0 \wedge x_2 = 0 \wedge y = 1)\end{aligned}$$

(c) Characteristic formula

Figure 4.1: Decision tree represented graphically (left), in python (middle), and by its characteristic formula (right) [29].

A global formulation of this explanation would be the decision clauses which give for each leaf node of the decision tree the clauses that must hold for the input to reach said leaf node. Again, using the decision tree from Figure 4.1, an input will reach the left leaf node iff it satisfies the following condition:

$$x_1 = 1 \wedge x_3 > 0. \tag{4.5}$$

4.3.2. Counterfactual Explanations for Decision Trees

Another method of explaining model predictions is using a Counterfactual Explanation (CFE). Using the terminology used in [29], we can define the *set of counterfactual explanations* for an input $\hat{\mathbf{x}} \in \mathcal{X}$ and a predictive model $f : \mathbf{x} \to \mathbb{R}$ as:

$$\mathrm{CF}_f(\hat{\mathbf{x}}) = \{\mathbf{x} \in \mathcal{X} | f(\mathbf{x}) \neq f(\hat{\mathbf{x}})\}. \tag{4.6}$$

In other words, a CFE explains a model prediction by giving alternative model input that the model classifies with a different label. However, in the case of this project, we are more interested in explanations that show why the given process is producing products with quality y instead of the desired quality $\bar{y}$, by giving the most efficient transition from the current state of the production process $\hat{\mathbf{x}}$ to another state $\mathbf{x}$. This results in the following optimization problem [22]:

$$\min_{\mathbf{x} \in \mathcal{X}} C(\mathbf{x}, \hat{\mathbf{x}}) \quad \text{s.t.} \quad f(\mathbf{x}) = \bar{y}, \tag{4.7}$$

where $C(\mathbf{x}, \hat{\mathbf{x}})$ defines the cost function of moving from process state $\hat{\mathbf{x}}$ to $\mathbf{x}$. In [22], the ℓ_2 distance was used to define cost, but it should be argued that vector distance does not reflect difficulty of state changing in a production process.

In order to generate counterfactuals for decision tree predictions we can use the fact that the exact structure of the decision tree is known to construct the *characteristic formula* ϕ_f of the model [29]. The characteristic formula $\phi_f(\mathbf{x}, y)$ is satisfied (i.e. it returns $true$), iff $f(\mathbf{x}) = y$. An example of a decision tree and its characteristic formula is shown in Figure 4.1. Finding a counterfactual explanation for model input $\hat{\mathbf{x}}$ and desired label $\bar{y}$ that minimizes the cost function C then equates to finding the solution to the following minimization problem:

$$\min_{\mathbf{x} \in \mathcal{X}} C(\mathbf{x}, \hat{\mathbf{x}}) \quad \text{s.t.} \quad \phi_f(\mathbf{x}, \bar{y}). \tag{4.8}$$

4.3.3. Generating Counterfactual Explanations

Finding the solutions to the minimization problem from Equation 4.8 then requires optimization by, for example, Mixed Integer Programming (MIP) and related solvers such as Gurobi[1] [51]. However, when the model gets more complex, it may take too much time to always compute the optimal counterfactual explanations. Therefore, one may opt to sacrifice accuracy in favour of computational cost by rewriting the problem as a satisfiability problem as proposed in [29]:

$$\mathbf{x} \in \mathcal{X} \ . \ C(\mathbf{x}, \hat{\mathbf{x}}) < C(\mathbf{x}^*, \hat{\mathbf{x}}) + \delta \wedge \phi_f(\mathbf{x}, \bar{y}), \tag{4.9}$$

where $\mathbf{x}^*$ is the optimal counterfactual explanation (i.e. with the smallest distance to $\hat{\mathbf{x}}$) and δ is the maximum accepted distance between $\mathbf{x}$ and $\mathbf{x}^*$. Finding the solution to this equation then requires a

[1] https://www.gurobi.com/documentation/9.5/quickstart_mac/cs_grbpy_the_gurobi_python.html

satisfiability solver such as Satisfiability Modulo Theory (SMT) solvers, which generalize boolean satisfiability (SAT) by adding quality reasoning, arithmetic, arrays, and other useful first order logic [48]. In [29] a focus was laid on PySMT [13] as it is an off-the-shelf solver that allows easy deployment, but other libraries exist, such as Z3Py[2] and GekkoPy[3].

[2]https://ericpony.github.io/z3py-tutorial/guide-examples.htm
[3]https://github.com/mariushelf/gekkopy

5

Combining Quality Prediction, Explainability, and Domain Knowledge Integration

Having discussed the preliminaries in the previous chapter, we will now discuss the methodology proposed by this thesis that builds on top of that work, which consists of two parts. The first is a CFE generation method built on top of the random forest target prediction model which, given the (current) process state, gives an alternative process state which should be more desirable according to domain-knowledge based global and local constraints. The second part is a process state forecasting method used to forecast the impact of process parameter changes suggested by the CFEs generated using the first part. This process state forecast can then also be used as input to the target prediction model, circumventing the problem of a low number of samples preventing the use of complex time series prediction models. The goal of this two-part system is to enable process engineers to analyse a process by inspecting the suggested process state changes as proposed by the CFEs and analysing how variables react to those changes. A schematic for this methodology can be seen in Figure 5.1.

5.1. Counterfactual Explanations for Tree Ensembles

In Section 4.3.2 we discussed counterfactual explanations for decision trees. However, as we are using tree ensembles in which the predictions of multiple trees are averaged, we need to create a characteristic formula for tree ensembles. This can be done by considering the characteristic formula of a tree ensemble to be a conjunction of the characteristic formulas of the individual trees. As the prediction of the tree ensemble is usually defined as the mean of the predictions of the individual trees, this gives us the following formula:

$$\phi_f(\mathbf{x}, y) = \left[\bigwedge_{i \in 1 \ldots n} \phi_{f_i}(\mathbf{x}, y_i)\right] \wedge \left[\frac{1}{n} \sum_{i \in 1 \ldots n} y_i = y\right], \tag{5.1}$$

where n is the number of trees in the ensemble and y_i is defined as the prediction of tree i. This characteristic function is now satisfied iff $\mathbf{x}$ is predicted to have label y.

Ideally, optimal counterfactual explanations are then generated by finding the minimum described by equation 4.8. However, this may take too much time when the size of ensembles increases. If we do not care about small gains in the cost function when that means that we get a solution much quicker, we can find satisfactory solutions by setting a maximum acceptable d_{max} (or finding better solutions by iteratively lowering d_{max}):

$$\mathbf{x} \in \mathcal{X} \quad \text{s.t.} \tag{5.2}$$

$$C(\mathbf{x}, \hat{\mathbf{x}}) < d_{max} \tag{5.3}$$

$$\phi_f(\mathbf{x}, \bar{y}). \tag{5.4}$$

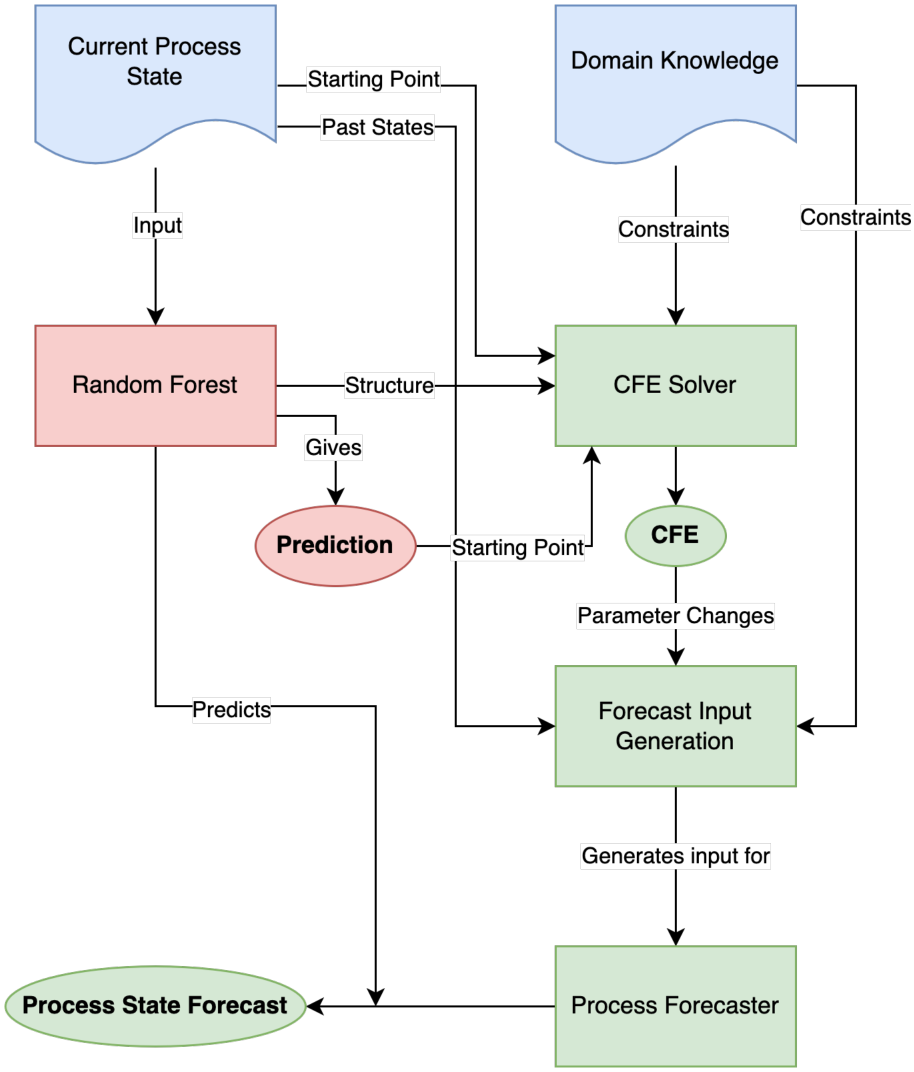


Figure 5.1: Structure of the proposed methodology.

Constraint	Predicate
Set variable i to a fixed set of values V	$\bigvee_j x_i = v_j$
Bound variable i to $[u, l]$	$x_i \geq l \wedge x_i \leq u$
Condition between two variables i and j	$x_i > x_j \rightarrow x_i > v$, $x_i < x_j \rightarrow x_i < v$, etc.
Link variable i and j to have the same value	$x_i = x_j$
Function relation between three variables i, j, and k	$x_i + x_j = x_k$, $x_i * x_j = x_k$, etc.
Aggregation Bound the maximum possible change of aggregation variable i that the variable j it aggregates allows, given aggregation window size d, allowed time t, and time series data **z** of variable j for the last d time steps. We assume x_j stays constant for t time steps.	$\sum_{i=t}^{d} z_i - tx_jx_i <= \sum_{i=t}^{d} z_i + tx_j$

Table 5.1: The relation between constraints on variables and how those constraints can be added to the counterfactual explanation solving process.

5.2. Improved Counterfactual Explanations

In [51] it is proposed that domain knowledge can be used in the form of constraints which are placed on the optimization or solving process. In this research their method is adapted to be more specific for production processes, with the aim of creating counterfactual explanations which are more plausible with respect to known facts about the target process.

5.2.1. Constraint Types

In order to express domain knowledge in the form of constraints we must first define what types of constraints we can apply to the generation of counterfactual explanations. These constraints must be defined such that they can be expressed as predicates so that they can be used with the SMT solver. Table 5.1 shows the constraints considered in this research from which the domain knowledge integration constraints can be built. The **Set** constraint allows us to fix variables to predetermined values (e.g. 0 and 1), the **Bound** constraint bounds a variable between an upper and lower bound, the **Condition** constraint allows for the encoding of causal relations between variables, the **Link** constraint fixes two variables to have the same value, and the **Function** constraint encodes known relations in the variables. The **Aggregation** constraint is defined specifically for aggregation variables constructed to give the model additional information about the past by summing the values of the aggregated variable over the past d time steps. This is important in situations where important variables show great amounts of fluctuation, creating scenarios where a value at one time step is too different from surrounding values, thereby limiting the ability of models to learn using that variable. When we know the minimum and maximum of the aggregated variable, we know the maximum amount the aggregation variable can change in a given window. This is important if we want to generate an alternative process state which should be reachable within reasonable time.

5.2.2. Use Cases

These building blocks can then be used in two main ways: incorporating known facts about the target process, and the investigation of specific scenarios. In Chapter 6 we will discuss what constraints we can extract in the context of the bottle filling machine.

Known Facts

When incorporating known facts about the target process we can formulate the constraints shown in Table 5.2. These constraints make sure that the generated counterfactual explanations do not violate logical relations due to the structure of the process. These constraints can also be used to incorporate certain expectations about the target process, but this should be done with caution as this restricts the ability of the model to show unexpected workings within the process.

Investigating Specific Scenarios

Another use case of constraints is that specific scenarios can be explored, which is best explained by an example. Consider a hypothetical bottle filling machine with quality indicator y and three process param-

Domain Knowledge	As Constraint
Limits on variables due to current scenarios	**Set** or **Bound**
Direct link between variable values due to process structure	**Link**
Relations between variables due to process structure	**Condition** or **Function**

Table 5.2: The relation between domain knowledge about the target process and what constraints are required to apply the knowledge to the generation of counterfactual explanations.

eters, a, b, and c. If y is currently 0.3, and we would like it to be 0.7 we could generate a counterfactual explanation such that:

$$\mathbf{x} \in \mathcal{X} \; . \; \phi_f(\mathbf{x}, 0.7), \tag{5.5}$$

which could, for instance, recommend setting (a, b, c) to $(0.1, 0.6, 0.7)$. But consider now that we cannot set parameter c higher than 0.5, perhaps due to a malfunction or an ingredient shortage. If we then restrict c to be less than 0.5 for this specific case we can force the model to generate counterfactual explanations which are more plausible in the current context.

5.3. The Cost of Changing Process State

In Section 4.3.2 a cost function $C(\hat{\mathbf{x}}, \mathbf{x})$ which defines the cost of moving from the original process state $\hat{\mathbf{x}}$ to the suggested new process state $\mathbf{x}$ (i.e. the CFE) was introduced. However, the ℓ_2 norm used in previous work is not a good indicator of that cost and therefore not meaningful to process engineers. With respect to the bottle filling problem, we can argue that stopping with filling can be done instantly, meaning that the associated cost should be zero, while increasing the temperature of a lemonade tank can take many minutes, equating to a high cost. If we therefore want to make this cost more reflective of the problem of moving between process states, the cost function should defined as $T(\mathbf{x}, \hat{\mathbf{x}}) \in \mathbb{R}$, being the amount of time steps it takes to get from process state $\hat{\mathbf{x}}$ to $\mathbf{x}$. In order to determine the amount of time it takes for a change in process parameters to take the desired effect on the process we could try and determine this manually by analysing existing data and then specify this as domain knowledge, but this may be tedious and inaccurate. Another option is to use the fact that we have high-resolution data of the process sensors and parameters to forecast the process state over the changes in parameters.

5.3.1. Forecasting Process State

We define the forecasting model to forecast the process state at the current time step t until time step $t + n_f$. As input, it will take all data n_w time steps before the current time step t, as well as the process parameters from time step t until $t+n_f$, as it is assumed they can be determined beforehand. The output will then be all remaining process variables from time step t until $t+n_f$. This will result in the following input matrix:

$$\mathbf{x}_{t-n_w,t+n_f} = \begin{pmatrix} p_{0,t-n_w} & p_{0,t-n_w+1} & \cdots & p_{0,t-1} & p_{0,t} & p_{0,t+1} & \cdots & p_{0,t+n_f-1} & p_{0,t+n_f} \\ p_{1,t-n_w} & \cdots & & & & & & & \\ \cdots & & \cdots & & & & & & \\ p_{|\mathbf{P}|-1,t-n_w} & & & \cdots & & & & & \\ p_{|\mathbf{P}|,t-n_w} & & & & & & & & \\ s_{0,t-n_w} & s_{0,t-n_w+1} & \cdots & s_{0,t-1} & s_{0,t} & \text{x} & \cdots & \text{x} & \text{x} \\ s_{1,t-n_w} & \cdots & & & & & & & \\ \cdots & & \cdots & & & & & & \\ s_{|\mathbf{S}|-1,t-n_w} & & & \cdots & & & & & \\ s_{|\mathbf{S}|,t-n_w} & & & & & & & & \end{pmatrix}, \tag{5.6}$$

where x defines the values that need to be forecast, $p_{i,j}$ defines the value of the i-th process parameter at time step j, and $s_{i,j}$ defines the value of the remaining process variable (sensor) i at time step j. The problem is now that we do not know yet how to obtain the process parameters for the forecast.

5.3.2. Generating Forecast Process Parameters

Consider again the current state $\hat{\mathbf{x}}_t = [\hat{\mathbf{p}}_{,t}\ \hat{\mathbf{s}}_{,t}]$ at time step t as discussed in Section 4.1. Obtaining the process parameters for the forecast input requires a path from the current process parameters $\hat{\mathbf{p}}_{,t}$ to the parameters $\mathbf{p}$ suggested by the CFE $\mathbf{x} = [\mathbf{p}\ \mathbf{s}]$. It may not be possible to change all the parameters instantly due to time-related variables, such as aggregation variables, being restricted in change per time step. Determining the input features can therefore be turned into the following optimization problem over new process parameters $\mathbf{Q} \in \mathbb{R}^{n_p \times n_f}$:

$$\min_{\mathbf{Q}} \sum_i \sum_{j=t}^{t+n_f} |q_{i,j} - p_i| \tag{5.7}$$

$$\text{s.t.}\quad q_{i,k} = p_i \quad \exists k \forall i \tag{5.8}$$

$$q_{i,0} = \hat{p}_{i,t} \quad \forall i \tag{5.9}$$

$$c \in C, \tag{5.10}$$

where C defines the same set of constraints applied to the generation of CFEs. This optimization over the difference between the old and new parameters ensures that the new parameters are as close as possible to those suggested by the CFE.

5.3.3. The Forecast Model

For the sake of simplicity, in this research a NN model predicting each of the values individually was used. Using more sophisticated (deep learning based) multivariate time series forecasting models, as in [4, 12, 62, 50, 39] is therefore directed to future work. The input to the NN model forecasting the process state at time t then becomes the concatenation of the rows of $\mathcal{X}_t$, with the output being the individual missing values marked x in Equation 5.6.

We can then calculate $f(\mathbf{x}_k)$ for all $k \in \{t, ..., t + n_f\}$ to obtain the target variable forecast using the process state forecast, and the amount of time steps it takes for $f(\mathbf{x}_k)$ to reach $\bar{y}$ to determine the cost of changing from $\hat{\mathbf{x}}$ to $\mathbf{x}$. Note that this has allowed us to indirectly forecast the target variable without explicitly training a time-series forecasting model on the sample data. This circumvents the problem of the limited number of samples that are assumed to be available.

5.4. Hypotheses and Expectations Before Applying the Methodology to the Data

Before we apply the methodology to the data discussed in Chapter 2, we should first discuss expectations and hypotheses. It is important to note that these hypotheses have been formulated independently from the thesis company.

5.4.1. Hypothesized Relations

1. First of all, we know that the amount of carbonation is regulated by a CO_2 setpoint, which directly influences the amount of CO_2 injected into the lemonade. A direct expected relation is then that the balance between the amount of incoming CO_2 and incoming lemonade will be a logically important hypothesized relation to the target model.
2. As discussed in Chapter 2 and in [66] the amount of bottle CO_2 is influenced by filling temperature. We can therefore hypothesize that the proposed methodology will therefore require temperatures to increase when wanting to lower bottle CO_2 and vice-versa.
3. We discussed the fact that the lemonade is cooled before the carbonation process, but not anymore thereafter. We can therefore hypothesize that if the filling stops the temperature of all sections after the cooling section of the filling process will start to increase, however slow, which can in turn influence the CO_2 levels in bottles in line with Hypothesis 2. The linear regression analysis performed in Section 2.3.3 has shown that the buffer tank level, controlled by filling speeds, negatively correlates with the temperatures in the bottle filling machine, further adding to this theory.
4. Room temperature is also expected to influence the filling process in other ways. First of all higher temperatures in different mechanical parts of the process may negatively affect the direct amount

of carbonation before the carbonated lemonade enters the carbonation tank. Secondly, it may be possible that higher room temperature increases the temperature of the empty bottles, increasing the temperature of the filled lemonade, but this could be negligible. Discussing the results with process experts before and after they are obtained may reveal these points to be true or not.

5. Another relation described in [66] is the one between bottle CO_2 and filling pressure, as a higher filling pressure equates to a higher bottle CO_2. We can therefore also hypothesize that pressure sensors will also have a positive contribution into determining the bottle CO_2.

5.4.2. Expectations with Respect to Performance of the Methodology

1. First of all, the random forest model will itself select the features which it has determined to be most important, as it will always select the features which split the data the best. This can create the possibility that if a leaf can be reached without splitting on a specific variable, the value for that variable may be arbitrary, i.e., its accepted value interval is very large. This is no problem when using the model for direct prediction only, but when counterfactual explanations are generated based on the intervals directly obtained from the splits it means that the value will remain the same as in the original process state, as counterfactual explanation generation optimizes over the vector distance.
2. However, the importance of a variable to the target prediction model may be different than to the process state forecasting model, meaning that the process forecasting model may not receive plausible process parameters if the target prediction model deems some parameters to not be important. This again highlights the importance of incorporating domain knowledge into the counterfactual explanation generation process, as this will also improve the forecasting accuracy of the process forecaster.

6
Results

The results of applying the methodology to the artificial and real data introduced in Chapter 2 will be discussed in detail in this chapter. We will first go over the methodology used for preparing the data so that it is ready for use with the different models. Secondly, we will discuss the domain knowledge based constraints obtained through logic and expert judgement. Then, as a verification of the explanation performance, the methodology is first applied to the artificial dataset to determine its effectiveness in a setting where the relations between the process variables is known. Ideally, this signalizes that the methodology is able to pick out the important process variables from the dataset and provides, through CFEs, alternative process parameters which allow for optimization in a real setting. Following a discussion of those results, the methodology is applied to the real-world dataset, the results will be discussed, and we will go over the hypotheses stated in the previous chapters to determine whether they can be confirmed. We close off with going over the key notes from a discussion of the results with a company expert.

6.1. Data Preparation

For the data to be used in a methodology it has to be cleaned and prepared first, after which additional feature processing steps can be taken. Finally, normalization may be done to prepare the data for methods which are sensitive to feature scaling.

Data Cleaning In order to clean the data, rows with missing values were removed from the dataset. Multiple sensors in the dataset have been non-operational for some periods and thus those related batches are not used.

Clipping of Features Some sensors reported values outside of their expected range, sometimes by a very large margin, for varying reasons. It is important that these values are clipped back as they can influence data normalization techniques.

De-trending of Features A combination of observation and discussion with experts revealed that a number of sensors showed linear trends where they are not expected. This was explained to be an issue with the calibration or rather a lack thereof. These trends have been removed via removal of the straight line best fit using linear regression provided by scikit-learn[1]. This is a relatively simplistic method but it suffices in this scenario where we are only looking to remove linear trends rather than complicated non-linear or recurring trends [73].

Converting Aggregation Features A number of features represented the sum of an underlying variable at time step t as follows:

$$V(t) = \int_{t_0}^{t} v(t)dt, \tag{6.1}$$

[1] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

and we can convert this back by differentiation:

$$v(t) = \frac{d}{dt}V(t). \tag{6.2}$$

But, due to the fact that the time steps are discrete, this comes down to

$$v(t) = V(t) - V(t-1). \tag{6.3}$$

Compensating for Sample Lab Analysis Delay As it takes some time for the analyses of bottle samples to register in the database, the samples have been shifted back in time to compensate. This delay was determined by the graduation company to be between three and five minutes.

Creating Aggregation Features As some features show high degrees of fluctuation between close time steps, additional features encoding the summation over a period $[t-n, t]$ have been created as follows:

$$V(t,n) = \sum_{x=t-n}^{t} v(x). \tag{6.4}$$

This allows the model to predict based on the number of bottles filled in the past n time steps, which may hold more information in cases where filling speeds are highly variant. Aggregation features have been created for the filling speed, as it shows great fluctuations, and incoming CO_2 flow.

