A STRATEGY BASED ON STATISTICAL MODELLING AND MULTI-OBJECTIVE OPTIMIZATION TO DESIGN A DISHWASHER CLEANING CYCLE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

KORKUT ANAPA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
SCIENTIFIC COMPUTING

DECEMBER 2023

Approval of the thesis:

A STRATEGY BASED ON STATISTICAL MODELLING AND MULTI-OBJECTIVE OPTIMIZATION TO DESIGN A DISHWASHER CLEANING CYCLE

submitted by **KORKUT ANAPA** in partial fulfillment of the requirements for the degree of **Master of Science in Scientific Computing Department, Middle East Technical University** by,

Prof. Dr. A. Sevtap Selçuk Kestel _____
Dean, Graduate School of **Applied Mathematics**

Assoc. Prof. Dr. Önder Türk _____
Head of Department, **Scientific Computing**

Assoc. Prof. Dr. Hamdullah Yücel _____
Supervisor, **Scientific Computing, METU**

Dr. Songül Bayraktar _____
Co-supervisor, **Arçelik A.Ş.**

Examining Committee Members:

Prof. Dr. Sinan Kalkan
Computer Engineering, METU _____

Assoc. Prof. Dr. Hamdullah Yücel
Scientific Computing, METU _____

Assoc. Prof. Dr. Furkan Başer
Actuarial Sciences, Ankara University _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: KORKUT ANAPA

Signature :

ABSTRACT

A STRATEGY BASED ON STATISTICAL MODELLING AND MULTI-OBJECTIVE OPTIMIZATION TO DESIGN A DISHWASHER CLEANING CYCLE

ANAPA, KORKUT
M.S., Department of Scientific Computing
Supervisor : Assoc. Prof. Dr. Hamdullah Yücel
Co-Supervisor : Dr. Songül Bayraktar

December 2023, 84 pages

This thesis proposes a novel approach based on statistical learning and multi-objective optimization to reduce the need for experiments during the design phase of new cleaning cycles for household dishwashers. First, regression models are built that are associated with the feature selection methods to predict the outputs of a dishwasher cleaning cycle by using the existing cleaning cycles' program flows as input data and the results of the performance laboratory tests of the related cleaning cycles as output data. Then, a multi-objective optimization problem is defined by assigning the regression models and chosen features as objective functions and unknown decision variables, respectively. The obtained optimization problem is then solved using evolutionary algorithms according to the designer's preferences (or customers' needs).

Keywords: Dishwasher Design, Feature Selection, Multi-Objective Optimization, Evolutionary Algorithms, Statistical Modelling

ÖZ

BULAŞIK MAKİNESİ TEMİZLEME DÖNGÜSÜ TASARLAMAK İÇİN İSTATİSTİKSEL MODELLEMEYE VE ÇOK AMAÇLI OPTİMİZASYONA DAYALI BİR STRATEJİ

ANAPA, KORKUT
Yüksek Lisans, Bilimsel Hesaplama Bölümü
Tez Yöneticisi : Doç. Dr. Hamdullah Yücel
Ortak Tez Yöneticisi : Dr. Songül Bayraktar

Aralık 2023, 84 sayfa

Bu tez, ev tipi bulaşık makineleri için yeni temizleme döngülerinin tasarım aşamasında, deney ihtiyacını azaltmak için istatistiksel öğrenmeye ve çok amaçlı optimizasyona dayalı yeni bir yaklaşım önermektedir. İlk olarak, mevcut temizleme döngülerinin program akışlarını girdi verileri olarak ve ilgili temizleme döngülerinin performans laboratuvar testlerinin sonuçlarını çıktı olarak kullanarak, bulaşık makinesi temizleme döngüsünün çıktılarını tahmin etmek için özellik seçim yöntemleriyle ilişkili regresyon modelleri oluşturulur. Daha sonra, regresyon modellerinin ve seçilen özelliklerin sırasıyla amaç fonksiyonları ve bilinmeyen karar değişkenleri olarak atanmasıyla çok amaçlı bir optimizasyon problemi tanımlanır. Elde edilen optimizasyon problemi daha sonra tasarımcının tercihlerine (veya müşterilerin ihtiyaçlarına) göre evrimsel algoritmalar kullanılarak çözülür.

Anahtar Kelimeler: Bulaşık Makinesi Tasarımı, İstatistiksel Modelleme, Öznitelik Seçimi, Çok Amaçlı Optimizasyon, Evrimsel Algoritmalar

ACKNOWLEDGMENTS

In the pursuit of academic excellence, there lies a path illuminated by the guidance, support, and inspiration of remarkable individuals. This thesis, the culmination of my academic journey, would not have been possible without the invaluable contributions of these outstanding mentors and collaborators.

Assoc. Prof. Dr. Hamdullah Yücel, my thesis supervisor, has been a constant source of wisdom and encouragement throughout this endeavor. His profound expertise, unwavering commitment, and patient guidance have enriched my academic experience immeasurably. I am deeply grateful for his tireless dedication to the pursuit of knowledge.

Dr. Songül Bayraktar, my esteemed co-supervisor, played a pivotal role in shaping the direction of my research. Her keen insights, critical feedback, and scholarly guidance have been instrumental in refining the depth and quality of this thesis. Her mentorship has been a true privilege.