Data Normalization An advantage of tree-based models is that they are invariant to feature scaling as scaling the features differently does not change the way the splits are determined. However, this is the case with NN-based methods [35] and other methods assuming 0 mean and unit variance. Therefore, before using the data with such methods it is scaled to unit variance (using scikit-learn[2]) as follows:

$$\mathcal{X}' = \frac{\mathcal{X} - \boldsymbol{\mu}}{\boldsymbol{\sigma}}, \tag{6.5}$$

where $\mathcal{X}$ represents the original data, $\mathcal{X}'$ the scaled data, $\boldsymbol{\mu}$ the vector of feature means, and $\boldsymbol{\sigma}$ the vector of feature standard deviations.

Train-test splitting When splitting the data into train and test sets, it is important to consider the fact that predicting the value of a sample based on the two samples that come before and after it in a batch is much easier than predicting that sample using two samples from another product batch. If train and test sets are created by randomly sampling from all samples the performance of a trained model on the test set may be skewed, as the model had knowledge about the batch the test samples came from. Therefore, the splitting of train and test data was done based on product batches instead, ensuring that the model has no knowledge of the product batches the samples in the test set originate from.

6.2. Determining Constraints

Before we can determine the effectiveness the proposed methodology, we need to put together all the domain knowledge we have on the bottle filling process so that we can express them as constraints. In this process we can differentiate between global constraints, which apply to all scenarios of the filling process, and situational constraints, constraints which only apply in specific cases we want to explore.

6.2.1. Global Constraints

Global constraints are the constraints which follow directly from the structure of the bottle filling process and thus always hold.

Bounding Variables For most of the sensors we know rough bounds in which we expect values to lie. These bounds can be applied as **Bound** constraints to the CFE and forecast input generation.

[2]https://scikit-learn.org/stable/index.html

Relation Between Carbonation Tank and Flow We know that when the carbonation tank is completely full, there is no flow possible into the tank. We can therefore limit the incoming flow to be very low when the level of the carbonation tank is too high using a **Condition** constraint.

Limiting Aggregation Variables Using Aggregation constraints we can limit the aggregation features defined in Section 6.1 using **AggregationBound** constraints.

6.2.2. Situational Constraints

Situational constraints can be used to explore specific scenarios in the bottle filling process, specifically when the set of explanations should be limited to be more plausible given knowledge about the current state of the bottle filling machine.

Filling Speed If there are arbitrary problems with the filling of bottles the maximum filling speed may be limited which can be encoded with a Bound constraint.

Flow and Filling Speed If we want to investigate the influence of different filling speeds or flow speeds throughout the process we can limit them using **Bound** constraints.

6.3. Models

The methodology consists of a target prediction model and a process state forecasting model. We will now discuss for each of the models how they have been constructed and how their hyperparameters were chosen.

6.3.1. Target Prediction Model

As discussed in Chapter 4, random forests were used to predict the quality of the production process. Specifically, scikit-learn's implementation of random forests[3] was used. Grid-based hyperparameter optimization was done to determine the optimal maximum depth and number of trees in the forest.

6.3.2. Process State Forecasting Model

As we also discussed in Chapter 4, for process state forecasting a neural network model was constructed. As its implementation the scikit-learn multi-layer perceptron regressor[4] was used. Determining the optimal hyperparameters for this neural network is much more time-consuming (60+ minutes per network instance) due to the large amount of data present to train on, making grid-based hyperparameter search not realistic. The exact methodology is different for the artificial and real datasets and will therefore be discussed in each related section.

6.4. Performance on Artificial Data

In order to determine the effectiveness of the methodology we can first analyse its performance on the artificial data introduced in Chapter 2. We will explore the accuracy of the model and explore the plausibility of the generated explanations.

Additional Variables Introduced

In order to give model more information about the filling done in the past 50 time steps, an aggregation variable `filling_50_sum` is introduced which sums the filling done over said period.

Constraints Used

In order to make the results more plausible a number of constraints were used. First the aforementioned aggregation variable `filling_50_sum` is bounded using an Aggregation Bound constraint which bounds the value of `filling_50_sum` in the generated CFEs so that it cannot change more than the value for `filling` in the CFE. Furthermore, if not otherwise stated, the minimum filling speed has been restricted to be greater than 0.5, as (logically) we require some filling to be done at least. For each scenario we

[3] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
[4] https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html

will view the generated CFE, the generated input for the process state forecaster, the forecast process state, and the CO_2 predictions for the each of the forecast time steps.

6.4.1. Hyperparameters

Before evaluating the explanations we will first cover how the hyperparameters for the target prediction and process state forecast models have been determined. For the target prediction model grid search yielded an optimal number of trees of 20 with a maximum depth of 8. The results can be seen in Figure 6.2. Predictions of this model on the test set, where it achieved a MAE of 0.047, can be seen in Figure 6.1. In order to determine the optimal number of layers and neurons per layer, results were extrapolated using the figures in Figure 6.3. Taking overfitting into account, a singular layer of 50 neurons was chosen.

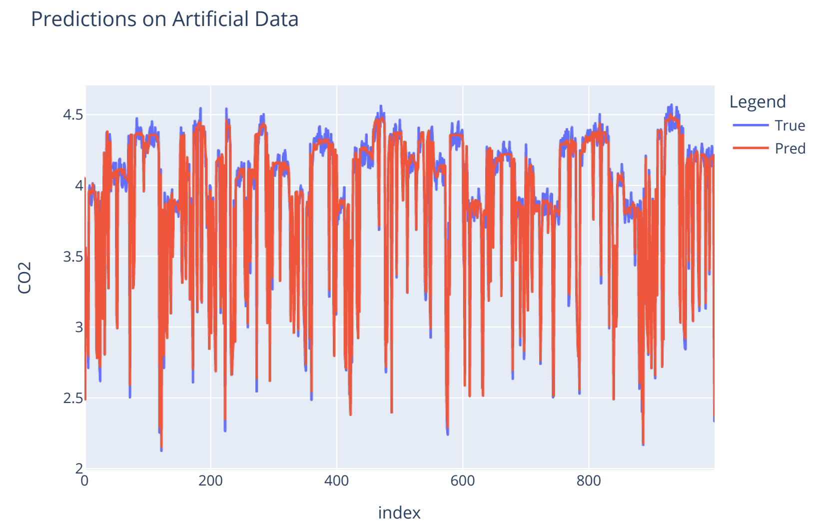


Figure 6.1: Predictions of the target prediction model on the test set of the artificial data, which achieved a MAE of 0.047.

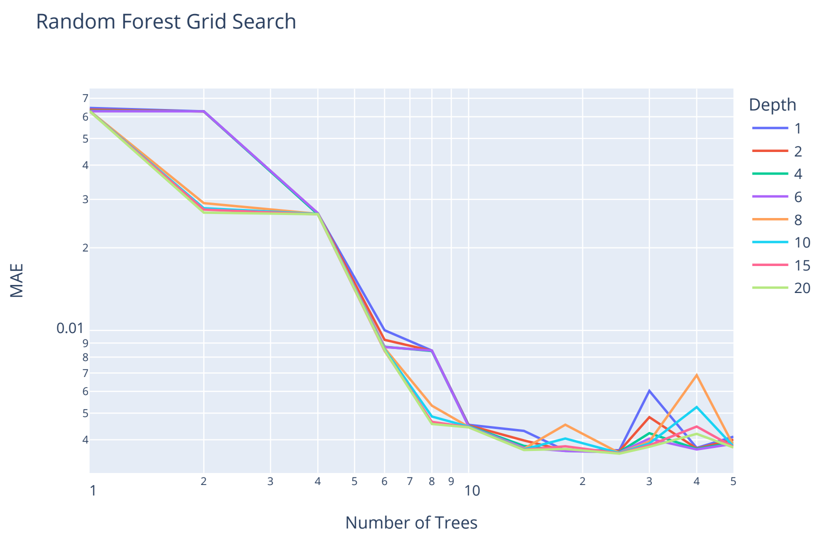
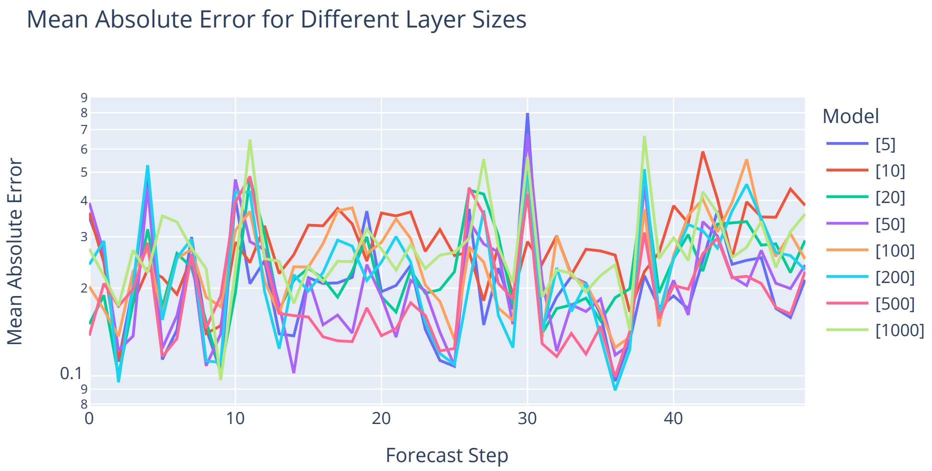


Figure 6.2: Results of using grid search to find the optimal hyperparameters for the random forest model on the artificial dataset.



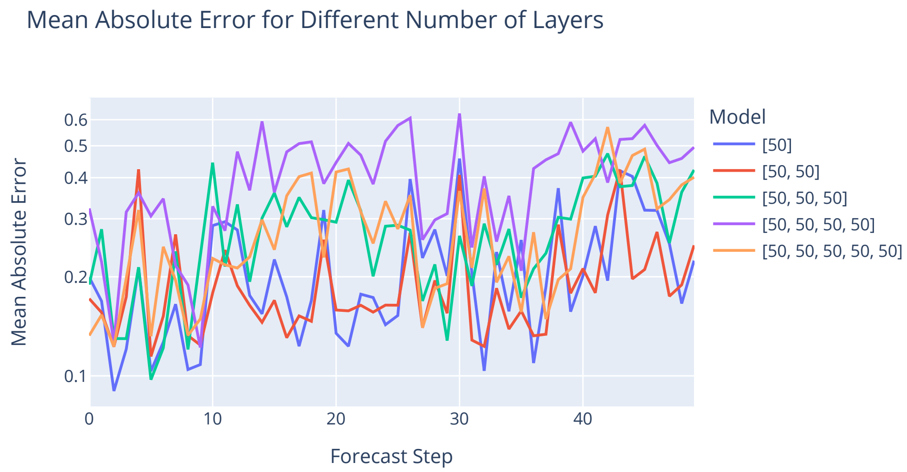


Figure 6.3: Test scores on the artificial dataset for different layer sizes (top) and number of layers (bottom).

6.4.2. Scenario 1: Lowering CO_2

In the first scenario the goal is to lower the bottle CO_2 from between 4.0 and 4.1 to between 2.9 and 3.1. The generated CFE shown in Table B.1 requires that the CO_2 setpoint should be lowered, the current filling speed greatly decreased, and the filler temperature should increase. In Figure 6.4 it can be seen that the suggested parameter changes (i.e. the changes to filling speed and the CO_2 setpoint) can be reached in 23 time steps and, according to the process state forecasting model, the changes cause temperatures to increase as desired. The forecast of the bottle CO_2 shows that there indeed is the desired decrease and it even goes lower than 3. From this first scenario we can conclude that the methodology is working as intended, highlighting that temperatures are a very important indicator of bottle CO_2. Important to note is that around the 545th time step the bottle CO_2 forecast increases again which seems to be correlated with an increase in filling speed and decrease in temperature. Future work could put more effort in making sure in the forecast parameter optimization that such jumps are minimized.

Figure 6.4: The generated CFE, forecast input, forecast process state, and corresponding CO_2 predictions for scenario 1. The teal and blue line indicate the start of the forecast and the point where the forecast input reaches the values suggested by the CFE respectively.

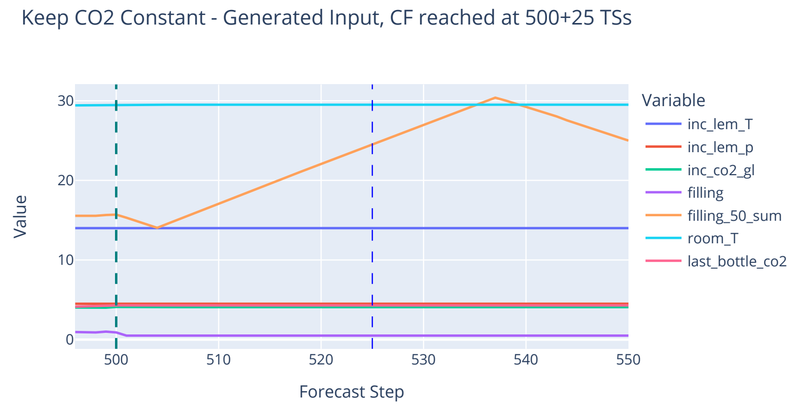
6.4.3. Scenario 2: Increasing CO_2

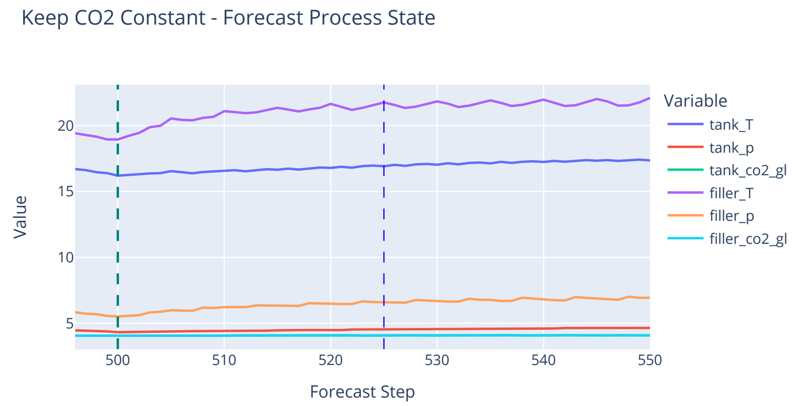
In the second scenario the goal is to increase the bottle CO_2 from a value lower than 3.5 to a value between 4.0 and 4.2. Table B.2 shows that the CFE requires that mainly the filler tank temperature should decrease greatly, while the current filling speed should increase. Figure 6.5 shows that the desired process parameters are reached right at the end of the forecast, but that the desired bottle CO_2 has already been reached around the 515th time step. We can see that the filler tank temperature is decreasing, as well as its pressure.

Figure 6.5: The generated CFE, forecast input, forecast process state, and corresponding CO_2 predictions for scenario 2. The teal and blue line indicate the start of the forecast and the point where the forecast input reaches the values suggested by the CFE respectively.

6.4.4. Scenario 3: Keeping CO_2 Constant

In the third scenario the goal is to keep the bottle CO_2 constant between 4.0 and 4.2. Table B.3 shows that the CFE requires a lower filling speed, higher filler aggregated sum, and slightly higher filler tank temperature. From Figure 6.6 it is clear that the bottle CO_2 is staying above 3.75 which is slightly lower than the desired bounds. The fact that it does not stay within the desired bounds may be attributed to the fact that the filling tank temperature is increasing too much, most likely due to the filling speed being too low.





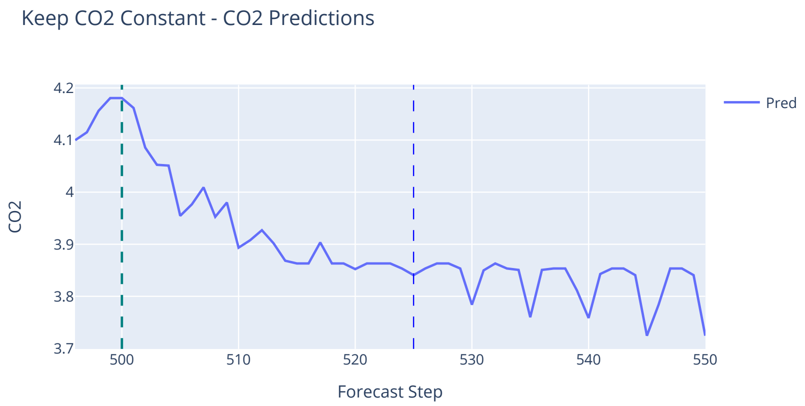


Figure 6.6: The generated CFE, forecast input, forecast process state, and corresponding CO_2 predictions for scenario 3. The teal and blue line indicate the start of the forecast and the point where the forecast input reaches the values suggested by the CFE respectively.

6.4.5. Scenario 4: Increasing CO_2 With Low Filling Speed

The fourth scenario tries to determine how the bottle CO_2 can be increased without increasing the filling speed above 0.25. The CFE shown in Table B.4 requires that the CO_2 setpoint and filler tank temperature need to decrease. Interestingly enough, the model does not require the filling speed to be much greater than zero even though it could be 0.25. The forecast seen in Figure 6.7 shows that the bottle CO_2 lowers to around 2.4 instead of increasing, indicating that it is probably not realistic to increase the temperature to the desired bounds without filling, which is to be expected when the filling tank temperature is so highly dependent on the filling speed.

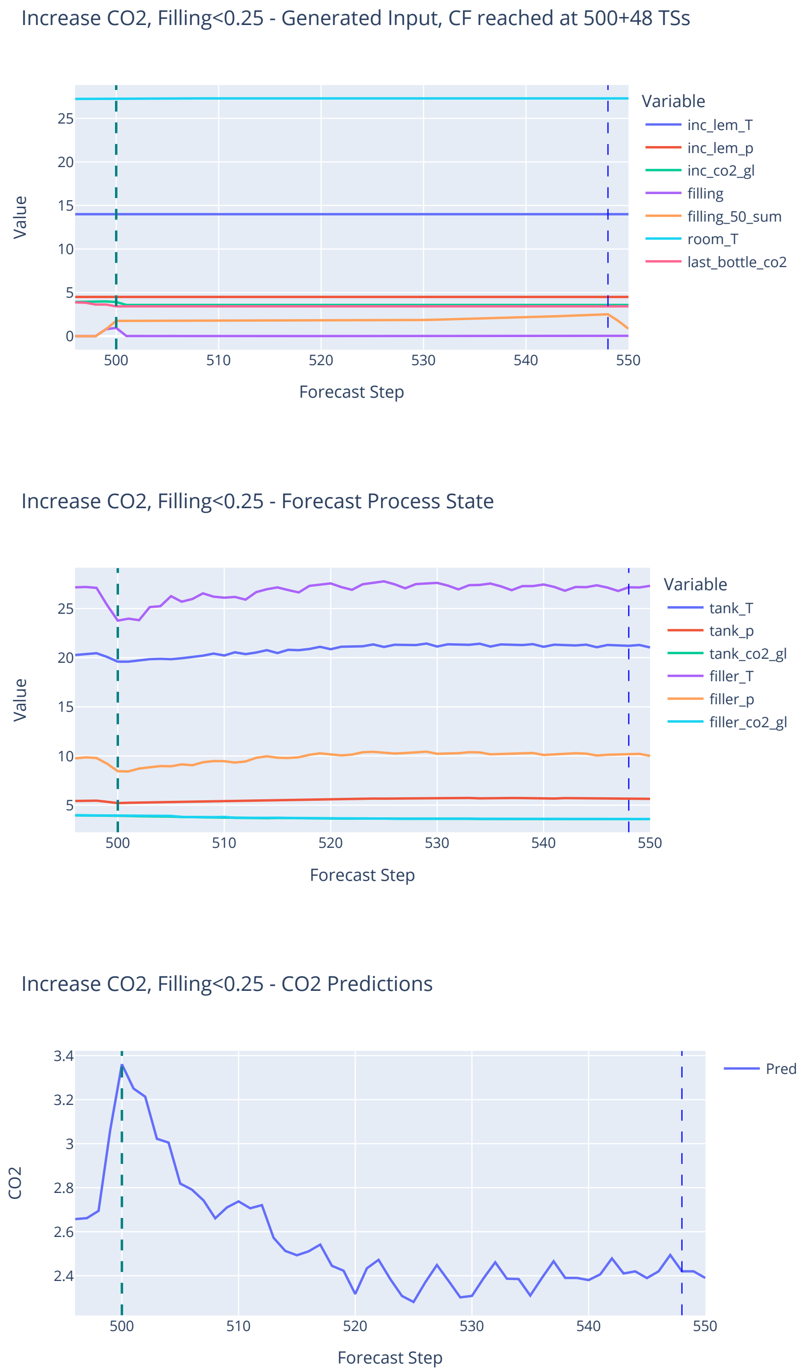


Figure 6.7: The generated CFE, forecast input, forecast process state, and corresponding CO_2 predictions for scenario 4. The teal and blue line indicate the start of the forecast and the point where the forecast input reaches the values suggested by the CFE respectively.

6.4.6. Scenario 5: Keeping CO_2 Constant While Reducing CO_2 Setpoint

The final scenario involves keeping the CO_2 between 4.0 and 4.2 while reducing the CO_2 setpoint to determine how the reduction of the setpoint can be compensated for. The CFE seen in Table B.5 requires that the filler tank temperature increases slightly and that the CO_2 setpoint is lowered to 3.57, which is lower than it is maximally allowed to be. Although it predicts that this would keep the bottle CO_2 constant, we can see in the forecast that this is not the case, as the bottle CO_2 drops off to around 3.8

while the rest of the process variables stay relatively constant. This indicates that the incoming CO_2 is probably set too low to maintain the required CO_2. This scenario also shows the importance of the process forecast, as the target prediction model by itself was not able to indicate the fact that the bottle CO_2 would decrease when applying the process parameters from the CFE.

Figure 6.8: The generated CFE, forecast input, forecast process state, and corresponding CO_2 predictions for scenario 5. The teal and blue line indicate the start of the forecast and the point where the forecast input reaches the values suggested by the CFE respectively.

6.4.7. Discussion

The different scenarios show that the methodology is able to effectively give CFEs to process states under varying restrictions, as well as forecast the influence of the changes to the process in terms of not only the target variable but also the process variables. The CFEs are more transparent than a direct forecast of the bottle CO_2 and due to the optimization in both the CFE generation and process parameter input generation only the necessary changes according to the target prediction model are provided.

But the methodology also shows limitations. The bottle CO_2 seems to mostly only react to the filler tank temperature. We can confirm this by viewing the feature importances[5] of the target prediction model in Table 6.1. It shows that the model really only uses filling speed and filler CO_2 for the prediction of the bottle CO_2. Therefore the generated CFEs do not require changes in the other variables as they do not influence the decision making of the model. On one hand this is a downside as the CFEs are less plausible (e.g. when suggesting higher filler tank temperature while not suggesting a higher tank temperature), but, on the other hand, this indicates that the solution to problems with bottle CO_2 need to be found in the filler tank temperature which is an important result on its own. Another minor issue with the forecast parameter generation is the fact that the optimization seems to have difficulties with keeping the filling aggregated sum to the value required by the CFE.

	Feature Importance
filler_T	1.658 ± 0.009
filler_co2_gl	0.122 ± 0.001
tank_co2_gl	0.03 ± 0.0
inc_co2_gl	0.002 ± 0.0
filler_p	0.0 ± 0.0
room_T	0.0 ± 0.0
tank_T	0.0 ± 0.0
filling_50_sum	0.0 ± 0.0
tank_p	0.0 ± 0.0
last_bottle_co2	0.0 ± 0.0
filling	0.0 ± 0.0
inc_lem_T	0.0 ± 0.0
inc_lem_p	0.0 ± 0.0

Table 6.1: Feature importances for the target prediction model trained on the artificial dataset.

6.5. Explanation Performance

Before evaluating the performance of the proposed methodology on the real-world dataset, the results of training both the target prediction model and process state forecasting model is discussed. Then, the evaluation of the explanations is done in two stages. First, we will explore generated explanations in similar scenarios as on the artificial data to determine whether the objective of the explanation has been fulfilled and whether they are plausible. In each scenario the original process state is hand picked according to the requirements specified by the scenario, where the original CO_2 value of the process state is taken to be the known sample value from the dataset. Then the results of an expert evaluation of the explanations will be discussed.

6.5.1. Training the Target Prediction Model

Hyperparameter tuning via grid search yielded an optimal number of trees of 40 with a maximum depth of 6. Predictions of this model on a test set can be seen in Figure 6.9. To demonstrate that this model does not merely output an approximation of the mean bottle CO_2 the distribution of train and test predictions can be seen in Figure 6.10, while a plot of the predictions can be seen in Figure 6.11. It is clear that the model has difficulties with accurately predicting strong deviations in normal bottle CO_2 labels. This could become a problem when generating CFEs for process states from which bottles are produced with

[5]Feature importances were generated using scikit-learn permutation importances: `https://scikit-learn.org/stable/modules/generated/sklearn.inspection.permutation_importance.html#sklearn.inspection.permutation_importance`

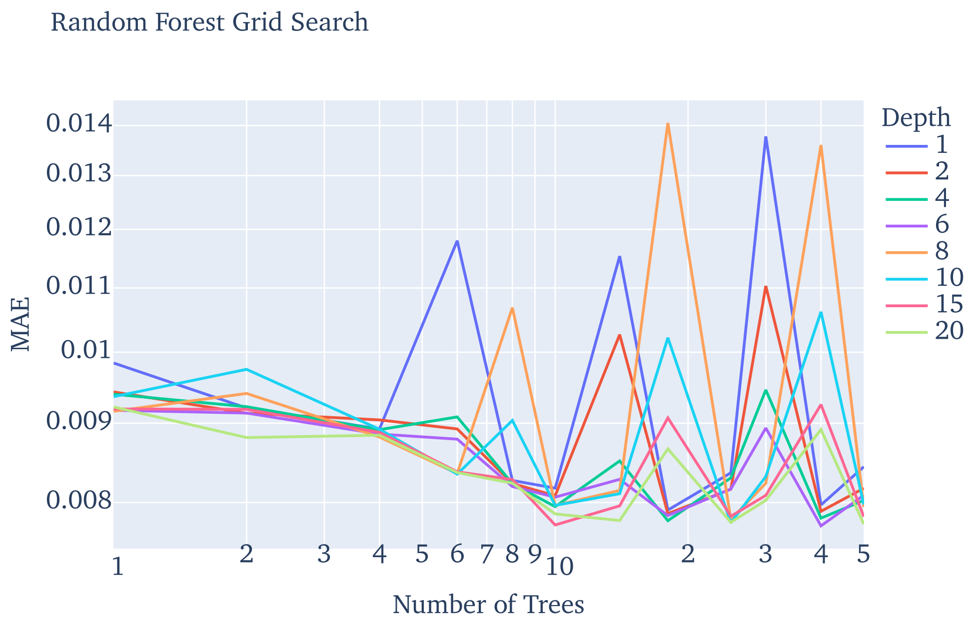


Figure 6.9: MAE for different combinations of random forest hyper parameters.

a significant difference from the average values, as the model may predict this process state to be normal and thus give no required changes in the CFE. This will be analysed in later sections.

Figure 6.11: Predictions on all test set samples using two random forests for comparison. The first with 40 trees of depth 6 obtained a MAE of 0.0040, while the second with 10 trees of depth 15 achieved a MAE of 0.0044.

Figure 6.10: Distributions for the entire dataset and predictions on the train and test set, demonstrating that the model does not merely output the general mean of the target values.

6.5.2. Training the Process State Forecasting Model

To recap: the process state forecasting model takes as input the previous n_w time steps of sensor data and process parameters, as well as the n_f generated process parameters. The output of the model then becomes the n_f forecast steps containing the forecast values for each sensor other than the process parameters. Similarly to the artificial dataset, $n_w = 10$ and $n_f = 50$ (500 seconds). Training data

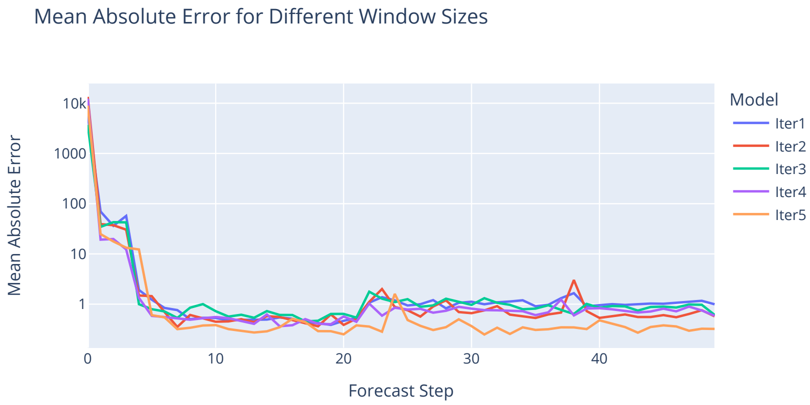
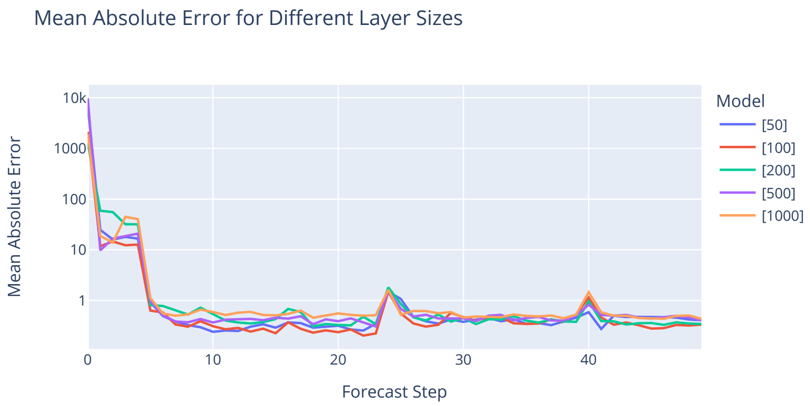


Figure 6.12: Different forecast window sizes.

was created by extracting sliding time series windows for each batch, ensuring that a window does not contain samples from two separate batches. In order to reduce complexity, the forecast was divided into smaller windows. The mean absolute error for each forecast step over different forecast sizes can be seen in Figure 6.12. It is not clear why in the error during the first forecast steps is very high, as the forecasts later on in this evaluation do not show abnormalities.



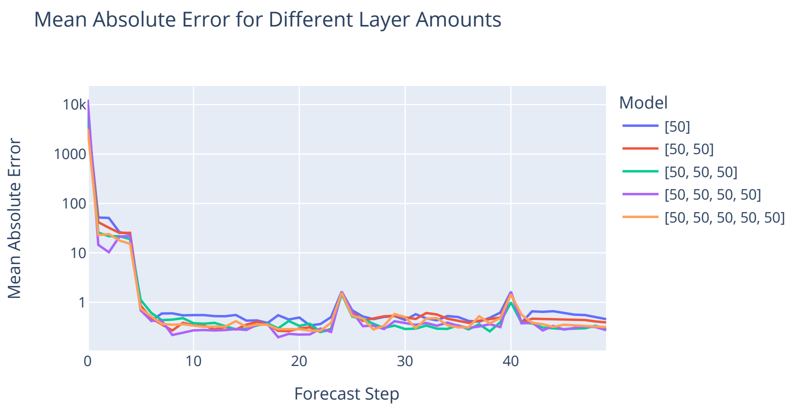


Figure 6.13: Test set losses for different layer sizes (top) and number of layers (bottom).

6.5.3. Scenarios

We will now discuss for each scenario what the goal is and whether the proposed methodology is able to correctly predict that the old process state does not produce bottles with a desired amount of CO_2, and whether it has achieved presenting a better process state. Whether the observations align with our expectations and hypotheses will then be discussed in the Discussion in Section 6.6.

Scenario 1: Lowering CO_2

In the first scenario the goal is to lower the CO_2 from above the specified upper limit (4.3) to between 3.95 and 4.05. The generated forecast input and CFE are shown in Figure 6.14, the process state forecast is shown in Figure 6.15, and the values for each variable in the CFE can be seen in Table B.6. The process parameters suggested by the CFE show that a decrease in flow into the cooler (`FT_031_130_031`), an increase in lemonade flow into the mixer (`FT_031_160_041`), a decrease in incoming CO_2 (`FT_031_140_037`), and a higher filling speed is required. Temperatures should increase in a number of sensors on pipes not used in production which could indicate that higher room temperature is needed, but this may also be due to a desire for a more stable production, as those pipes are used during start up or at the end of production which are more volatile stages. More importantly though an increase is required at the buffer tank (`TX_031_160_901`) and the temperature of incoming CO_2 (`TX_031_140_905`), indicating a required increase in lemonade and CO_2 temperature. The forecast shows that, as expected, the inline CO_2 measurement (`QT_031_140_083_PV`) decreases and that the CO_2 temperature increases. However, the other temperatures seem mostly to decrease, which could be the result of the increased filling speed and lemonade flow into the mixer. The CO_2 forecast decreases as expected, but increases again near the end. Possibly related to the fact that the buffer tank (`LT_031_160_901`) is empty at that point. This is unrealistic as the tanks may not run dry during production and could have been resolved with extra constraints defining the exact relations between the flows.

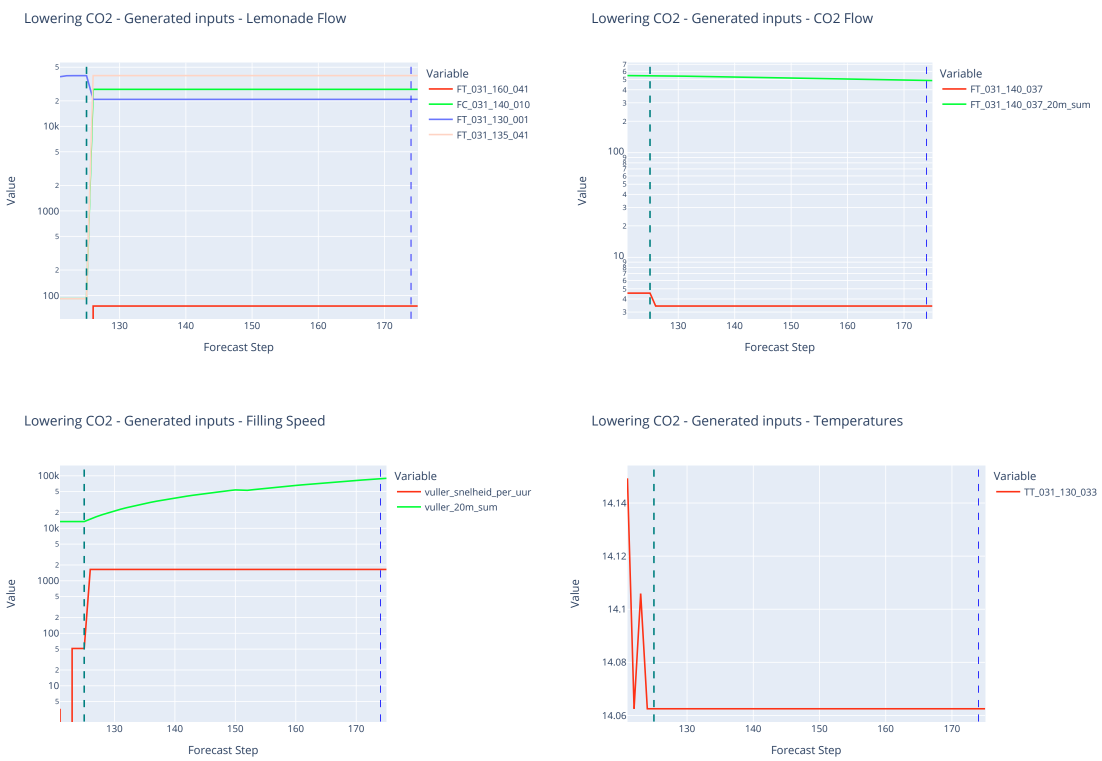


Figure 6.14: The generated CFE, forecast input, for Scenario 1. The teal and blue line indicate the start of the forecast and the point where the forecast input reaches the values suggested by the CFE respectively.

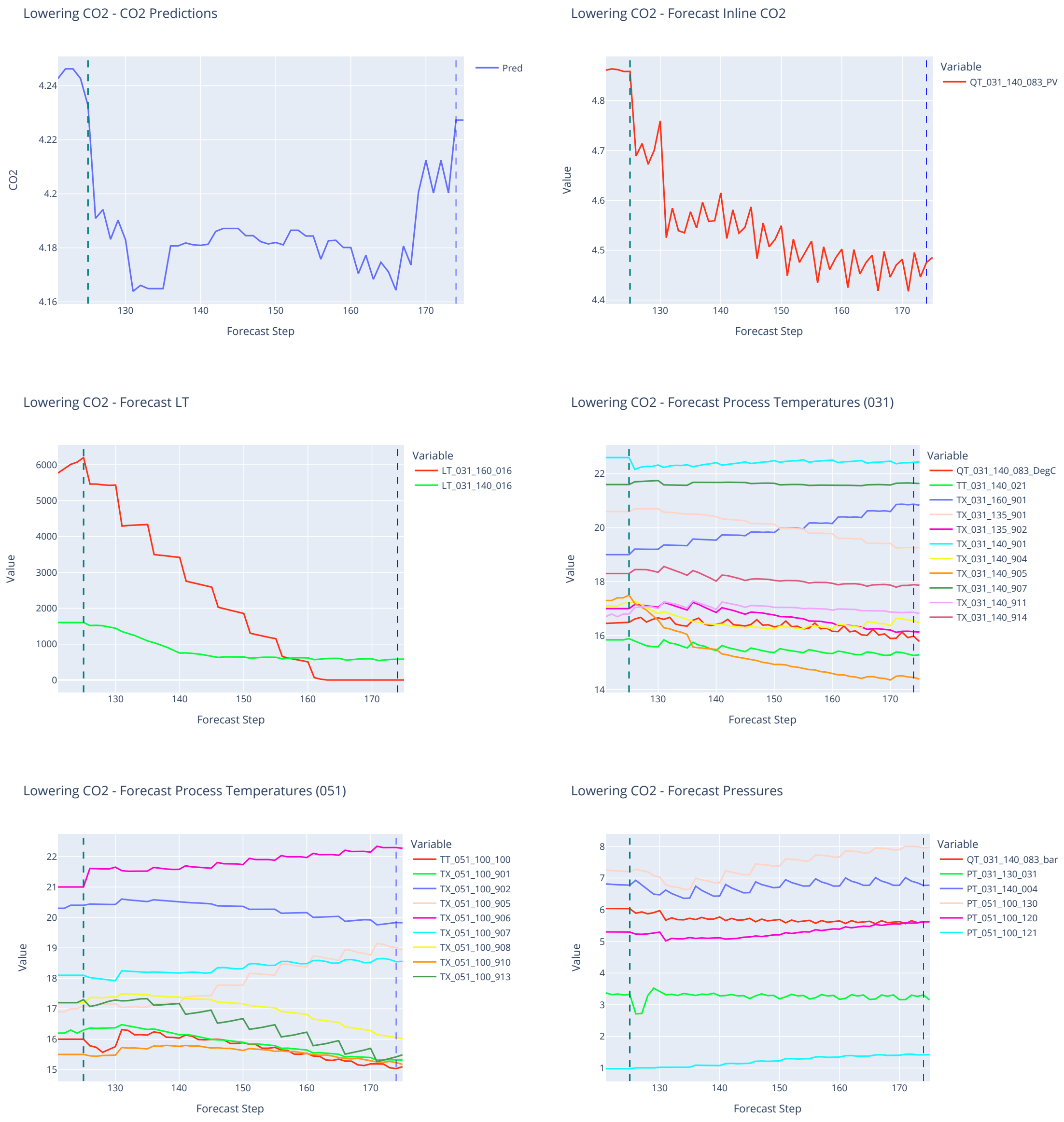


Figure 6.15: The forecast for all of the variables in Scenario 1. The teal and blue line indicate the start of the forecast and the point where the forecast input reaches the values suggested by the CFE respectively.

Scenario 2: Increasing CO_2
In the second scenario the goal was to increase CO_2 from below (<4.1) to within acceptable bounds (4.15-4.25). The generated forecast input and CFE are shown in Figure 6.16, the process state forecast is shown in Figure 6.17, and all the values for the CFE are shown in in Table B.7. The CFE suggested that the aggregated filling speed over the last 20 minutes and flow into the cooler and carbonation tank decrease, while incoming CO_2 flow and current filling speed increase, which we can expect to indeed increase bottle CO_2. Furthermore, an increase in temperatures is required in the filler tank and two temperature sensors on pipes related to bottle cleaning. The higher temperatures in cleaning pipes would suggest a lower filling speed, which could also explain the required higher filler tank temperature. The forecast shows an expected increase in the inline CO_2 measurement, as well as a decrease in buffer tank level which is expected when the flow into the cooler is lowered and the filling speed is increased. The forecast shows an increase in pressure as well which is in line with the idea that pressure increases bottle CO_2. While the CFE required an increase in filler tank temperature (`TT_051_100_100`), the temperature decreases in the forecast which is actually more in line with our expectations when the lemonade flow to the filler increases, and also more desired as a lower temperature should increase bottle CO_2. The pressures from the carbonation tank to the filler also increase, possibly also due to the increase in flow, which should also contribute to higher bottle CO_2.

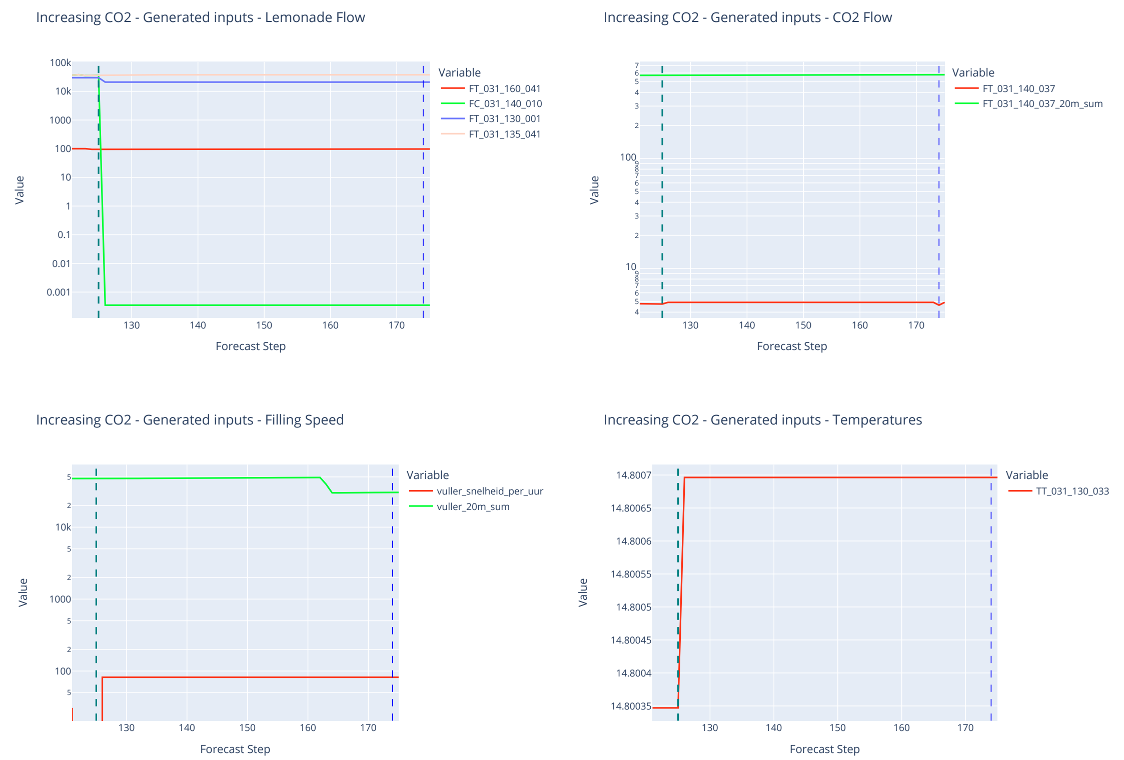


Figure 6.16: The generated CFE, forecast input, for Scenario 2. The teal and blue line indicate the start of the forecast and the point where the forecast input reaches the values suggested by the CFE respectively.

Scenario 3: Constant CO_2
For the third scenario the goal was to keep the CO_2 levels within the acceptable bounds (4.15-4.25). The generated forecast input and CFE are shown in Figure 6.18, the process state forecast is shown in Figure 6.19, and all the values for the CFE are shown in in Table B.8. The CFE suggests decreasing the flow into the buffer tank and cooler, while increasing the lemonade flow into the mixer and CO_2 while also increasing the amount of flow to the filler. It also required an increase in a number of temperature sensors, including the filling tank temperature. The resulting forecast shows an increase in inline CO_2 and pressures, with most temperatures dropping as well. We would associate that with an increase in CO_2 rather than keeping it consistent and that is what we are seeing in the CO_2 prediction as well. This

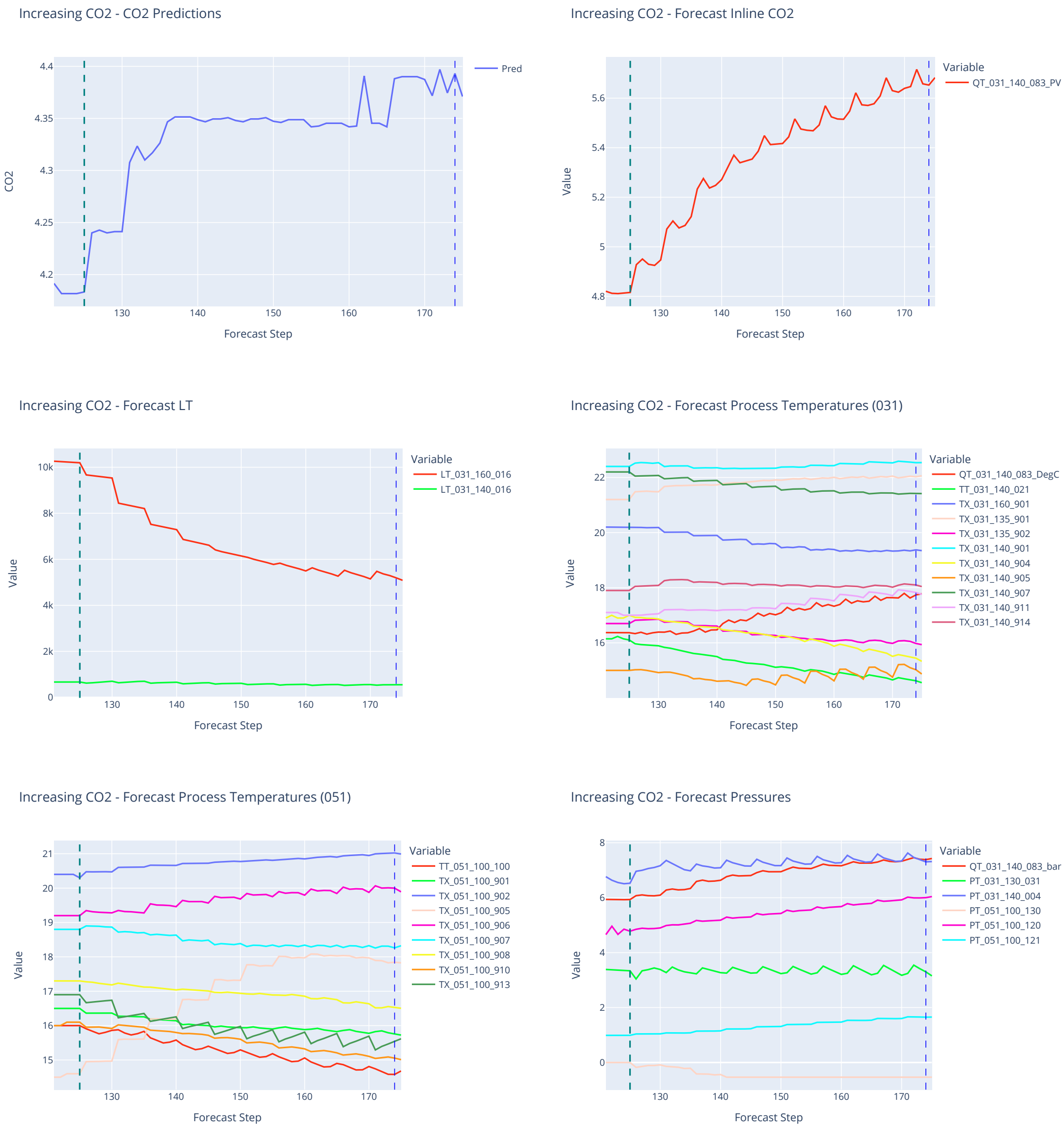


Figure 6.17: The forecast for all of the variables in Scenario 2. The teal and blue line indicate the start of the forecast and the point where the forecast input reaches the values suggested by the CFE respectively.

could be due to the fact that, in theory, no changes were needed as the process was already producing bottles with the correct amount of CO_2, while the CFE generation still generated a new process state which was different. This is probably due to the fact that, as discussed in Section 5.1, we are not in fact running an optimization to find the CFE, but iteratively trying to find a CFE with a smaller distance to the original and stopping at a certain threshold. In this specific case, the process parameters did not make it so that the required temperature changes were achieved to let the CO_2 stay constant, but rather decreased the temperatures which increased the CO_2 greatly.

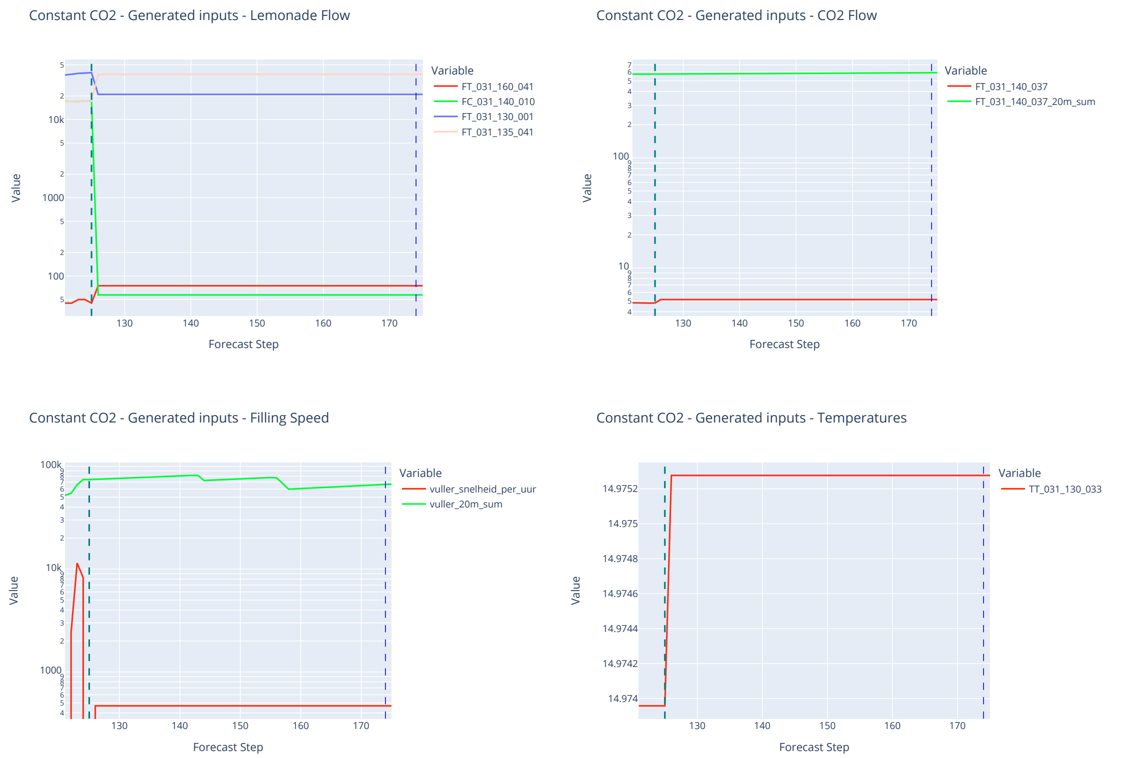


Figure 6.18: The generated CFE, forecast input, for Scenario 3. The teal and blue line indicate the start of the forecast and the point where the forecast input reaches the values suggested by the CFE respectively.

Scenario 4: Increasing CO_2 Without Increasing Filling

In the fourth scenario the goal is to increase the CO_2 from below bounds (<4.1) to within bounds (4.15-4.25) without increasing the filling speed, as we stated in Hypothesis 3 that low filling speed may cause a lower CO_2 due to warming of the lemonade. This scenario may therefore highlight what can be done to the process to counteract the many stops in production. The generated forecast input and CFE are shown in Figure 6.20, the process state forecast is shown in Figure 6.21, and all the values for the CFE are shown in in Table B.9. The CFE suggests decreasing lemonade flow into the mixer, carbonation tank, cooler, and filler, and decreasing incoming CO_2 flow to almost zero. Furthermore, an increase is required in inline pressure and temperature. The forecast shows an eventual increase in inline CO_2, which may not be realistic given the fact that the incoming CO_2 flow is set to almost zero. In the carbonation section most temperatures increase except for the temperature of the buffer tank and carbonation tank. In the filling section temperatures increase as well, except one pipe for cleaning and one for keeping filler tank pressure constant. These temperature changes seem so slowly increase bottle CO_2 while the sharp increase near the end seems to come from the large increase in both inline CO_2 and pressure.

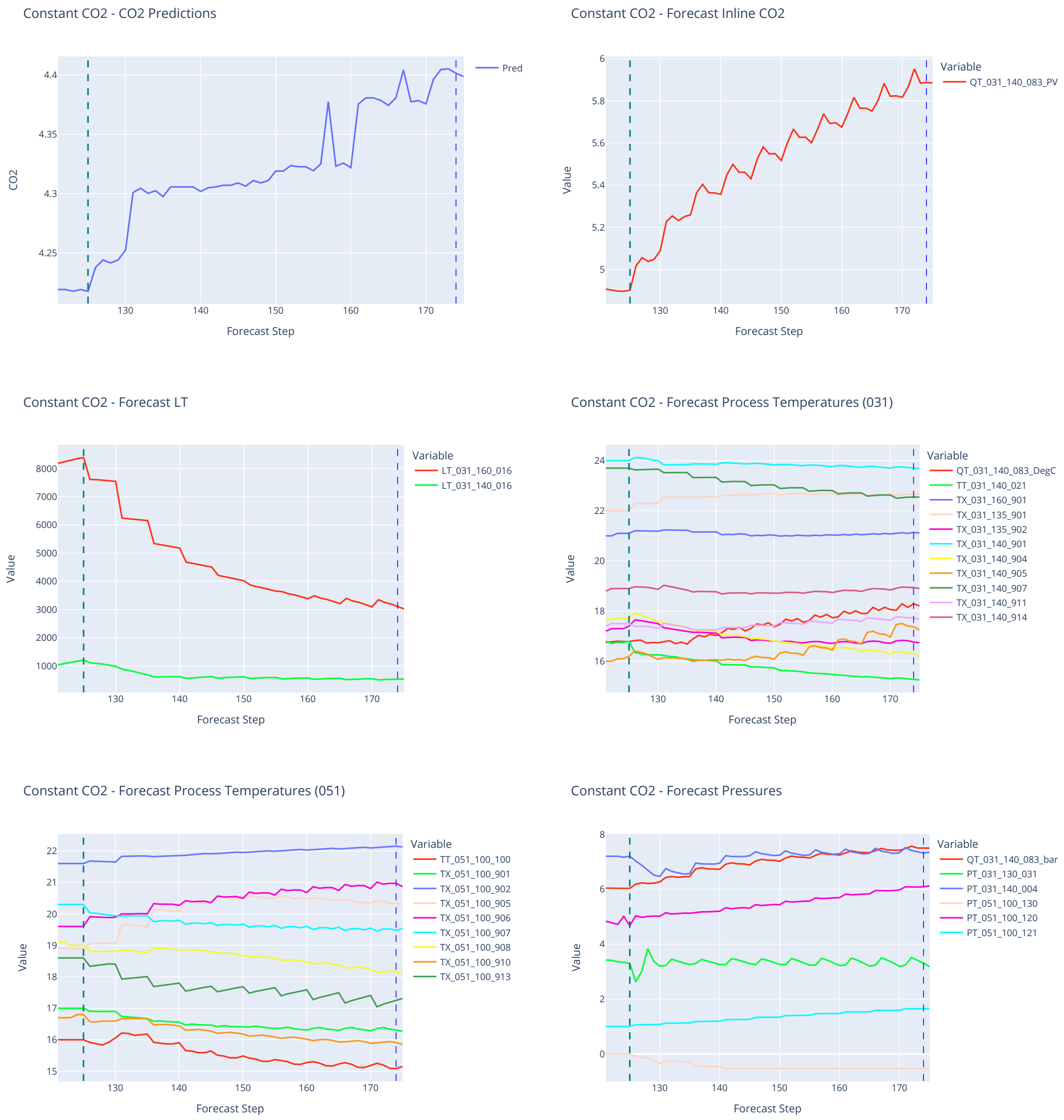


Figure 6.19: The forecast for all of the variables in Scenario 3. The teal and blue line indicate the start of the forecast and the point where the forecast input reaches the values suggested by the CFE respectively.

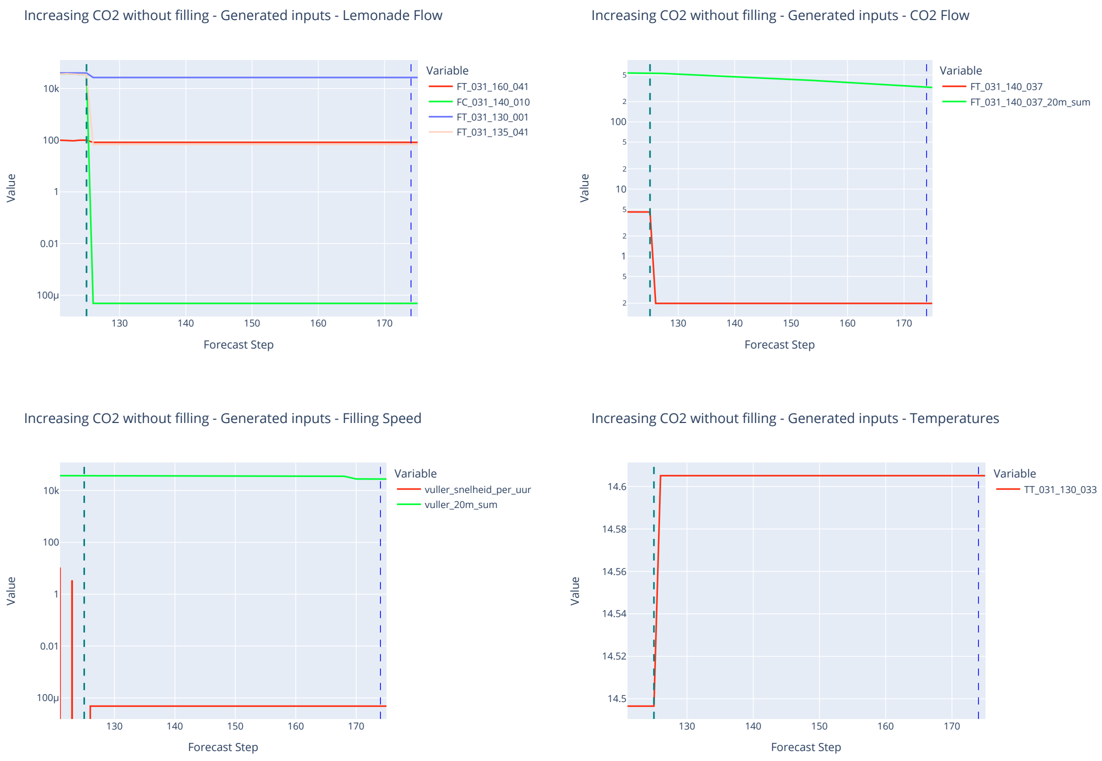


Figure 6.20: The generated CFE, forecast input, for Scenario 4. The teal and blue line indicate the start of the forecast and the point where the forecast input reaches the values suggested by the CFE respectively.

Scenario 5: Decreasing CO_2 Setpoint

The fifth scenario involves keeping the CO_2 within bounds (4.15-4.25) while lowering the CO_2 setpoint. We hypothesized that there are more factors to the prediction of CO_2 than only the CO_2 setpoint, and the suggestions made by the CFE may highlight which improvements may be made to improve bottle CO_2 without increasing the setpoint. The generated forecast input and CFE are shown in Figure 6.22, the process state forecast is shown in Figure 6.23, and all the values for the CFE are shown in in Table B.10. The CFE requires, in conjunction with the lower incoming CO_2, an increase in lemonade flow into the cooler, buffer tank, and mixer, while decreasing flow to the filler. The forecast then shows an expected decrease in inline CO_2 due to the decrease in CO_2 setpoint. The temperatures in the carbonation section most temperatures stay constant or show a slight decrease and then an increase, while in the filler section all temperatures slightly decrease, except for the filler tank temperature which increases slightly. It is likely that the CO_2 increase at the end of the forecast is caused by the decrease in temperatures, but it is probably not realistic that the CO_2 is able to stay high while the inline CO_2 drops to below 3.

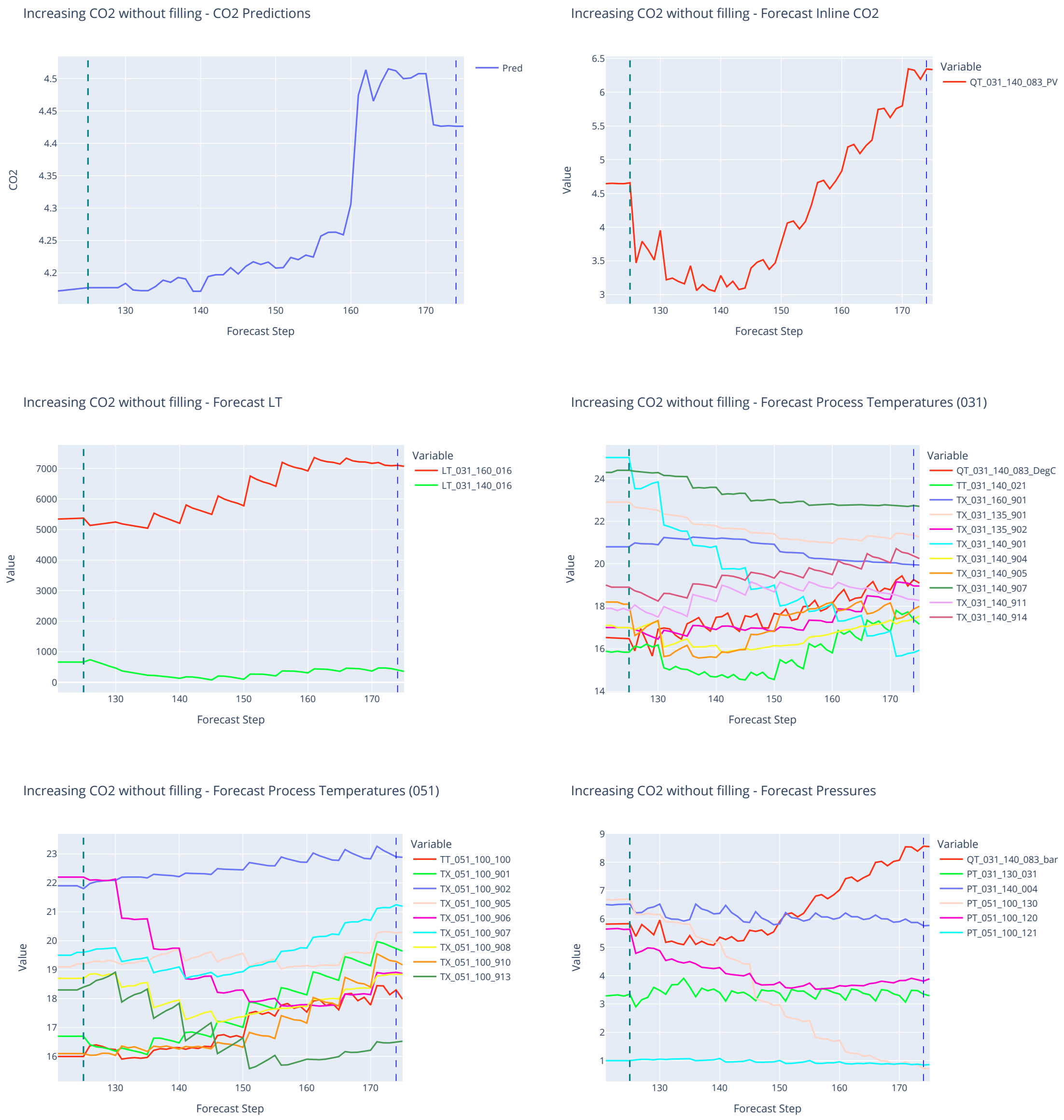


Figure 6.21: The forecast for all of the variables in Scenario 4. The teal and blue line indicate the start of the forecast and the point where the forecast input reaches the values suggested by the CFE respectively.

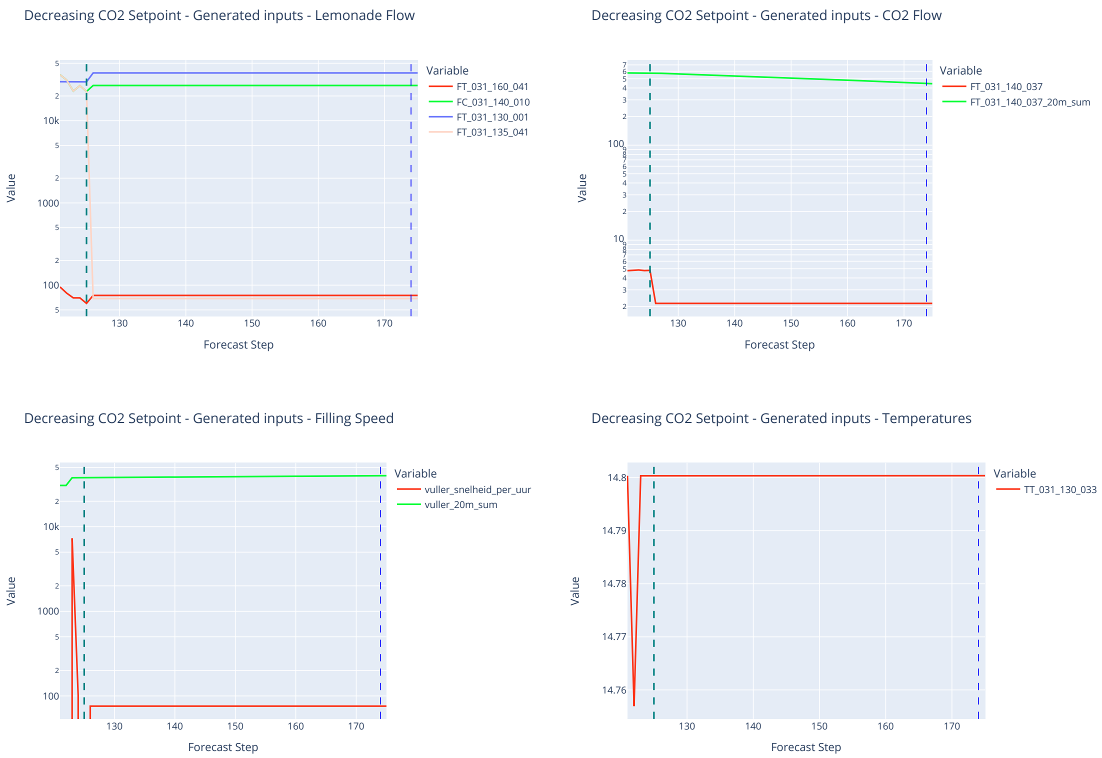


Figure 6.22: The generated CFE, forecast input, for Scenario 5. The teal and blue line indicate the start of the forecast and the point where the forecast input reaches the values suggested by the CFE respectively.

Scenario 6: Lowering CO_2 below specified lower limit.
For the last scenario we try to lower the CO_2 to below the specified bounds to a value lower than 4, as purposely generating CFEs to obtain 'bad' process states may reveal what factors worsen bottle CO_2. The generated forecast input and CFE are shown in Figure 6.24, the process state forecast is shown in Figure 6.25, and all the values for the CFE are shown in in Table B.11. The CFE suggests increasing lemonade flow into the cooler, mixer, and carbonation tank, while reducing filling speed and incoming CO_2. Furthermore, temperature increases are required in many sensors, such as the temperature from the cooler (`TT_031_130_033`) and the temperature going into the carbonation tank (`TT_031_140_021`) as expected when trying to lower bottle CO_2. The inline CO_2 measurement decreases and the inline pressure and temperature measurements increase as they should. The temperature of the buffer tank increases as well, likely due to the reduced flow of lemonade into the cooler and an increase in temperature from the cooler. However, the other temperatures and pressures are forecast to decrease and increase respectively, likely due to the increase in flow of lemonade in the carbonation section making it so that the CO_2 forecast is increasing rather than decreasing. This means that there is still a disconnect between the requirements set by the CFE for the sensor changes and the forecast sensor changes when using the new process parameters given by the CFE.

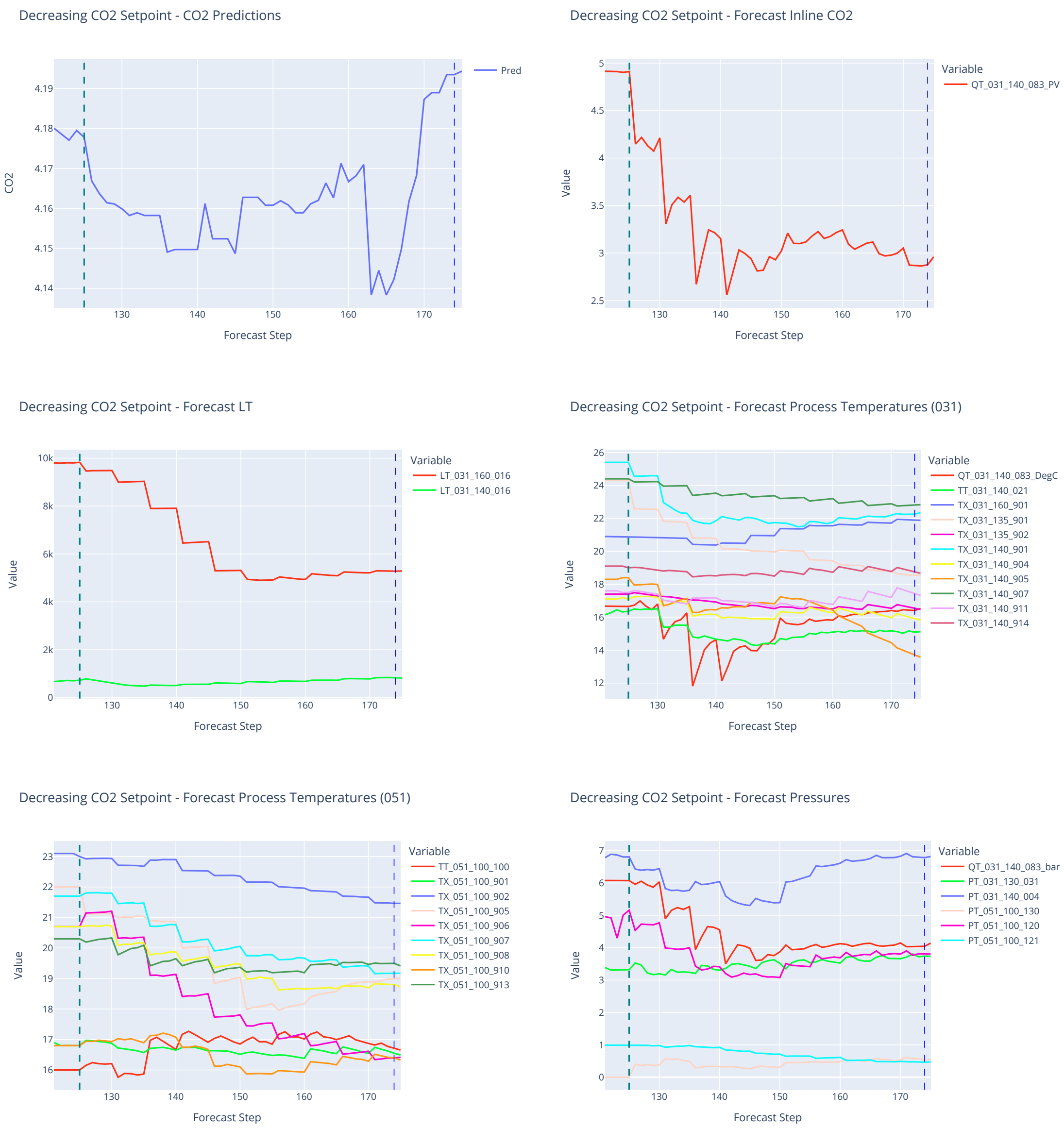


Figure 6.23: The forecast for all of the variables in Scenario 5. The teal and blue line indicate the start of the forecast and the point where the forecast input reaches the values suggested by the CFE respectively.

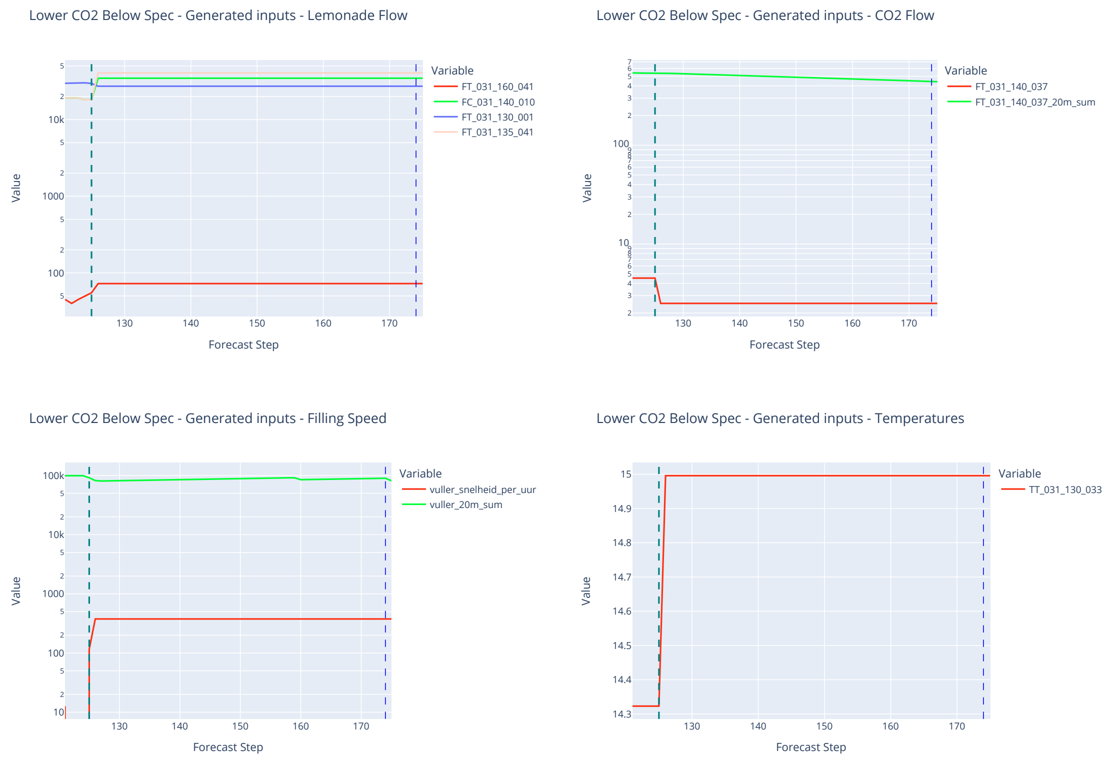


Figure 6.24: The generated CFE, forecast input, for Scenario 6. The teal and blue line indicate the start of the forecast and the point where the forecast input reaches the values suggested by the CFE respectively.

6.6. Discussion

We will now discuss the results obtained by applying the proposed methodology to the real-world data set by going over the performance of each of the three components of the method, after which we try to confirm or reject the hypotheses we formulated in Section 5.4.

6.6.1. Target Prediction

Analysing the results reveals that the target prediction model is able to predict the amount of CO_2 in a bottle with a mean absolute deviation of 0.0050 compared to the mean value of 4.2022 in the dataset. It is clear from the plots of the predictions in Figure 6.11 that the model is able to follow the general trends of the test samples quite clearly. However, the model still has difficulties with the larger fluctuations in the test set. The primary reason for this issue is probably that the dataset is too imbalanced for the random forest to determine the cause of these spikes, as demonstrated by the sample value distributions seen in Figure 6.10. However, it is also important to note that the labels may not represent the problem very well, as the raw CO_2 values are entered manually by operators after a measurement. These operators are held personally accountable if the values are too high or too low, which may lead to them entering incorrect values in their own best interest. This means that the model can create a wrong association between a bad process state and a good bottle CO_2 value. Furthermore, some sensors have a low resolution, such as `TT_051_100_100` which denotes the filling tank temperature and only has a resolution of 1°. When a change of a few degrees can have a high impact on the result, this could mean that the random forest had a harder time creating conclusions based on this sensor.

6.6.2. CFEs

From the CFEs generated in the scenarios we can conclude that they are able to effectively show what variables need to be changed for the target prediction model to predict the desired values. In Scenarios 1 and 6 the goal was to lower the CO_2 the CFEs suggested to decrease the CO_2 flow and increase the

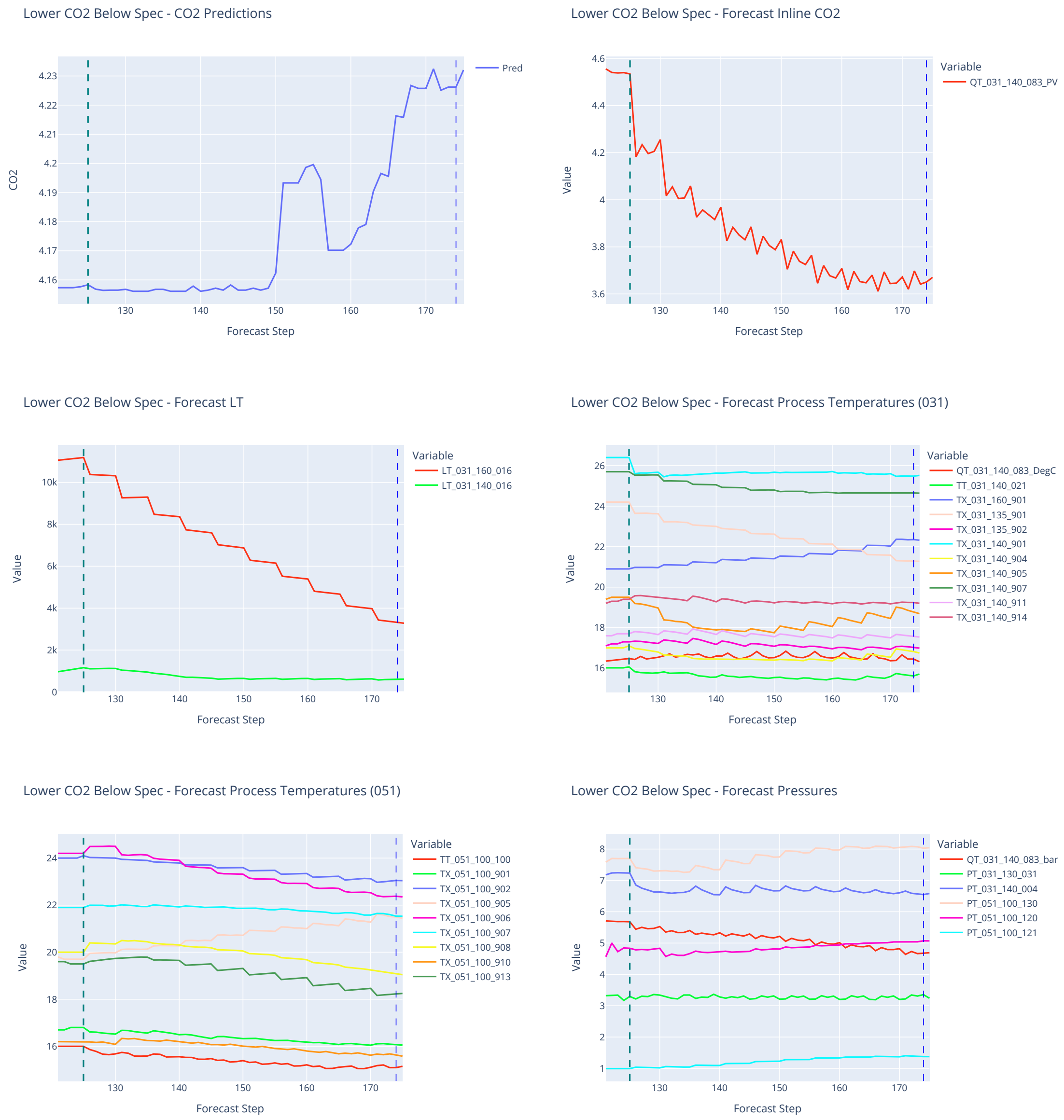


Figure 6.25: The forecast for all of the variables in Scenario 6. The teal and blue line indicate the start of the forecast and the point where the forecast input reaches the values suggested by the CFE respectively.

lemonade flow into the mixer, which is in line with Hypothesis 1. Furthermore, temperatures are required to increase in sensors on lemonade pipes, such as `TX_031_135_901`, but also on `TX_051_100_908`, residing on a filler pipe used for transport of bottle cleaning fluid. This pipe will increase in temperature when not used, and thus this increase can be interpreted as a desired decrease in filling speed. which is in line with Hypothesis 3. An increase is also required in temperature sensor `TX_051_100_913` which related pipe is used for the transport of CO_2 gas to keep the filler tank pressurized, and thus an increase in temperature means a reduction in added gas. This could mean that the desire is to reduce the pressure in the tank, or that the system is more stable such that the adding of gas is not needed, but as we do not have pressure information on the filler tank it is hard to make any conclusions.

In Scenarios 2 and 4 the goal was to increase the CO_2. Both CFEs required a decrease in aggregated filling speed and in lemonade flow into the carbonation tank and cooler. The fact that Scenario 2 did not require an aggregated filling speed increase goes against Hypothesis 3 and it is unclear why this is not required. The fact that the CO_2 was forecast to increase to 4.4 may indicate that a further filling speed increase may have increased the CO_2 too much. While in Scenario 2 the incoming CO_2 was required to increase in accordance with Hypothesis 1, this was not the case in Scenario 4 where the restrictions in filling speed may have prevented an increase in incoming CO_2 in the CFE. However, the CFE did contain a required increase in inline pressure which agrees with Hypothesis 5. It is unclear why in both scenarios temperature increases were required in lemonade pipes and tanks, as this disagrees with Hypothesis 2, but this could be due to the decrease in lemonade flow being correlated with higher temperatures. The same holds for the required temperature increase in the gas pipe for cleaning bottles which is likely in correlation with the filling speed which is required to decrease in both scenarios.

In Scenario 3 the goal was to keep the CO_2 constant but the changes by the CFE do not seem to be in line with expectations. First of all, the amount of lemonade fed into the process and filling speed are decreased while the amount of incoming CO_2 is increased. According to hypotheses 1 we would expect the changes to increase the bottle CO_2 more than the temperature changes would counteract. But then again, it would be more logical for the CFE to give no changes at all as the original prediction for bottle CO_2 was within the accepted bounds already. The fact that changes were still required is probably a result of the fact that the CFE returned is not optimal but the result of finding a solution with a bounded distance to the current process state.

Lastly, in Scenario 5 the goal was to determine how the effects of lowering the amount of incoming CO_2 could be counteracted. The CFE required increased lemonade flow into the cooler, carbonation tank, and mixer, while reducing flow to the filler. Considering Hypothesis 1, we would expect the bottle CO_2 to decrease rather than increase, but it could also be the case that the effects of increased lemonade flow would decrease temperatures and thus increase bottle CO_2 in line with Hypothesis 2. The CFE also requires an increase in the inline pressure measurement, which agrees with Hypothesis 5, and an increase in inline CO_2, but this seems unrealistic as the amount of incoming CO_2 is being lowered, preventing an increase in inline measurements. Finally, the CFE requires an increase in temperatures in pipes related to filling while an increase can only be realized with a decrease in filling speed, which is not required by the CFE. This again suggests that the generated CFEs do not fully capture causality within the process.

Considering all CFEs together it is clear that the target prediction model does not consider all variables to be important. We can confirm this by considering the permutation importances for the model in Table 6.2, where we can see that there is a very steep drop off of importance to where only the first twelve features have an importance of at least 0.01. The only pressure sensor actually subject to a required change was the inline pressure measurement, which could be due to the fact that the pressure in all pipes is well regulated as can be seen in Figure A.4. Therefore, the model probably has not attributed change in bottle CO_2 to changes in pressure in the other sensors and thus did not create splits on these features. The same can be said for many of the temperature sensors, but this is to be expected as the `TX` sensors are placed on the outside of pipes, making them react slowly to changes within said pipes, as well as the fact that many pipes are used only during cleaning or draining rather than during filling. Thus, these sensors only provide limited information on the prediction of bottle CO_2, making the target prediction model split based on other variables instead. The fact that some sensors are ignored and thus never subject to a required change can be misleading, or even an issue, as the reality may be that those sensors are correlated with other sensors and thus would take another value if the new process parameters from the CFE would be applied to the process. This could result in the bottle CO_2 being predicted differently. Finally, we were able to confirm multiple hypotheses stated in Section 5.4 apart from Hypothesis 3 and 4. We hypothesized that lower filling speed and higher room temperature would lead to increased temperatures across all sensors, but were unable to confirm this using the CFEs. This could be due to the fact that the CFEs struggle to capture causality between different variables in the process.

6.6.3. Process State Forecasting

Considering our analyses of the different scenarios in Section 6.6.2, we can now discuss whether the results of the process state forecasting are in line with what we would expect from applying the process parameters changes required by the CFEs.

In Scenario 1 and 6 the goal was to lower bottle CO_2. This was partially achieved in the first scenario as the CO_2 started increasing again after 40 time steps. It is unclear why this is the case but it could potentially be due to the fact that the filler tank temperature is decreasing and multiple pressures are increasing which would lead to increased bottle CO_2 according to Hypothesis 2 and 5, even though the inline CO_2 is decreasing as expected. A similar pattern is seen in Scenario 6 where the bottle CO_2 is never forecast to decrease. Furthermore, the fact that the level of the buffer tank decreases to zero is unrealistic as this is not allowed to happen, indicating that the forecasting model does not capture all facts about the bottle filling machine.

In Scenario 2 and 4 the goal was to increase bottle CO_2 and the CO_2 indeed shows an increase in both cases. In both Scenarios the inline CO_2 measurement is forecast to increase but this seems unrealistic for Scenario 4, as the incoming CO_2 was reduced by the CFE, especially when considering that the carbonation tank was almost completely run dry, guaranteeing that there is a low amount of CO_2 in the system left. In both cases filling speed is decreased and lemonade flow is reduced across the system which was hypothesized in Hypothesis 3 to increase temperatures. In the forecast of Scenario 2 temperatures did not seem to decrease, however. Even though flow into the carbonation tank is set to almost zero the temperature of that tank is still decreasing as if it was still receiving flow. This could also be due to the fact that its temperature was high in the first place and is now being decreased due to the lemonade in it. We can see an increase in the inline temperature measurement and another temperature sensor on a pipe to the filler (`TX_100_100_901`), but then one would expect an effect on filler tank temperature which is not present. In Scenario 4 the results are more according to expectations as most temperatures are increasing except for some temperatures related to drainage (`TX_031_140_901`) and filling (`TX_051_100_906`, `TX_051_100_908`, and `TX_051_100_913`). Pressures in the forecast of Scenario 2 are all increasing, probably due to the consistent filling speed, which would be desired according to Hypothesis 5. In Scenario 4 the inline pressure is increasing as well, which is logical as barely any filling is done and still flow is provided to the filler. Another important aspect to consider is the fact that the inline measurements seem to be highly correlated, which could be a sign of multicollinearity. This could make the forecast less realistic as it is not logical that CO_2, pressure, and temperature are so strongly correlated.

In Scenario 3 the goal was to keep the CO_2 constant, but the forecast is almost the same as the forecast of Scenario 2, probably due to the fact that in both cases the CFE required an increase in CO_2, increasing in both the inline CO_2 forecast.

Lastly, in Scenario 5 the goal was to determine how to counteract the influence of decreasing the amount of incoming CO_2. The forecast indeed shows a gradual decrease in inline CO_2, while the CO_2 forecast stays relatively constant. Except for the flow to the filler, the CFE required all flows to increase which caused a global decrease in temperatures in line with Hypothesis 3. In the bottle CO_2 forecast we can first see a decrease and then an increase which can be attributed to the fact that the pressures also first decrease and then increase again. This pattern can be seen in multiple variables, such as the buffer tank level, but there is not a good reason for this sudden change as the flows stay the same. It seems that the forecast model has captured the preferred lower bound on the level of the buffer tank, but logically it is not possible to maintain this without altering the flows in the process. This signals that future work should focus on better constraining the flows and levels in the process in order to make the results more plausible.

Considering all scenarios together we were able to show that we are able to effectively forecast the process state and that the results generally in line with our hypotheses stated in 5.4, apart from Hypotheses 4 as we could not confirm whether room temperature is of effect on the process state. The forecast tends to be unrealistic when it comes to flows and tank levels, indicating that future work is needed to further integrate domain knowledge on this aspect into the methodology. Furthermore, the model seems to be subject to multicollinearity, indicating that feature selection may also be done to ensure that

Feature	Importance ($\mu \pm \sigma$)
QT_031_140_083_PV	0.524 ± 0.018
FT_031_140_037_20m_sum	0.267 ± 0.013
LT_031_160_016	0.112 ± 0.013
TX_051_100_907	0.109 ± 0.006
TX_051_100_913	0.093 ± 0.008
QT_031_140_083_bar	0.063 ± 0.002
TX_051_100_908	0.023 ± 0.002
TX_051_100_905	0.019 ± 0.003
FT_031_140_037	0.016 ± 0.001
PT_051_100_121	0.014 ± 0.001
TX_031_135_901	0.012 ± 0.001
PT_051_100_130	0.01 ± 0.001
TT_031_130_033	0.009 ± 0.001
TX_051_100_906	0.009 ± 0.001
TX_031_140_905	0.008 ± 0.001
TX_031_160_901	0.008 ± 0.001
TX_031_140_907	0.007 ± 0.001
vuller_20m_sum	0.007 ± 0.001
TX_051_100_910	0.007 ± 0.0
TX_031_140_901	0.007 ± 0.001
PT_051_100_120	0.006 ± 0.0
TX_051_100_902	0.006 ± 0.002
FT_031_130_001	0.005 ± 0.0
PT_031_130_031	0.004 ± 0.0
LT_031_140_016	0.003 ± 0.0
TX_031_140_914	0.003 ± 0.0
TX_031_140_911	0.003 ± 0.0
FT_031_135_041	0.003 ± 0.0
TT_031_140_021	0.002 ± 0.0
vuller_snelheid_per_uur	0.002 ± 0.0
PT_031_140_004	0.001 ± 0.0
TX_051_100_901	0.001 ± 0.0
FT_031_160_041	0.001 ± 0.0
QT_031_140_083_DegC	0.001 ± 0.0
FC_031_140_010	0.001 ± 0.0
TX_031_140_904	0.001 ± 0.0
TT_051_100_100	0.001 ± 0.0
TX_031_135_902	0.001 ± 0.0

Table 6.2: Permutation importances for the target prediction model trained on the real dataset.

the model does not learn erroneous relations between variables.

6.6.4. Discussion with the Thesis Company Expert

In a brief evaluation with a thesis company expert some of the results were discussed in order to determine whether they are realistic. First of all, the fact that CFEs suggest lower lemonade flow to increase the CO_2 to lemonade ratio could be confirmed by the expert to be true. Furthermore, the idea of an idling bottle filler increasing the temperatures in the process, formulated in Hypothesis 3, did also seem realistic and this is an aspect the company wants to further investigate. A problem highlighted by the expert was, however, that it is not always possible to further reduce filling temperatures as this may result in condensation on the bottles which is undesirable for multiple reasons. An interesting domain knowledge-related point is that this means that filling temperature can be lower in the winter than in the summer. Another idea uncovered by the model, that higher temperatures in drain (e.g. `TX_031_140_902`) and filler tank pressure gas (`TX_051_100_913`) pipes equate to higher bottle CO_2, also seemed logical to the expert, as these pipes are only used, and thus low in temperature during the start up of the process and when the filler is idling. An interesting point raised by the expert was that he was able to recognize the step-based pattern in the forecast, which is also visible in the real data, but could also be mistaken for a quirk of the neural network predictions. In the real data this pattern is the result of the carbonation tank being kept at the right pressure in bursts.

Although the forecasts seem to be mostly in line with expectations, temperatures 30°C and up during production are not realistic. Furthermore, the method sometimes reduces flow to zero, which is not allowed in the first place, but then also forecasts the tank at the end of that flow to stay at a continuous level while still giving off flow itself. This is not realistic and shows a need for further integration of domain knowledge.

7 Conclusion

The goal of this thesis was to create a novel approach to quality prediction for production processes with the ability of giving explanations to its predictions aimed at uncovering the hidden workings of a production process. In order to increase the plausibility of the explanations, domain knowledge in the form of constraints can be applied. The main research question this thesis sought out to answer was:

> How can a Machine Learning model be developed which predicts product quality, while giving production-process-specific explanations that domain-specific stakeholders can understand?

To answer this question a methodology was developed which combines three separate fields of research: product quality prediction, explainable artificial intelligence, and domain knowledge integration. For product quality prediction, random forests were used as they are not only flexible and transparent in their functionality, but also resilient to overfitting. Counterfactual explanations were then used to exploit the structure of the random forest to determine changes to the current process state to increase the product quality. Finally, process forecasting is used to determine the impact of the proposed process parameter changes. This methodology was then used to answer the following sub-questions:

- Can random forests be used to indicate what variables influence product quality?
- Can counterfactual explanations be used to suggest a better production process state?
- Can time series forecasting be used to forecast the influence of process parameter changes?
- Can domain knowledge be used to make counterfactual explanations more plausible with respect to production processes?

7.1. Summary of this Thesis

After the introduction, this thesis started off with a sketch of the project's context in Chapter 2. The problem of bottle filling was introduced, together with exploration of the target bottle filling machine and the data that it produces. We also explored an artificially created dataset with the purpose of highlighting what the methodology could achieve. The related literature with respect to product quality prediction, explainable artificial intelligence, and domain knowledge integration was discussed in Chapter 3. It was noted that little research exists which come close to the goals set out for this thesis, but that the necessary methods exist to bridge the gap. These methods, i.e. the preliminaries, were introduced in Chapter 4. Product quality prediction was formally introduced and it was shown how random forests can be used for quality prediction. Then, counterfactual explanations for decision trees were explained as they are able to exploit the structure of a tree-based model to determine how a process state needs to change for the model to change its prediction to be more desirable. Chapter 5 then gave the contributions of this research and the structure of the proposed methodology. We discussed how counterfactual explanations can be used with random forests, and how the optimization of finding the counterfactual explanations can be constrained with domain knowledge to increase their plausibility. We then covered what domain knowledge constraints look like and how they can be applied. The last element of the methodology, process forecasting, is introduced as a way to determine the impact of the proposed process parameter changes. We rounded of the contributions by formulating hypotheses and expectations before applying the methodology to the available data. We hypothesized that the explanations should highlight lower temperatures and higher pressures to increase CO_2 in bottles, but also that it may be possible that the counterfactual explanations only suggest changes to a small number of variables that are important to the random forest. In Chapter 6 we then started off with discussing how the data was cleaned and prepared for use, after which we went over the different constraints that could be determined from the known facts about the process. We continued with covering the exact structure of the random forest and neural networks used in the methodology and how their hyperparameters were determined. We then first discussed the results when applying the methodology on the artificial dataset and concluded that the methodology was able to correctly identify how the artificial process could be improved, but also that features not important to the random forest model were not considered in the counterfactual explanations, which is misleading. In the results on the real data, we concluded that the methodology was able to identify problems with the current process state and provide changes that were forecast to improve the process state such that it produced bottles with a more desired amount of CO_2. We then discussed some of the problems with the methodology, including that the random forest only suggests changes to variables it splits the data on and that the process state forecasting sometimes gives unrealistic results, probably due to unrealistic generated forecasting input and collinearity in the process state forecasting model. Nevertheless we were able to confirm the hypotheses stated in Section 5.4.

7.2. Conclusions

We can answer the first research question *Can random forests be used to indicate what variables influence product quality?* positively as we have demonstrated that random forests are able to show through feature importance and counterfactual explanations what variables one can use to determine the product quality in a bottle filling machine setting. With respect to the second research question *Can counterfactual explanations be used to suggest a better production process state?*, we can state that it is indeed possible for counterfactual explanations to suggest better production process states. The suggested states show which changes need to be made to the process for the target prediction model to improve its prediction. However, it should be noted that counterfactual explanations generated using the structure of a random forest only suggest changes to variables on which the random forest has split its training data, meaning that the other variables are ignored. This may result in unrealistic process states as some variables change while other, possibly correlated, variables do not change with them. Having trained a neural network on the large amount of time series data available of the bottle filling machine, and having used it to forecast the impact of the process parameters obtained by generating counterfactual explanations, we can answer the third research question, *Can time series forecasting be used to forecast the influence of process parameter changes?*. It is indeed possible to forecast the impact and the results are in line with our hypotheses, but the results are not always realistic, possibly due to multicollinearity and lack of constraints on flows and tank levels in the specific case of the bottle filling machine. By formulating a number of constraints based on factual knowledge about the bottle filling machine and enforcing those constraints in the generation of counterfactual explanations we are able to confirm our last research question, *Can domain knowledge be used to make counterfactual explanations more plausible with respect to production processes?*, as well, as we have shown that these constraints remove implausible process states from the search space of counterfactual explanations and forecast input generation.

7.3. Contributions

The contributions of this thesis are then as follows:

1. A novel methodology for providing domain knowledge driven explainable product quality prediction, making use of process state time series data, as well as the limited number of samples of the target variable.
2. A guideline for working with production process data.
3. Insights into the creation and use of domain knowledge constraints to improve explanation plausibility.
4. Recommendations to the thesis company based on findings obtained by applying the methodology to data of their bottle filling machine.

7.4. Limitations & Recommendations for Future Work

The limitations of this project of this project can be divided into problems with the target problem itself and the limitations of the proposed methodology.

7.4.1. Data

Even though this research has shown that it is possible to predict the CO_2 in filled bottles there was first of all a limitation in data available to the problem which prevented greater predictive accuracy. As discussed in Section 2.2.1, the amount of CO_2 in a bottle is determined by a function of temperature and pressure. However, for the filler tank only a temperature sensor was available, which means that the model needs to use data further away from the filler to determine the pressure and CO_2 contents of filled lemonade, making the problem harder. Furthermore, the available bottle samples are skewed as operators enter the results manually. Values which are too low or too high trigger alarms which operators are personally held accountable for, meaning that operators may, for instance, enter slightly higher values if it means that alarms are not triggered. This means that the model now has a harder time when training on the sample data, as there are less out-of-bounds samples and some of the in-bounds samples now have inaccurate values. An important takeaway from these issues is that applying artificial intelligence to industrial processes requires a lot of data whose sources are close to the target and is not subject to human biases which muddy the relation between the target and other variables. Biases within the sensors should be made clear to the model developer as a simple upwards linear trend within reported sensor values due to calibration issues may render its values useless for use in machine learning prediction. With respect to the target bottle filling machine of this project, a recommendation would be to place additional pressure, level, and CO_2 sensors in the filler tank, which are expected to greatly improve the predictive quality of this model. Furthermore, automation in the taking of bottle samples could make sure that the sample values entered in the database are actually accurate and not skewed.

7.4.2. Target Prediction

When considering limitations to the quality prediction methods used in this research, the first limitation is the fact that time-related information is only partially considered when predicting the target value. Only aggregation variables are considered which denote the sum over a fixed window of past data. An interesting direction for research is to extend the model input by explicitly adding past values of variables or considering entire time series. Counterfactual explanations can then reveal the importance of past variables and what they should have been, or they can be locked to observed past values to ensure a new process state is generated which can follow from the current state. This research should also point out whether the increase in complexity of the random forest, and thus increase the time complexity of finding counterfactual explanations, does not make the methodology too slow to be used in a live setting.

7.4.3. Counterfactual Explanations

Although the counterfactual explanations are able to indicate what changes need to be done to the original process state, the counterfactuals will only suggest changes to variables which are important (i.e. used) by the random forest model. It is important to note, however, that domain knowledge can (and should) be used to enforce changes in the other variables as well, as this increases the plausibility of the resulting counterfactual explanations. A very interesting research direction is to use this methodology on other variables in the process than the target variable, e.g. the filler tank temperature or the inline CO_2 measurement. This may better uncover how the target variable is influenced by the variables important to its prediction. Furthermore, no variable weights were specified when determining the distance between two process states in an optimization setting. As the data was standardized before distance calculation, this means that each variable effectively contributes equally to distance. In the future domain knowledge could be used to determine which variables are allowed to change more than others.

7.4.4. Process State Forecasting

Although the neural network model used for the forecasting of the process state was able to forecast with accuracy, sometimes the results were unrealistic as the resulting forecast showed great fluctuation or changes which were too great in a short amount of time. An interesting research direction is to take the domain knowledge constraints used in the counterfactual generation and forecasting input generation and enforce them in the forecasting of the process state, resulting in a more realistic process state forecast.

Furthermore, as more complicated explainable time series-based process forecasting models were out of the scope of this research, future work should aim at determining whether such models are able to better explain what variables are of influence on the process forecast. Also, no feature selection was done to increase the performance of the forecasting model as it considers exactly those features also used by the target prediction model.

7.4.5. Miscellaneous

With respect to the proposed methodology as a whole there is also more general future work to be considered. A major element left out of this research is the incorporation of the results of process state forecasting into the generation of counterfactual explanations. It was theorized that instead of optimizing over vector distance to determine the optimal counterfactual explanation, that the time-based cost predicted by the process state forecasting model could be used. However, this would mean that either these costs are determined beforehand, or that one actually optimizes over the output of a machine learning model, which would be computationally much harder compared to optimizing over a vector distance. Another important direction for future work is to further improve the way explanations are provided to the user. Currently, the output of the methodology still requires the user to make the conclusions to obtain qualitative results. Ideally, this is automated and aggregated over the entire dataset so that the user can quickly obtain all needed information about a process. Furthermore, this research has focused on local explanations (i.e. the explaining of single process states). This could be extended to also incorporate global explanations, for instance, using SHAP for random forests or the aggregation of decision paths. With respect to the domain knowledge integration, all constraints added to the explanation and parameter optimization processes are considered to be fact and thus if conflicting constraints are added the optimization may fail to find a solution. Z3 offers the ability to specify constraints to be soft, allowing the optimizer to drop them if this allows for 'better' solutions. It is important to note that, with an eye on explainability, this dropping of constraints should be made clear to the user as they may not be aware of the fact that their domain knowledge is not being used anymore. Additionally, the impact of domain knowledge constraints could be determined by comparing results with and without them enabled.

7.4.6. Recommendations to the Thesis Company

Finally, we conclude with discussing recommendations to the thesis company based on the findings of this research. First of all, it seems that we can confirm nearly all of the hypotheses formulated in Section 5.4, which should prove a great starting point for the company to investigate further the reason for the great fluctuations in bottle CO_2. The CFEs have shown that the relation between incoming lemonade and CO_2 flow (Hypothesis 1) do indeed attribute to the final bottle CO_2, but it is also clear that the inline CO_2 measurement does not always reflect the actual bottle CO_2. The CO_2 is also greatly influenced by temperature (Hypothesis 2), which in turn is influenced by filling speeds which influences the flow of lemonade through the process (Hypothesis 3). Therefore, to improve the stability of CO_2 in the produced bottles the amount of stops of the filler should first of all be minimized. Furthermore, it seems that a slower, not stopping filler should be more desirable than a filler working in short, high output bursts as this would result in great fluctuations in temperature and thus unpredictable bottle CO_2 changes. The effect of pressure (Hypothesis 5) was confirmed by the model, but as pressure can be more easily regulated this should not be the focus starting off. If the company desires to apply data-driven methods in the future, a focus should also be put on improving the amount and quality of bottle sample data. The fact that there are only a couple thousand samples available, coupled with the fact that the individual samples do not reflect the actual output of the bottle filling machine at that given point severely bottlenecks the quality of predictive models. Nevertheless, the results of this project should give a good direction for further investigation of the bottle filling machine.

7.5. Reflection

The experience of writing this thesis has been on an entirely new level for me when compared to the previous projects I have attended up until this point. The past ten months featured many ups and downs, but in the end, I feel that I not only contributed to the three major fields this research bridges, but that I also learned a lot as well.

With respect to what I have gained, I want to start off with saying that this research has showed me that I want to continue working with explainable artificial intelligence as I feel that the insights these kinds of methods give are really interesting compared to non-explainable methods which usually only provide basic predictions. I knew little about explainable artificial intelligence when starting off, and this thesis has opened many doors for me, while also showing me that the industry still needs to pick up in this area. Furthermore, the knowledge I gained about applying all kinds of methods in a real world setting was exactly what I hoped to achieve during this project, as the real world is not something you come in contact with a lot within the doors of the university, and I believe this has also prepared me for what lies beyond my studies. I want to add that this project has also showed me that there is still much to learn even specific to this project, as I continuously wanted to explore additional ways of solving the problems I encountered, even though towards the end of the project I had to settle on the progress I had made. I am looking forward to continue on this quest of learning after this project.

While I have learned many things in this project, it has also been one of, if not the hardest, projects in my career. First of all, due to the distance between me and the company, and the fact my project started off still in the middle of the COVID-crisis, made it hard for me to always be on location to get to meet the people important to this project. This may also be the cause of the fact that I started off with struggling with pinpointing what would be a good subject for the thesis and finding a balance between the goals of the company and scientific contributions. During this process while I explored existing research I sometimes decided on too complex subjects and made it too hard for myself. I would have put more emphasis on gathering all the knowledge about the bottle filling machine, which I ended up doing that in the later stages of the project any way, as I believe that would have better guided me to a suitable topic. This would have also helped me see that the initial dataset I was working on did not represent the target variable, which lead me to spend time on methods for noisy datasets while this was not necessary for the problem. Furthermore, I learned that it is important to fully consider the compatibility of theorized extensions to the project. Initially, the time-series process state forecasting method was meant to serve for the calculation of a time-based cost function for the counterfactual explanation generation, but I eventually learned that this was too complex and I had to resort to using the forecast and cost as explanations only, while the time put into these methods could also have been spent on other domain-knowledge-based or explanation methods. Nevertheless, the many topics that I worked on led to great opportunities for future work and I believe that may be even more important.

I want to end on a high note and say that even though this project has definitely been difficult, I am happy with the result and I hope that the knowledge I gained during the last ten months will serve me well in the years to come.

References

[1] Assaf, Roy. "Explainable Deep Neural Networks for Multivariate Time Series Predictions". In: *IJCAI* 28 (Aug. 2019), pp. 6488–6490. URL: https://www.ijcai.org/proceedings/2019/0932.pdf.

[2] Liang Bao, Xiaofeng Yuan, and Zhiqiang Ge. "Co-training partial least squares model for semi-supervised soft sensor development". en. In: *Chemometrics and Intelligent Laboratory Systems* 147 (Oct. 2015), pp. 75–85. ISSN: 01697439. DOI: 10.1016/j.chemolab.2015.08.002. URL: https://linkinghub.elsevier.com/retrieve/pii/S0169743915001938 (visited on 03/18/2022).

[3] Gérard Biau and Erwan Scornet. "A random forest guided tour". In: *Test* 25.2 (2016). Publisher: Springer, pp. 197–227.

[4] Borovykh, A., Bohte, S., and Oosterlee, C. W. "Conditional time series forecasting with convolutional neural networks". In: (Mar. 2017).

[5] G.E.P. Box. "Robustness in the Strategy of Scientific Model Building". en. In: *Robustness in Statistics*. Elsevier, 1979, pp. 201–236. ISBN: 978-0-12-438150-6. DOI: 10.1016/B978-0-12-438150-6.50018-2. URL: https://linkinghub.elsevier.com/retrieve/pii/B9780124381506500182 (visited on 02/18/2022).

[6] I.T. Cameron and G.D. Ingram. "A survey of industrial process modelling across the product and process lifecycle". en. In: *Computers & Chemical Engineering* 32.3 (Mar. 2008), pp. 420–438. ISSN: 00981354. DOI: 10.1016/j.compchemeng.2007.02.015. URL: https://linkinghub.elsevier.com/retrieve/pii/S0098135407000567 (visited on 01/27/2022).

[7] Penny Chong et al. "Mouse Authentication Without the Temporal Aspect – What Does a 2D-CNN Learn?" In: *2018 IEEE Security and Privacy Workshops (SPW)*. San Francisco, CA: IEEE, May 2018, pp. 15–21. ISBN: 978-1-5386-8276-0. DOI: 10.1109/SPW.2018.00011. URL: https://ieeexplore.ieee.org/document/8424627/ (visited on 10/15/2021).

[8] Tirtharaj Dash et al. "Incorporating Domain Knowledge into Deep Neural Networks". In: *arXiv:2103.00180 [cs]* (Mar. 2021). arXiv: 2103.00180. URL: http://arxiv.org/abs/2103.00180 (visited on 10/18/2021).

[9] Glenn De'ath and Katharina E. Fabricius. "CLASSIFICATION AND REGRESSION TREES: A POWERFUL YET SIMPLE TECHNIQUE FOR ECOLOGICAL DATA ANALYSIS". en. In: *Ecology* 81.11 (Nov. 2000), pp. 3178–3192. ISSN: 0012-9658. DOI: 10.1890/0012-9658(2000)081[3178:CARTAP]2.0.CO;2. URL: http://doi.wiley.com/10.1890/0012-9658(2000)081[3178:CARTAP]2.0.CO;2 (visited on 11/03/2021).

[10] Jürgen Dieber and Sabrina Kirrane. "Why model why? Assessing the strengths and limitations of LIME". In: *arXiv:2012.00093 [cs]* (Nov. 2020). arXiv: 2012.00093. URL: http://arxiv.org/abs/2012.00093 (visited on 09/09/2021).

[11] Filip Karlo Dosilovic, Mario Brcic, and Nikica Hlupic. "Explainable artificial intelligence: A survey". In: *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. Opatija: IEEE, May 2018, pp. 0210–0215. ISBN: 978-953-233-095-3. DOI: 10.23919/MIPRO.2018.8400040. URL: https://www.researchgate.net/profile/Mario-Brcic/publication/325398586_Explainable_Artificial_Intelligence_A_Survey/links/5b0bec90a6fdcc8c2534d673/Explainable-Artificial-Intelligence-A-Survey.pdf (visited on 09/09/2021).

[12] Chenyou Fan et al. "Multi-Horizon Time Series Forecasting with Temporal Attention Learning". en. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Anchorage AK USA: ACM, July 2019, pp. 2527–2535. ISBN: 978-1-4503-6201-6. DOI: 10.1145/3292500.3330662. URL: https://dl.acm.org/doi/10.1145/3292500.3330662 (visited on 09/22/2021).

[13] Marco Gario and Andrea Micheli. “PySMT: a solver-agnostic library for fast prototyping of SMT-based algorithms”. In: *SMT Workshop 2015*. 2015.

[14] Zhiqiang Ge and Xinru Chen. “Dynamic Probabilistic Latent Variable Model for Process Data Modeling and Regression Application”. In: *IEEE Transactions on Control Systems Technology* 27.1 (Jan. 2019), pp. 323–331. ISSN: 1063-6536, 1558-0865, 2374-0159. DOI: 10.1109/TCST.2017.2767022. URL: https://ieeexplore.ieee.org/document/8106664/ (visited on 03/18/2022).

[15] Zhiqiang Ge and Zhihuan Song. “Semisupervised Bayesian method for soft sensor modeling with unlabeled data samples”. en. In: *AIChE Journal* 57.8 (Aug. 2011), pp. 2109–2119. ISSN: 00011541. DOI: 10.1002/aic.12422. URL: https://onlinelibrary.wiley.com/doi/10.1002/aic.12422 (visited on 03/18/2022).

[16] Len Gelman et al., eds. *Advances in Condition Monitoring and Structural Health Monitoring: WCCM 2019*. en. Lecture Notes in Mechanical Engineering. Singapore: Springer Singapore, 2021. ISBN: 9789811591983 9789811591990. DOI: 10.1007/978-981-15-9199-0. URL: http://link.springer.com/10.1007/978-981-15-9199-0 (visited on 03/28/2022).

[17] Julie Gerlings, Millie Søndergaard Jensen, and Arisa Shollo. “Explainable AI, but explainable to whom?” In: *arXiv:2106.05568 [cs]* (June 2021). arXiv: 2106.05568. URL: http://arxiv.org/abs/2106.05568 (visited on 01/31/2022).

[18] Benyamin Ghojogh and Mark Crowley. “The Theory Behind Overfitting, Cross Validation, Regularization, Bagging, and Boosting: Tutorial”. In: *arXiv:1905.12787 [cs, stat]* (May 2019). arXiv: 1905.12787. URL: http://arxiv.org/abs/1905.12787 (visited on 01/28/2022).

[19] Przemyslaw Grzegorzewski, Andrzej Kochanski, and Janusz Kacprzyk, eds. *Soft Modeling in Industrial Manufacturing*. Vol. 183. Studies in Systems, Decision and Control. Cham: Springer International Publishing, 2019. ISBN: 978-3-030-03200-5 978-3-030-03201-2. DOI: 10.1007/978-3-030-03201-2. URL: http://link.springer.com/10.1007/978-3-030-03201-2 (visited on 02/17/2022).

[20] Tian Guo, Tao Lin, and Nino Antulov-Fantulin. “Exploring Interpretable LSTM Neural Networks over Multi-Variable Data”. In: *arXiv:1905.12034 [cs, stat]* (May 2019). arXiv: 1905.12034. URL: http://arxiv.org/abs/1905.12034 (visited on 10/15/2021).

[21] Maya Gupta et al. “Monotonic Calibrated Interpolated Look-Up Tables”. In: *arXiv:1505.06378 [cs]* (Jan. 2016). arXiv: 1505.06378. URL: http://arxiv.org/abs/1505.06378 (visited on 10/13/2021).

[22] Suryabhan Singh Hada and Miguel Á Carreira-Perpiñán. “Exploring counterfactual explanations for classification and regression trees”. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2021, pp. 489–504.

[23] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. “Random Forests”. In: *The Elements of Statistical Learning*. Series Title: Springer Series in Statistics. New York, NY: Springer New York, 2009, pp. 587–604. ISBN: 978-0-387-84857-0 978-0-387-84858-7. DOI: 10.1007/978-0-387-84858-7_15. URL: http://link.springer.com/10.1007/978-0-387-84858-7_15 (visited on 03/01/2022).

[24] Yang Hui et al. “Assembly quality evaluation for linear axis of machine tool using data-driven modeling approach”. en. In: *Journal of Intelligent Manufacturing* (Oct. 2020). ISSN: 0956-5515, 1572-8145. DOI: 10.1007/s10845-020-01666-y. URL: https://link.springer.com/10.1007/s10845-020-01666-y (visited on 01/28/2022).

[25] Alexey Ignatiev et al. “From contrastive to abductive explanations and back again”. In: *International Conference of the Italian Association for Artificial Intelligence*. Springer, 2020, pp. 335–355.

[26] Alexey Ignatiev et al. “Using MaxSAT for Efficient Explanations of Tree Ensembles”. In: AAAI, 2022.

[27] Yacine Izza and Joao Marques-Silva. “On Explaining Random Forests with SAT”. In: *arXiv:2105.10278 [cs]* (May 2021). arXiv: 2105.10278. URL: http://arxiv.org/abs/2105.10278 (visited on 03/22/2022).

[28] Sami Ben Jabeur, Salma Mefteh-Wali, and Jean-Laurent Viviani. "Forecasting gold price with the XGBoost algorithm and SHAP interaction values". en. In: *Annals of Operations Research* (July 2021). ISSN: 0254-5330, 1572-9338. DOI: 10.1007/s10479-021-04187-w. URL: https://link.springer.com/10.1007/s10479-021-04187-w (visited on 10/14/2021).

[29] Amir-Hossein Karimi et al. "Model-Agnostic Counterfactual Explanations for Consequential Decisions". In: *arXiv:1905.11190 [cs, stat]* (Feb. 2020). arXiv: 1905.11190. URL: http://arxiv.org/abs/1905.11190 (visited on 01/31/2022).

[30] George Em Karniadakis et al. "Physics-informed machine learning". en. In: *Nature Reviews Physics* 3.6 (June 2021), pp. 422–440. ISSN: 2522-5820. DOI: 10.1038/s42254-021-00314-5. URL: http://www.nature.com/articles/s42254-021-00314-5 (visited on 10/12/2021).

[31] Johannes Kruse, Benjamin Schäfer, and Dirk Witthaut. "Revealing drivers and risks for power grid frequency stability with explainable AI". In: *arXiv:2106.04341 [physics]* (June 2021). arXiv: 2106.04341. URL: http://arxiv.org/abs/2106.04341 (visited on 09/20/2021).

[32] Martin von Kurnatowski et al. "Compensating data shortages in manufacturing with monotonicity knowledge". In: *arXiv:2010.15955 [cs, math]* (Jan. 2021). arXiv: 2010.15955. URL: http://arxiv.org/abs/2010.15955 (visited on 10/13/2021).

[33] Murat Kuzlu et al. "Gaining Insight Into Solar Photovoltaic Power Generation Forecasting Utilizing Explainable Artificial Intelligence Tools". In: *IEEE Access* 8 (2020), pp. 187814–187823. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3031477. URL: https://ieeexplore.ieee.org/document/9225146/ (visited on 09/21/2021).

[34] Markus Langer et al. "What Do We Want From Explainable Artificial Intelligence (XAI)? – A Stakeholder Perspective on XAI and a Conceptual Model Guiding Interdisciplinary XAI Research". In: *Artificial Intelligence* 296 (July 2021). arXiv: 2102.07817, p. 103473. ISSN: 00043702. DOI: 10.1016/j.artint.2021.103473. URL: http://arxiv.org/abs/2102.07817 (visited on 09/13/2021).

[35] Boyi Li et al. "On feature normalization and data augmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 12383–12392.

[36] Weihua Li and S.Joe Qin. "Consistent dynamic PCA based on errors-in-variables subspace identification". en. In: *Journal of Process Control* 11.6 (Dec. 2001), pp. 661–678. ISSN: 09591524. DOI: 10.1016/S0959-1524(00)00041-X. URL: https://linkinghub.elsevier.com/retrieve/pii/S095915240000041X (visited on 03/18/2022).

[37] Yanrui Li and Chunjie Yang. "Domain knowledge based explainable feature construction method and its application in ironmaking process". en. In: *Engineering Applications of Artificial Intelligence* 100 (Apr. 2021), p. 104197. ISSN: 09521976. DOI: 10.1016/j.engappai.2021.104197. URL: https://linkinghub.elsevier.com/retrieve/pii/S0952197621000440 (visited on 10/18/2021).

[38] Bryan Lim and Stefan Zohren. "Time-series forecasting with deep learning: a survey". en. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 379.2194 (Apr. 2021), p. 20200209. ISSN: 1364-503X, 1471-2962. DOI: 10.1098/rsta.2020.0209. URL: https://royalsocietypublishing.org/doi/10.1098/rsta.2020.0209 (visited on 09/20/2021).

[39] Bryan Lim et al. "Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting". In: *arXiv:1912.09363 [cs, stat]* (Sept. 2020). arXiv: 1912.09363. URL: http://arxiv.org/abs/1912.09363 (visited on 09/20/2021).

[40] Wei-Yin Loh. "Classification and regression trees". en. In: *WIREs Data Mining and Knowledge Discovery* 1.1 (Jan. 2011), pp. 14–23. ISSN: 1942-4787, 1942-4795. DOI: 10.1002/widm.8. URL: https://onlinelibrary.wiley.com/doi/10.1002/widm.8 (visited on 05/20/2022).

[41] Scott Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *arXiv:1705.07874 [cs, stat]* (Nov. 2017). arXiv: 1705.07874. URL: http://arxiv.org/abs/1705.07874 (visited on 09/14/2021).

[42] Scott M. Lundberg, Gabriel G. Erion, and Su-In Lee. "Consistent Individualized Feature Attribution for Tree Ensembles". In: *arXiv:1802.03888 [cs, stat]* (Mar. 2019). arXiv: 1802.03888. URL: http://arxiv.org/abs/1802.03888 (visited on 09/20/2021).

[43] Xing Luo, Dongxiao Zhang, and Xu Zhu. “Deep learning based forecasting of photovoltaic power generation by incorporating domain knowledge”. en. In: *Energy* 225 (June 2021), p. 120240. ISSN: 03605442. DOI: 10.1016/j.energy.2021.120240. URL: https://linkinghub.elsevier.com/retrieve/pii/S0360544221004898 (visited on 10/20/2021).

[44] Gary C. McDonald. “Ridge regression”. en. In: *Wiley Interdisciplinary Reviews: Computational Statistics* 1.1 (July 2009), pp. 93–100. ISSN: 1939-5108, 1939-0068. DOI: 10.1002/wics.14. URL: https://onlinelibrary.wiley.com/doi/10.1002/wics.14 (visited on 03/18/2022).

[45] Jeremy Miles. “Tolerance and Variance Inflation Factor”. en. In: *Wiley StatsRef: Statistics Reference Online*. Ed. by N. Balakrishnan et al. 1st ed. Wiley, Sept. 2014. ISBN: 978-1-118-44511-2. DOI: 10.1002/9781118445112.stat06593. URL: https://onlinelibrary.wiley.com/doi/10.1002/9781118445112.stat06593 (visited on 05/24/2022).

[46] Brent Mittelstadt, Chris Russell, and Sandra Wachter. “Explaining Explanations in AI”. In: *Proceedings of the Conference on Fairness, Accountability, and Transparency* (Jan. 2019). arXiv: 1811.01439, pp. 279–288. DOI: 10.1145/3287560.3287574. URL: http://arxiv.org/abs/1811.01439 (visited on 02/18/2022).

[47] Grégoire Montavon et al. “Layer-Wise Relevance Propagation: An Overview”. en. In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Ed. by Wojciech Samek et al. Vol. 11700. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 193–209. ISBN: 978-3-030-28953-9 978-3-030-28954-6. DOI: 10.1007/978-3-030-28954-6_10. URL: http://link.springer.com/10.1007/978-3-030-28954-6_10 (visited on 10/15/2021).

[48] Leonardo de Moura and Nikolaj Bjørner. “Z3: An Efficient SMT Solver”. In: *Tools and Algorithms for the Construction and Analysis of Systems*. Ed. by David Hutchison et al. Vol. 4963. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 337–340. ISBN: 978-3-540-78799-0 978-3-540-78800-3. DOI: 10.1007/978-3-540-78800-3_24. URL: http://link.springer.com/10.1007/978-3-540-78800-3_24 (visited on 03/29/2022).

[49] Ozan Ozyegen, Igor Ilic, and Mucahit Cevik. “Evaluation of Local Explanation Methods for Multivariate Time Series Forecasting”. In: *arXiv:2009.09092 [cs, stat]* (Sept. 2020). arXiv: 2009.09092. URL: http://arxiv.org/abs/2009.09092 (visited on 09/20/2021).

[50] Leonardos Pantiskas, Kees Verstoep, and Henri Bal. “Interpretable Multivariate Time Series Forecasting with Temporal Attention Convolutional Neural Networks”. In: *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. Canberra, ACT, Australia: IEEE, Dec. 2020, pp. 1687–1694. ISBN: 978-1-72812-547-3. DOI: 10.1109/SSCI47803.2020.9308570. URL: https://ieeexplore.ieee.org/document/9308570/ (visited on 09/21/2021).

[51] Axel Parmentier and Thibaut Vidal. “Optimal Counterfactual Explanations in Tree Ensembles”. In: *arXiv:2106.06631 [cs, math]* (June 2021). arXiv: 2106.06631. URL: http://arxiv.org/abs/2106.06631 (visited on 01/31/2022).

[52] Quingyi, Pan, Wenbo, Hun, and Jun, Zhu. “Series Saliency: Temporal Interpretation for Multivariate Time Series Forecasting”. In: (Dec. 2020). URL: https://arxiv.org/pdf/2012.09324.pdf.

[53] M. Raissi, P. Perdikaris, and G.E. Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. en. In: *Journal of Computational Physics* 378 (Feb. 2019), pp. 686–707. ISSN: 00219991. DOI: 10.1016/j.jcp.2018.10.045. URL: https://linkinghub.elsevier.com/retrieve/pii/S0021999118307125 (visited on 10/12/2021).

[54] Lei Ren et al. “A Data-Driven Approach of Product Quality Prediction for Complex Production Systems”. In: *IEEE Transactions on Industrial Informatics* 17.9 (Sept. 2021), pp. 6457–6465. ISSN: 1551-3203, 1941-0050. DOI: 10.1109/TII.2020.3001054. URL: https://ieeexplore.ieee.org/document/9112632/ (visited on 03/15/2022).

[55] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “”Why Should I Trust You?”: Explaining the Predictions of Any Classifier”. In: *arXiv:1602.04938 [cs, stat]* (Aug. 2016). arXiv: 1602.04938. URL: http://arxiv.org/abs/1602.04938 (visited on 09/14/2021).

[56] Laura Rieger et al. “Interpretations are useful: penalizing explanations to align neural networks with prior knowledge”. In: *arXiv:1909.13584 [cs, stat]* (Oct. 2020). arXiv: 1909.13584. URL: http://arxiv.org/abs/1909.13584 (visited on 10/25/2021).

[57] Thomas Rojat et al. “Explainable Artificial Intelligence (XAI) on TimeSeries Data: A Survey”. In: *arXiv:2104.00950 [cs]* (Apr. 2021). arXiv: 2104.00950. URL: http://arxiv.org/abs/2104.00950 (visited on 09/10/2021).

[58] Laura von Rueden et al. “Informed Machine Learning – A Taxonomy and Survey of Integrating Knowledge into Learning Systems”. In: *IEEE Transactions on Knowledge and Data Engineering* (2021). arXiv: 1903.12394, pp. 1–1. ISSN: 1041-4347, 1558-2191, 2326-3865. DOI: 10.1109/TKDE.2021.3079836. URL: http://arxiv.org/abs/1903.12394 (visited on 10/12/2021).

[59] Udo Schlegel et al. “Towards a Rigorous Evaluation of XAI Methods on Time Series”. In: *arXiv:1909.07082 [cs]* (Sept. 2019). arXiv: 1909.07082. URL: http://arxiv.org/abs/1909.07082 (visited on 09/15/2021).

[60] Cedric Schockaert, Reinhard Leperlier, and Assaad Moawad. “Attention Mechanism for Multivariate Time Series Recurrent Model Interpretability Applied to the Ironmaking Industry”. In: *arXiv:2007.12617 [cs]* (July 2020). arXiv: 2007.12617. URL: http://arxiv.org/abs/2007.12617 (visited on 09/21/2021).

[61] Skipper Seabold and Josef Perktold. “statsmodels: Econometric and statistical modeling with python”. In: *9th Python in Science Conference*. 2010.

[62] Rajat Sen, Hsiang-Fu Yu, and Inderjit Dhillon. “Think Globally, Act Locally: A Deep Neural Network Approach to High-Dimensional Time Series Forecasting”. In: *arXiv:1905.03806 [cs, stat]* (Oct. 2019). arXiv: 1905.03806. URL: http://arxiv.org/abs/1905.03806 (visited on 10/18/2021).

[63] Julian Senoner, Torbjørn Netland, and Stefan Feuerriegel. “Using Explainable Artificial Intelligence to Improve Process Quality: Evidence from Semiconductor Manufacturing”. en. In: *Management Science* (Dec. 2021), mnsc.2021.4190. ISSN: 0025-1909, 1526-5501. DOI: 10.1287/mnsc.2021.4190. URL: http://pubsonline.informs.org/doi/10.1287/mnsc.2021.4190 (visited on 01/31/2022).

[64] Oscar Serradilla et al. “Interpreting Remaining Useful Life estimations combining Explainable Artificial Intelligence and domain knowledge in industrial machinery”. In: *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. Glasgow, United Kingdom: IEEE, July 2020, pp. 1–8. ISBN: 978-1-72816-932-3. DOI: 10.1109/FUZZ48607.2020.9177537. URL: https://ieeexplore.ieee.org/document/9177537/ (visited on 04/22/2022).

[65] Thomas Spooner et al. “Counterfactual Explanations for Arbitrary Regression Models”. In: *arXiv:2106.15212 [cs]* (June 2021). arXiv: 2106.15212. URL: http://arxiv.org/abs/2106.15212 (visited on 09/10/2021).

[66] David Steen. “Carbonated beverages”. en. In: *Chemistry and Technology of Soft Drinks and Fruit Juices*. Ed. by Philip R. Ashurst. Chichester, UK: John Wiley & Sons, Ltd, Jan. 2016, pp. 146–173. ISBN: 978-1-118-63494-3 978-1-4443-3381-7. DOI: 10.1002/9781118634943.ch7. URL: https://sci-hub.se/https://dx.doi.org/10.1002/9781118634943.ch7 (visited on 09/13/2021).

[67] Clifton D. Sutton. “Classification and Regression Trees, Bagging, and Boosting”. en. In: *Handbook of Statistics*. Vol. 24. Elsevier, 2005, pp. 303–329. ISBN: 978-0-444-51141-6. DOI: 10.1016/S0169-7161(04)24011-1. URL: https://linkinghub.elsevier.com/retrieve/pii/S0169716104240111 (visited on 02/23/2022).

[68] D. Wang, Jun Liu, and R. Srinivasan. “Data-Driven Soft Sensor Approach for Quality Prediction in a Refining Process”. In: *IEEE Transactions on Industrial Informatics* 6.1 (Feb. 2010), pp. 11–17. ISSN: 1551-3203, 1941-0050. DOI: 10.1109/TII.2009.2025124. URL: http://ieeexplore.ieee.org/document/5184939/ (visited on 03/14/2022).

[69] Dorina Weichert et al. “A review of machine learning for the optimization of production processes”. en. In: *The International Journal of Advanced Manufacturing Technology* 104.5-8 (Oct. 2019), pp. 1889–1902. ISSN: 0268-3768, 1433-3015. DOI: 10.1007/s00170-019-03988-5. URL: http://link.springer.com/10.1007/s00170-019-03988-5 (visited on 09/23/2021).

[70] Eyal Winter. "Chapter 53 The shapley value". en. In: *Handbook of Game Theory with Economic Applications*. Vol. 3. Elsevier, 2002, pp. 2025–2054. ISBN: 978-0-444-89428-1. DOI: 10.1016/S1574-0005(02)03016-3. URL: https://linkinghub.elsevier.com/retrieve/pii/S1574000502030163 (visited on 09/14/2021).

[71] Jim Woodward. "Explanation, Invariance, and Intervention". en. In: *Philosophy of Science* 64 (Dec. 1997), S26–S41. ISSN: 0031-8248, 1539-767X. DOI: 10.1086/392584. URL: https://www.journals.uchicago.edu/doi/10.1086/392584 (visited on 02/18/2022).

[72] Haiyue Wu, Aihua Huang, and John W. Sutherland. *Layer-Wise Relevance Propagation for Interpreting LSTM-RNN Decisions in Predictive Maintenance*. preprint. In Review, May 2021. DOI: 10.21203/rs.3.rs-522677/v1. URL: https://www.researchsquare.com/article/rs-522677/v1 (visited on 10/15/2021).

[73] Zhaohua Wu et al. "On the trend, detrending, and variability of nonlinear and nonstationary time series". en. In: *Proceedings of the National Academy of Sciences* 104.38 (Sept. 2007), pp. 14889–14894. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.0701020104. URL: https://pnas.org/doi/full/10.1073/pnas.0701020104 (visited on 05/10/2022).

[74] Fan Yang, Mengnan Du, and Xia Hu. "Evaluating Explanation Without Ground Truth in Interpretable Machine Learning". In: *arXiv:1907.06831 [cs, stat]* (Aug. 2019). arXiv: 1907.06831. URL: http://arxiv.org/abs/1907.06831 (visited on 11/04/2021).

[75] Le Yao and Zhiqiang Ge. "Deep Learning of Semisupervised Process Data With Hierarchical Extreme Learning Machine and Soft Sensor Application". In: *IEEE Transactions on Industrial Electronics* 65.2 (Feb. 2018), pp. 1490–1498. ISSN: 0278-0046, 1557-9948. DOI: 10.1109/TIE.2017.2733448. URL: http://ieeexplore.ieee.org/document/8002611/ (visited on 03/18/2022).

[76] Shen Yin et al. "A Review on Basic Data-Driven Approaches for Industrial Process Monitoring". In: *IEEE Transactions on Industrial Electronics* 61.11 (Nov. 2014), pp. 6418–6428. ISSN: 0278-0046, 1557-9948. DOI: 10.1109/TIE.2014.2301773. URL: http://ieeexplore.ieee.org/document/6717991/ (visited on 03/15/2022).

[77] Xue Ying. "An Overview of Overfitting and its Solutions". In: *Journal of Physics: Conference Series* 1168 (Feb. 2019), p. 022022. ISSN: 1742-6588, 1742-6596. DOI: 10.1088/1742-6596/1168/2/022022. URL: https://iopscience.iop.org/article/10.1088/1742-6596/1168/2/022022 (visited on 06/06/2022).

[78] Seungil You et al. "Deep Lattice Networks and Partial Monotonic Functions". In: *arXiv:1709.06680 [cs, stat]* (Sept. 2017). arXiv: 1709.06680. URL: http://arxiv.org/abs/1709.06680 (visited on 10/12/2021).

A
Data Plots

These plots represent the data taken from a single batch.

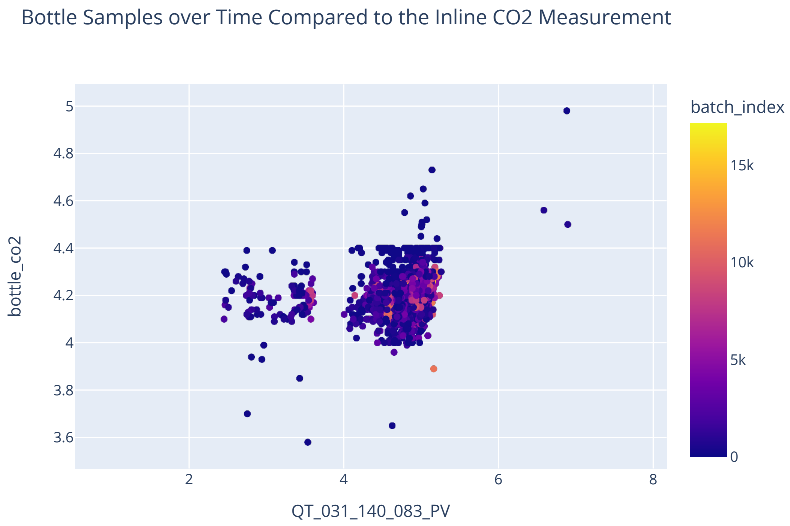


Figure A.1: The four bottle samples for the selected batch.

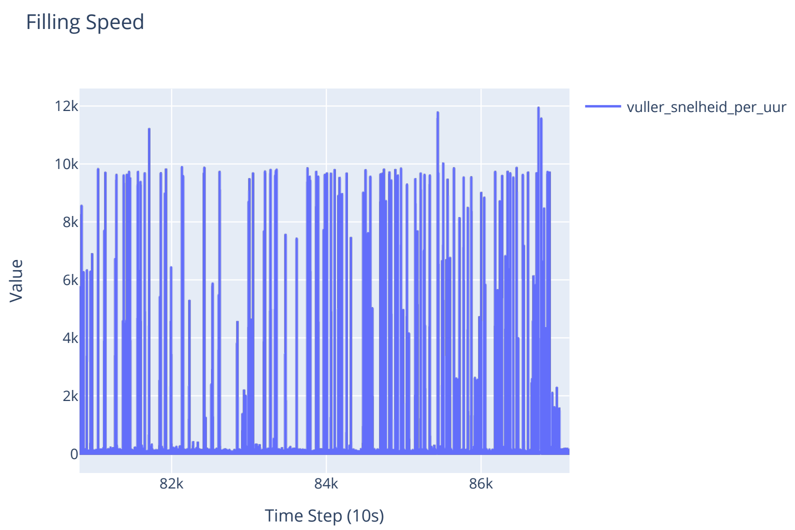


Figure A.2: The amount of filling done by the bottle filler during the selected batch. The values fluctuate greatly, indicating a need for an aggregation variable.

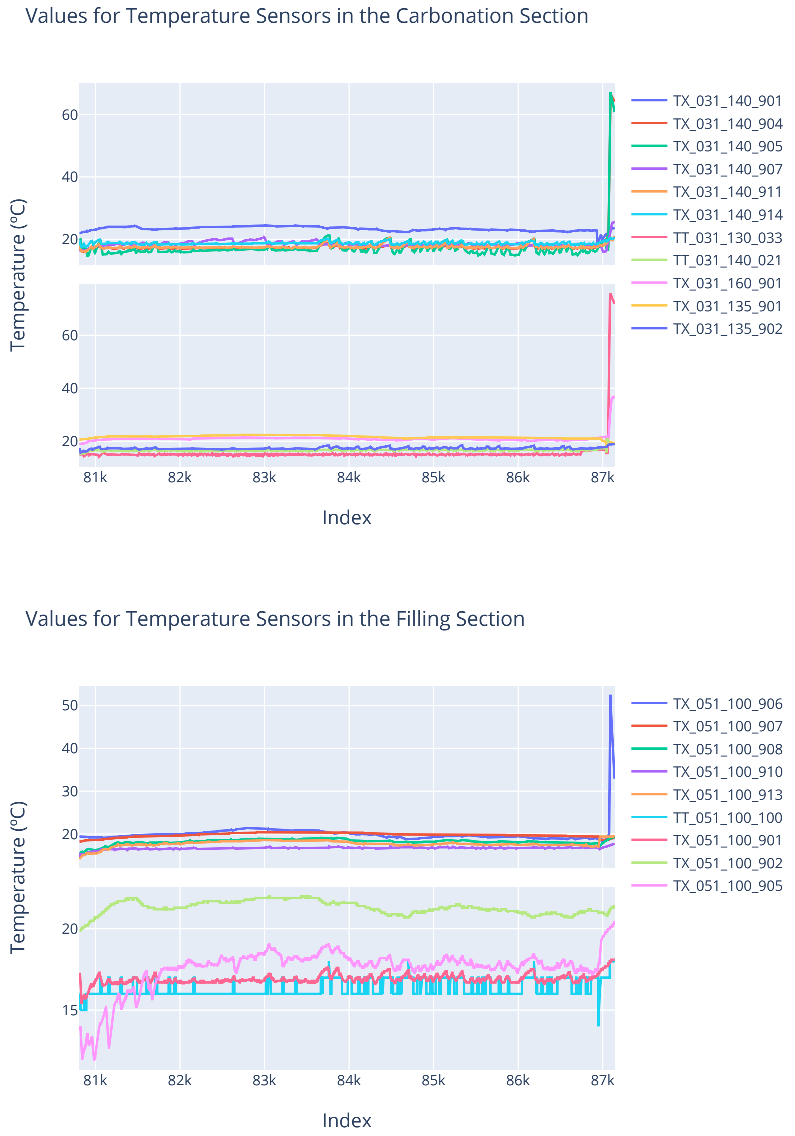


Figure A.3: Values for the pressure sensors in the carbonation (upper) and filing (lower) sections.

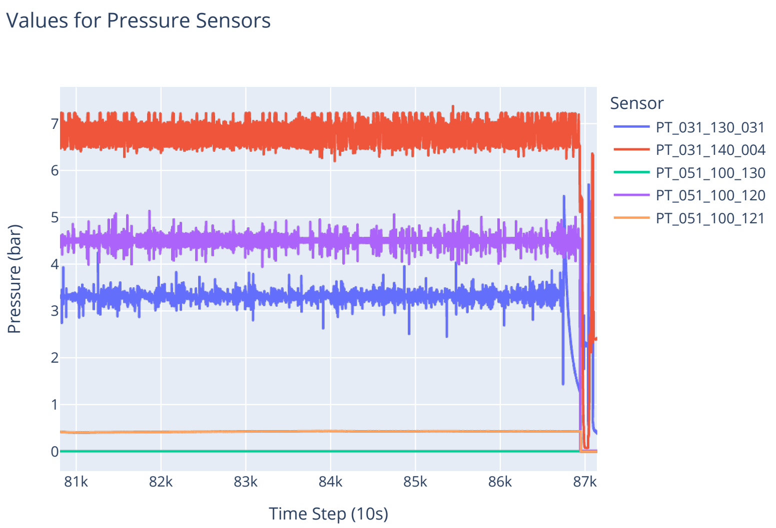
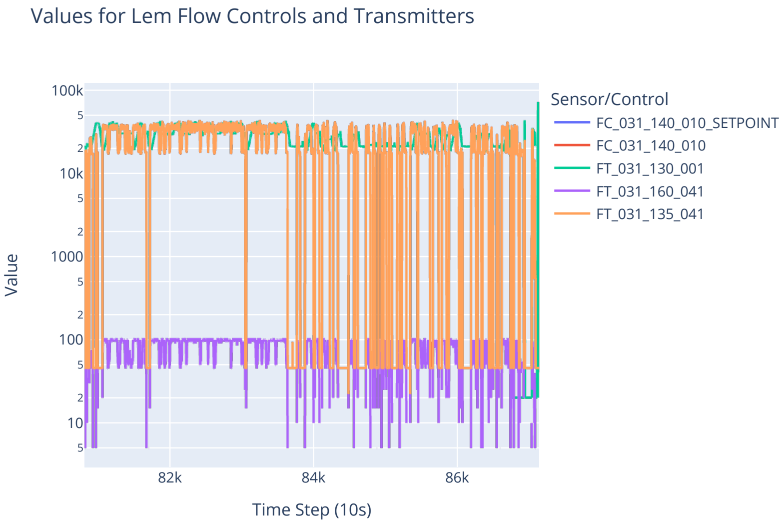


Figure A.4: Values for the pressure sensors in the process.



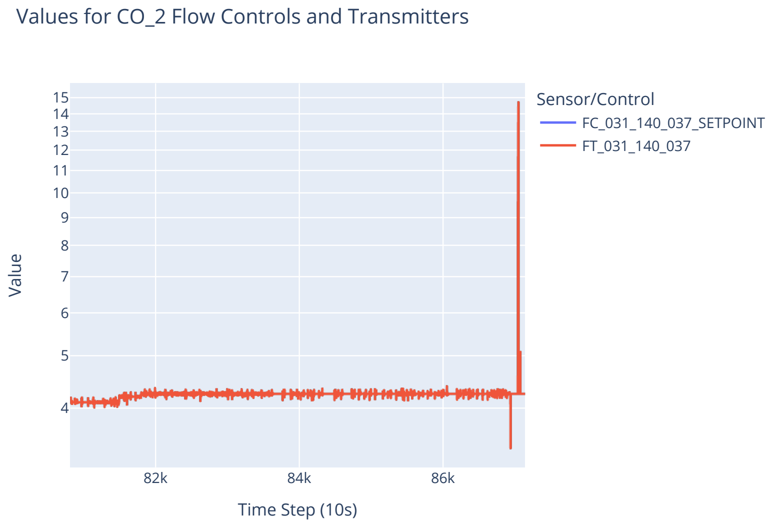


Figure A.5: Values for the flow sensors for lemonade flow and CO_2 flow.

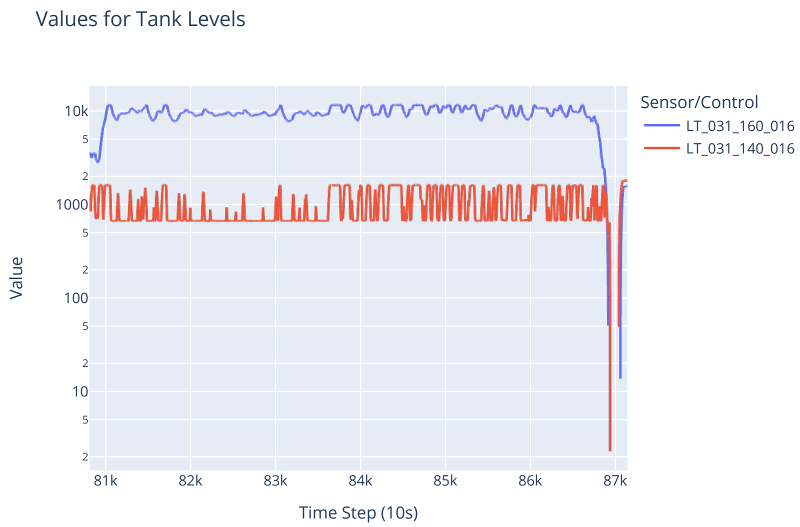


Figure A.6: Values for the two tanks in the process.

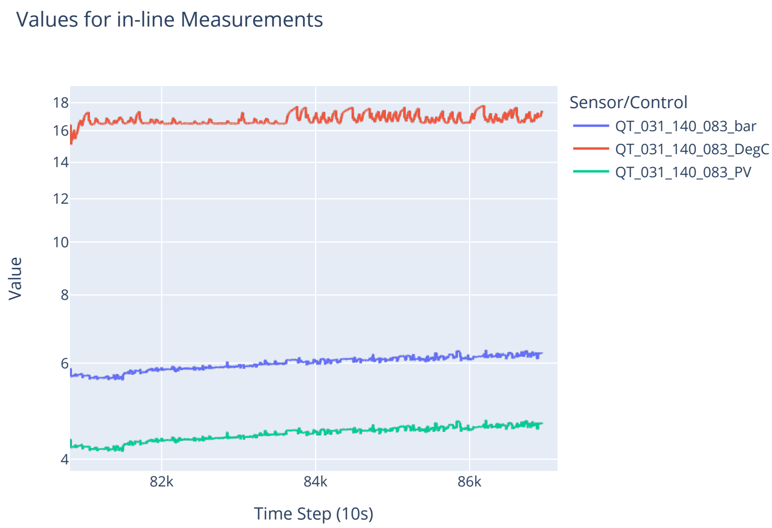


Figure A.7: Values for the inline CO_2, temperature, and pressure measurement.

B

Counterfactual Explanations

This chapter contains the counterfactual explanations and differences to the old process state for the results on both the artificial and real world dataset.

B.1. Artificial Data

Table B.1: The difference between the old and suggested process state. Cost indicates how long it takes for the changes to take effect.

	P_cur	P_alt	Diff	Cost
bottle_co2 (pred)	3.984609	3.070194	-0.914415	7.000000
inc_lem_T	14	14.000000	0.000000	23.000000
inc_lem_p	4.500000	4.500000	0.000000	23.000000
inc_co2_gl	4.216527	3.836640	-0.379887	23.000000
tank_T	18.257002	18.263106	0.006103	0.000000
tank_p	4.876343	4.882446	0.006103	0.000000
tank_co2_gl	4.200494	4.206598	0.006103	nan
filler_T	21.361955	25.878398	4.516443	7.000000
filler_p	7.065151	7.071254	0.006103	nan
filler_co2_gl	4.198920	4.205024	0.006103	nan
filling	0.850000	0.157122	-0.692878	23.000000
last_bottle_co2	3.009920	3.009920	-0.000000	23.000000
room_T	28.467281	28.473385	0.006103	23.000000
filling_50_sum	7.850000	7.856103	0.006103	23.000000

Table B.2: The difference between the old and suggested process state. Cost indicates how long it takes for the changes to take effect.

	P_cur	P_alt	Diff	Cost
bottle_co2 (pred)	2.767138	4.191789	1.424650	nan
inc_lem_T	14	14.000000	0.000000	49.000000
inc_lem_p	4.500000	4.500000	0.000000	49.000000
inc_co2_gl	4.121436	4.092623	-0.028813	49.000000
tank_T	18.275703	18.275707	0.000004	0.000000
tank_p	4.887308	4.887308	-0.000000	0.000000
tank_co2_gl	4.092079	4.140775	0.048696	nan
filler_T	27.103899	19.100785	-8.003113	nan
filler_p	8.791627	8.791631	0.000004	nan
filler_co2_gl	4.091811	4.038814	-0.052997	nan
filling	0.000000	0.500000	0.500000	49.000000
last_bottle_co2	4.327898	4.327898	-0.000000	49.000000
room_T	28.092729	27.781183	-0.311546	49.000000
filling_50_sum	23.000000	25.000000	2.000000	49.000000

Table B.3: The difference between the old and suggested process state. Cost indicates how long it takes for the changes to take effect.

	P_cur	P_alt	Diff	Cost
bottle_co2 (pred)	4.180580	4.161622	-0.018958	0.000000
inc_lem_T	14	14.000000	0.000000	25.000000
inc_lem_p	4.500000	4.500000	0.000000	25.000000
inc_co2_gl	4.100047	4.079961	-0.020085	25.000000
tank_T	16.193222	16.193415	0.000193	0.000000
tank_p	4.319735	4.319928	0.000193	0.000000
tank_co2_gl	4.051118	4.051311	0.000193	4.000000
filler_T	18.941511	19.148892	0.207380	0.000000
filler_p	5.512781	5.512974	0.000193	0.000000
filler_co2_gl	4.050836	4.051669	0.000833	0.000000
filling	0.900000	0.500000	-0.400000	25.000000
last_bottle_co2	4.296629	4.296629	-0.000000	25.000000
room_T	29.503941	29.504134	0.000193	25.000000
filling_50_sum	15.700000	25.000000	9.300000	25.000000

Table B.4: The difference between the old and suggested process state. Cost indicates how long it takes for the changes to take effect.

	P_cur	P_alt	Diff	Cost
bottle_co2 (pred)	3.360744	4.068923	0.708179	nan
inc_lem_T	14	14.000000	0.000000	48.000000
inc_lem_p	4.500000	4.500000	0.000000	48.000000
inc_co2_gl	3.919997	3.575490	-0.344507	48.000000
tank_T	19.597759	19.603087	0.005328	0.000000
tank_p	5.204486	5.209814	0.005328	nan
tank_co2_gl	3.959561	3.964889	0.005328	nan
filler_T	23.775057	18.275818	-5.499239	nan
filler_p	8.448919	8.454247	0.005328	0.000000
filler_co2_gl	3.962494	3.907715	-0.054779	4.000000
filling	0.950000	0.035106	-0.914893	48.000000
last_bottle_co2	3.418333	3.418333	-0.000000	48.000000
room_T	27.287861	27.293189	0.005328	48.000000
filling_50_sum	1.750000	1.755328	0.005328	48.000000

Table B.5: The difference between the old and suggested process state. Cost indicates how long it takes for the changes to take effect.

	P_cur	P_alt	Diff	Cost
bottle_co2 (pred)	4.063641	4.067449	0.003808	0.000000
inc_lem_T	14	14.000000	0.000000	0.000000
inc_lem_p	4.500000	4.500000	0.000000	0.000000
inc_co2_gl	3.879086	3.570274	-0.308812	0.000000
tank_T	15.162072	15.162184	0.000112	0.000000
tank_p	4.028523	4.028635	0.000112	1.000000
tank_co2_gl	3.859511	3.894394	0.034883	nan
filler_T	17.555392	18.227176	0.671784	nan
filler_p	4.719361	4.719473	0.000112	0.000000
filler_co2_gl	3.858517	3.915612	0.057095	nan
filling	0.800000	0.895002	0.095002	0.000000
last_bottle_co2	3.249847	3.249847	-0.000000	0.000000
room_T	27.914868	27.914980	0.000112	0.000000
filling_50_sum	44.750000	44.750112	0.000112	0.000000

B.2. Real Data

Table B.6: The difference between the old and suggested process state. Cost indicates how long it takes for the changes to take effect.

	P_cur	P_alt	Diff	Cost
bottle_co2 (pred)	4.232639	3.962990	-0.269649	nan
FC_031_140_010	91.666664	27362.500099	27270.833435	49.000000
FT_031_130_001	39739.582000	20821.758888	-18917.823112	49.000000
FT_031_140_037	4.550000	3.424037	-1.125963	49.000000
FT_031_160_041	0.000000	75.000000	75.000000	49.000000
FT_031_135_041	91.666664	39794.789161	39703.122497	49.000000
QT_031_140_083_bar	6.035161	6.035161	-0.000000	4.000000
QT_031_140_083_DegC	16.491929	16.492027	0.000099	0.000000
QT_031_140_083_PV	4.858758	4.816518	-0.042240	4.000000
PT_031_130_031	3.326389	3.326488	0.000099	2.000000
PT_031_140_004	6.774884	6.774983	0.000099	1.000000
PT_051_100_130	7.202071	7.202170	0.000099	0.000000
PT_051_100_120	5.296347	5.296446	0.000099	0.000000
PT_051_100_121	0.980582	0.980681	0.000099	0.000000
vuller_snelheid_per_uur	51.386000	1645.434208	1594.048208	49.000000
LT_031_160_016	6200.293500	6470.061866	269.768366	nan
LT_031_140_016	1602.430500	1599.999901	-2.430599	nan
TT_031_130_033	14.062500	14.062500	0.000000	49.000000
TT_031_140_021	15.885416	15.885515	0.000099	0.000000
TT_051_100_100	16.000000	16.000099	0.000099	0.000000
TX_031_160_901	19.000000	24.450099	5.450099	nan
TX_031_135_901	20.600000	20.850098	0.250098	0.000000
TX_031_135_902	17.000000	17.000099	0.000099	0.000000
TX_031_140_901	22.600000	22.950099	0.350099	4.000000
TX_031_140_904	17.200000	17.200099	0.000099	0.000000
TX_031_140_905	17.500000	24.850098	7.350098	nan
TX_031_140_907	21.600000	21.600099	0.000099	0.000000
TX_031_140_911	16.800000	16.800099	0.000099	0.000000
TX_031_140_914	18.300000	18.313156	0.013156	0.000000
TX_051_100_901	16.300000	16.300099	0.000099	0.000000
TX_051_100_902	20.400000	29.350099	8.950099	nan
TX_051_100_905	17.100000	18.400098	1.300098	20.000000
TX_051_100_906	21.000000	21.000099	0.000099	0.000000
TX_051_100_907	18.100000	18.100099	0.000099	0.000000
TX_051_100_908	17.200000	18.650099	1.450099	nan
TX_051_100_910	15.500000	15.950099	0.450099	0.000000
TX_051_100_913	17.300000	25.050099	7.750099	nan
vuller_20m_sum	13439.190150	89552.832031	76113.641881	49.000000
FT_031_140_037_20m_sum	546.078328	489.857254	-56.221074	49.000000

Table B.7: The difference between the old and suggested process state. Cost indicates how long it takes for the changes to take effect.

	P_cur	P_alt	Diff	Cost
bottle_co2 (pred)	4.183358	4.165700	-0.017658	0.000000
FC_031_140_010	36666.668000	0.000349	-36666.667651	49.000000
FT_031_130_001	29956.598000	21004.051758	-8952.546242	49.000000
FT_031_140_037	4.739412	4.910928	0.171516	49.000000
FT_031_160_041	95.000000	97.500349	2.500349	49.000000
FT_031_135_041	36666.668000	37778.125349	1111.457349	49.000000
QT_031_140_083_bar	5.931780	5.931780	-0.000000	nan
QT_031_140_083_DegC	16.367477	16.367826	0.000349	0.000000
QT_031_140_083_PV	4.815911	4.857508	0.041598	0.000000
PT_031_130_031	3.338541	3.338541	0.000000	1.000000
PT_031_140_004	6.526042	6.526391	0.000349	nan
PT_051_100_130	0.000000	0.000349	0.000349	4.000000
PT_051_100_120	4.780526	4.780875	0.000349	0.000000
PT_051_100_121	0.988828	0.988828	-0.000000	2.000000
vuller_snelheid_per_uur	0.000000	81.894999	81.894999	49.000000
LT_031_160_016	10194.520500	10204.520849	10.000349	nan
LT_031_140_016	669.560200	671.560549	2.000349	2.000000
TT_031_130_033	14.800347	14.800696	0.000349	49.000000
TT_031_140_021	16.102430	16.102779	0.000349	0.000000
TT_051_100_100	16.000000	16.500349	0.500349	0.000000
TX_031_160_901	20.200000	20.200000	0.000000	0.000000
TX_031_135_901	21.200000	21.200349	0.000349	0.000000
TX_031_135_902	16.700000	16.700349	0.000349	0.000000
TX_031_140_901	22.400000	22.400349	0.000349	0.000000
TX_031_140_904	17.000000	17.000349	0.000349	0.000000
TX_031_140_905	15.000000	15.000349	0.000349	0.000000
TX_031_140_907	22.200000	22.200349	0.000349	0.000000
TX_031_140_911	17.000000	17.250349	0.250349	0.000000
TX_031_140_914	17.900000	17.900349	0.000349	0.000000
TX_051_100_901	16.500000	16.500349	0.000349	0.000000
TX_051_100_902	20.300000	20.300349	0.000349	0.000000
TX_051_100_905	14.600000	14.600349	0.000349	0.000000
TX_051_100_906	19.200000	19.200000	0.000000	0.000000
TX_051_100_907	18.800000	23.350349	4.550349	nan
TX_051_100_908	17.300000	19.850349	2.550349	nan
TX_051_100_910	16.100000	16.100349	0.000349	0.000000
TX_051_100_913	16.900000	17.750349	0.850349	nan
vuller_20m_sum	47277.600120	30421.255364	-16856.344756	49.000000
FT_031_140_037_20m_sum	571.151996	578.698396	7.546401	49.000000

Table B.8: The difference between the old and suggested process state. Cost indicates how long it takes for the changes to take effect.

	P_cur	P_alt	Diff	Cost
bottle_co2 (pred)	4.217660	4.212158	-0.005502	0.000000
FC_031_140_010	17164.582000	57.292983	-17107.289017	49.000000
FT_031_130_001	39638.312000	20912.906592	-18725.405408	49.000000
FT_031_140_037	4.800278	5.157154	0.356876	49.000000
FT_031_160_041	45.000000	75.000000	30.000000	49.000000
FT_031_135_041	17164.582000	37778.126318	20613.544318	49.000000
QT_031_140_083_bar	6.033312	6.033312	-0.000000	nan
QT_031_140_083_DegC	16.784092	16.861022	0.076930	0.000000
QT_031_140_083_PV	4.901725	4.936425	0.034699	nan
PT_031_130_031	3.296007	3.297325	0.001318	3.000000
PT_031_140_004	7.203125	7.204443	0.001318	15.000000
PT_051_100_130	0.000000	0.001318	0.001318	nan
PT_051_100_120	4.671137	4.843919	0.172782	nan
PT_051_100_121	1.004881	1.006199	0.001318	nan
vuller_snelheid_per_uur	0.000000	468.060644	468.060644	49.000000
LT_031_160_016	8387.879000	8387.879000	0.000000	nan
LT_031_140_016	1206.597200	1206.598518	0.001318	nan
TT_031_130_033	14.973958	14.975276	0.001318	49.000000
TT_031_140_021	16.796875	16.798193	0.001318	0.000000
TT_051_100_100	16.000000	17.001318	1.001318	5.000000
TX_031_160_901	21.100000	21.101318	0.001318	0.000000
TX_031_135_901	22.000000	23.251318	1.251318	15.000000
TX_031_135_902	17.400000	17.401318	0.001318	0.000000
TX_031_140_901	24.000000	24.001318	0.001318	0.000000
TX_031_140_904	17.700000	17.650000	-0.050000	0.000000
TX_031_140_905	16.200000	16.200000	0.000000	0.000000
TX_031_140_907	23.700000	23.751318	0.051318	0.000000
TX_031_140_911	17.500000	17.951318	0.451318	0.000000
TX_031_140_914	18.900000	18.900000	0.000000	0.000000
TX_051_100_901	17.000000	17.001318	0.001318	0.000000
TX_051_100_902	21.600000	21.601318	0.001318	0.000000
TX_051_100_905	18.900000	18.901318	0.001318	0.000000
TX_051_100_906	19.600000	19.601318	0.001318	0.000000
TX_051_100_907	20.300000	20.951318	0.651318	nan
TX_051_100_908	19.000000	22.651319	3.651319	nan
TX_051_100_910	16.800000	17.651319	0.851319	nan
TX_051_100_913	18.600000	19.351318	0.751318	nan
vuller_20m_sum	74114.562098	66946.391943	-7168.170155	49.000000
FT_031_140_037_20m_sum	575.984618	593.879486	17.894868	49.000000

Table B.9: The difference between the old and suggested process state. Cost indicates how long it takes for the changes to take effect.

	P_cur	P_alt	Diff	Cost
bottle_co2 (pred)	4.177146	4.218222	0.041076	0.000000
FC_031_140_010	34397.918000	0.000048	-34397.917952	49.000000
FT_031_130_001	39780.090000	26746.239258	-13033.850742	49.000000
FT_031_140_037	4.552155	0.198381	-4.353774	49.000000
FT_031_160_041	100.000000	82.500048	-17.499952	49.000000
FT_031_135_041	34397.918000	68.749998	-34329.168002	49.000000
QT_031_140_083_bar	5.833068	6.082417	0.249349	25.000000
QT_031_140_083_DegC	16.477550	17.214059	0.736509	5.000000
QT_031_140_083_PV	4.658160	4.658208	0.000048	30.000000
PT_031_130_031	3.344618	3.344618	0.000000	2.000000
PT_031_140_004	6.523148	6.523196	0.000048	4.000000
PT_051_100_130	6.707789	6.707837	0.000048	nan
PT_051_100_120	5.637896	5.637944	0.000048	nan
PT_051_100_121	0.999029	0.999077	0.000048	0.000000
vuller_snelheid_per_uur	0.000000	0.000048	0.000048	49.000000
LT_031_160_016	5374.400000	6000.000000	625.600000	20.000000
LT_031_140_016	662.037000	662.037000	0.000000	1.000000
TT_031_130_033	14.496528	14.605083	0.108555	49.000000
TT_031_140_021	15.842013	15.972270	0.130257	0.000000
TT_051_100_100	16.000000	16.000048	0.000048	0.000000
TX_031_160_901	20.800000	20.800048	0.000048	0.000000
TX_031_135_901	22.900000	22.900048	0.000048	0.000000
TX_031_135_902	17.000000	17.000048	0.000048	0.000000
TX_031_140_901	25.000000	25.005394	0.005394	nan
TX_031_140_904	17.000000	17.000048	0.000048	0.000000
TX_031_140_905	18.100000	18.100048	0.000048	25.000000
TX_031_140_907	24.400000	24.400048	0.000048	0.000000
TX_031_140_911	17.800000	17.800048	0.000048	0.000000
TX_031_140_914	18.900000	18.900048	0.000048	0.000000
TX_051_100_901	16.700000	16.700000	0.000000	0.000000
TX_051_100_902	21.800000	21.800048	0.000048	0.000000
TX_051_100_905	19.200000	19.200048	0.000048	0.000000
TX_051_100_906	22.200000	22.200048	0.000048	0.000000
TX_051_100_907	19.600000	19.600000	0.000000	0.000000
TX_051_100_908	18.700000	19.950048	1.250048	nan
TX_051_100_910	16.100000	16.100048	0.000048	0.000000
TX_051_100_913	18.400000	18.400048	0.000048	0.000000
vuller_20m_sum	37371.229400	27724.932305	-9646.297095	49.000000
FT_031_140_037_20m_sum	535.082791	325.111679	-209.971112	49.000000

Table B.10: The difference between the old and suggested process state. Cost indicates how long it takes for the changes to take effect.

	P_cur	P_alt	Diff	Cost
bottle_co2 (pred)	4.177800	4.231006	0.053206	0.000000
FC_031_140_010	22664.584000	26823.959008	4159.375008	49.000000
FT_031_130_001	29693.287000	38109.087891	8415.800891	49.000000
FT_031_140_037	4.805851	2.159022	-2.646829	49.000000
FT_031_160_041	60.000000	75.000023	15.000023	49.000000
FT_031_135_041	22710.418000	68.749998	-22641.668002	49.000000
QT_031_140_083_bar	6.071759	6.293732	0.221972	nan
QT_031_140_083_DegC	16.650813	16.650836	0.000023	0.000000
QT_031_140_083_PV	4.911516	5.126567	0.215051	nan
PT_031_130_031	3.320312	3.320336	0.000023	2.000000
PT_031_140_004	6.803819	6.803842	0.000023	39.000000
PT_051_100_130	0.000000	0.000023	0.000023	nan
PT_051_100_120	5.157947	5.157947	0.000000	nan
PT_051_100_121	0.991430	0.991430	0.000000	0.000000
vuller_snelheid_per_uur	0.000000	75.763351	75.763351	49.000000
LT_031_160_016	9825.967000	9821.967000	-4.000000	nan
LT_031_140_016	726.851870	726.851893	0.000023	1.000000
TT_031_130_033	14.800347	14.800370	0.000023	49.000000
TT_031_140_021	16.362848	16.362871	0.000023	0.000000
TT_051_100_100	16.000000	16.000023	0.000023	0.000000
TX_031_160_901	20.900000	20.900000	0.000000	0.000000
TX_031_135_901	24.300000	24.300023	0.000023	nan
TX_031_135_902	17.400000	17.400023	0.000023	0.000000
TX_031_140_901	25.400000	25.400023	0.000023	nan
TX_031_140_904	17.100000	17.103205	0.003205	0.000000
TX_031_140_905	18.400000	18.800023	0.400023	nan
TX_031_140_907	24.400000	24.400023	0.000023	0.000000
TX_031_140_911	17.500000	17.500023	0.000023	0.000000
TX_031_140_914	19.000000	19.000023	0.000023	0.000000
TX_051_100_901	16.800000	16.800023	0.000023	0.000000
TX_051_100_902	23.000000	23.000023	0.000023	0.000000
TX_051_100_905	22.000000	22.000023	0.000023	nan
TX_051_100_906	20.700000	23.600024	2.900024	nan
TX_051_100_907	21.700000	21.700023	0.000023	0.000000
TX_051_100_908	20.700000	22.650024	1.950024	nan
TX_051_100_910	16.800000	16.800023	0.000023	0.000000
TX_051_100_913	20.300000	20.300023	0.000023	0.000000
vuller_20m_sum	37965.283700	40083.147772	2117.864072	49.000000
FT_031_140_037_20m_sum	575.889639	443.923218	-131.966422	49.000000

Table B.11: The difference between the old and suggested process state. Cost indicates how long it takes for the changes to take effect.

	P_cur	P_alt	Diff	Cost
bottle_co2 (pred)	4.158316	3.934201	-0.224115	nan
FC_031_140_010	18218.750000	34650.000000	16431.250000	49.000000
FT_031_130_001	29592.014000	27252.604550	-2339.409450	49.000000
FT_031_140_037	4.514945	2.503692	-2.011252	49.000000
FT_031_160_041	55.000000	72.500058	17.500058	49.000000
FT_031_135_041	18218.750000	40596.873105	22378.123105	49.000000
QT_031_140_083_bar	5.680339	5.680339	-0.000000	nan
QT_031_140_083_DegC	16.474509	16.474509	-0.000000	0.000000
QT_031_140_083_PV	4.534034	4.534034	-0.000000	nan
PT_031_130_031	3.296007	3.296064	0.000058	0.000000
PT_031_140_004	7.234954	7.235012	0.000058	nan
PT_051_100_130	7.699259	7.699317	0.000058	15.000000
PT_051_100_120	4.832255	4.832313	0.000058	0.000000
PT_051_100_121	0.992734	0.992792	0.000058	0.000000
vuller_snelheid_per_uur	116.069000	376.228970	260.159970	49.000000
LT_031_160_016	11185.285000	11194.285058	9.000058	nan
LT_031_140_016	1160.301000	1160.301058	0.000058	nan
TT_031_130_033	14.322917	14.995718	0.672801	49.000000
TT_031_140_021	16.059029	17.534779	1.475750	nan
TT_051_100_100	16.000000	16.000058	0.000058	0.000000
TX_031_160_901	20.900000	20.900058	0.000058	0.000000
TX_031_135_901	24.200000	25.250058	1.050058	nan
TX_031_135_902	17.300000	17.300058	0.000058	0.000000
TX_031_140_901	26.400000	32.450057	6.050057	nan
TX_031_140_904	17.100000	17.100058	0.000058	0.000000
TX_031_140_905	19.500000	19.500058	0.000058	0.000000
TX_031_140_907	25.700000	25.700000	0.000000	0.000000
TX_031_140_911	17.700000	17.700058	0.000058	0.000000
TX_031_140_914	19.400000	19.400058	0.000058	0.000000
TX_051_100_901	16.800000	16.800058	0.000058	0.000000
TX_051_100_902	24.100000	24.100058	0.000058	0.000000
TX_051_100_905	19.700000	25.950059	6.250059	nan
TX_051_100_906	24.200000	24.200058	0.000058	0.000000
TX_051_100_907	21.900000	25.700058	3.800058	nan
TX_051_100_908	20.000000	24.100058	4.100058	nan
TX_051_100_910	16.200000	16.200058	0.000058	0.000000
TX_051_100_913	19.500000	25.450058	5.950058	nan
vuller_20m_sum	91349.910200	81322.460938	-10027.449262	49.000000
FT_031_140_037_20m_sum	540.027632	440.217712	-99.809920	49.000000