Fatih Küçük, your assistance during the data preparation phase was indispensable. Your meticulous attention to detail, tireless effort, and technical expertise were instrumental in laying the foundation for this work. I extend my sincere appreciation for your invaluable contributions.

Uğur Kan, our esteemed domain expert, offered profound insights that enriched the intellectual fabric of this thesis. Your expertise, dedication, and willingness to share your knowledge were pivotal in enhancing the rigor of our research. I am deeply thankful for your invaluable support.

Selin Sarıal, your willingness to engage in thoughtful discussions and provide insightful feedback added depth and clarity to this thesis. Your intellectual generosity and collaborative spirit have been a source of inspiration.

To the esteemed thesis examining committee members:

Prof. Dr. Sinan Kalkan from the Department of Computer Engineering at METU and Assoc. Prof. Dr. Furkan Başer from the Department of Actuarial Sciences at Ankara University, your rigorous examination and invaluable feedback have been instrumental in elevating the scholarly quality of this thesis. Your expertise and discerning insights have contributed significantly to the refinement of this work, and for that, I am deeply appreciative.

To all those who have supported and believed in me throughout this journey, I extend my heartfelt gratitude. Your encouragement has been the wind beneath my wings, propelling me toward academic achievement. Your contributions have transformed this thesis into a testament to our collective dedication to the pursuit of knowledge.

In the grand tapestry of academic exploration, you have each woven threads of wisdom, expertise, and camaraderie that have enriched my educational experience beyond measure. This thesis stands as a tribute to your mentorship, guidance, and unwavering support.

Thank you for being the pillars of my academic journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DW	Dishwasher
CPI	Cleaning Performance Index
DPI	Drying Performance Index
EC	Energy Consumption
TD	Duration of Cleaning Cycle
WC	Water Consumption
DM	Decision Maker
EN	European Standards
IEC	International Electrotechnical Commission
CEN	European Committee for Standardization
CENELEC	European Committee for Electrotechnical Standardization
ETSI	European Telecommunications Standards Institute
MCDM	Multi-Criteria Decision Making
EMO	Evolutionary Multi-Objective Optimization
HDLSS	High Dimension Low Sample Size
RSS	Residual Sum of Squares
PCR	Principal Components Regression
PLS	Partial Least Squares
RFE	Recursive Feature Elimination
SFS	Sequential Feature Selection
PSO	Particle Swarm Optimization
HHO	Harris Hawks Optimization
GWO	Grey Wolf Optimizer
DFO	Dragonfly Algorithm
GO	Genetic Optimization
NN	Neural Network
K-NN	K-Nearest Neighbors
SVR	Support Vector Regression

SVM	Support Vector Machine
RF	Random Forest
GBM	Gradient Boosting Machine
CatBoost	Categorical Boosting
MAE	Mean Absolute Error
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
CV	Cross Validation
PF	Pareto Front
MOLP	Multi-Objective Linear Program
MOOP	Multi-Objective Optimization Problem
GDF	Geoffrion-Dyer-Feinberg
MOEA	Multi-Objective Evolutionary Algorithms
MOGA	Multi-Objective Genetic Algorithm
SPEA	Strength Pareto Evolutionary Algorithm
MOO	Multi-Objective Optimization
C-TAEA	Constrained Two-Archive Evolutionary Algorithm
NSGA-III	Non-Dominated Sorting Genetic Algorithm III
RVEA	Reference Vector Guided Evolutionary Algorithm
1PW	Pre Wash Block
2MW	Main Wash Block
3MF	Micro Filter Cleaning
4CR	Cold Rinse
5ER	Extra Rinse
6HR	Hot Rinse
7RS	Last Second Rinse
8DS	Water Storage Unit Filling
9RY	Resin Wash
10DY	Drying

CHAPTER 1

INTRODUCTION

Household dishwashers are humans' most critical partners in kitchens and significantly affect the sustainable world. Dishwashers can clean the dishes with 75% less water and 25% less energy concerning hand-washing, with a satisfying result [11]. The result of a dishwasher cleaning cycle can be expressed with the Cleaning Performance Index (CPI) and Drying Performance Index (DPI) according to EN 60436 standard, that is, "Electric dishwashers for household use - methods for measuring the performance" [26].

Sinner's Circle [64] explains the primary cleaning mechanism. Four factors are critical for hand-washing and dishwasher cleaning: chemistry, temperature, time, and mechanical action. The interaction of these factors for cleaning performance is given in Figure 1.1. Chemistry in the Sinner's Circle is detergent, and the dishwasher performs the cleaning process using an appropriate detergent specified in EN 60436 standard. The other factors in the Sinner's Circle, i.e., time, temperature, and mechanical actions, are the inputs of the cleaning cycle. They are designed and defined in the dishwasher cleaning cycle (also called the program). According to the needs of customers, there are different kinds of programs. The most important one is the eco program which is the energy label program of the dishwasher. The eco program is designed to use minimum energy and water to achieve acceptable CPI and DPI performances. Therefore, its duration is relatively longer compared to other programs.

A designer should consider the customer's needs and optimize the CPI, DPI, Energy Consumption (EC), Water Consumption (WC), and Time Duration (TD) to satisfy them. For example, in a fast program, the duration of the program needs to be short,

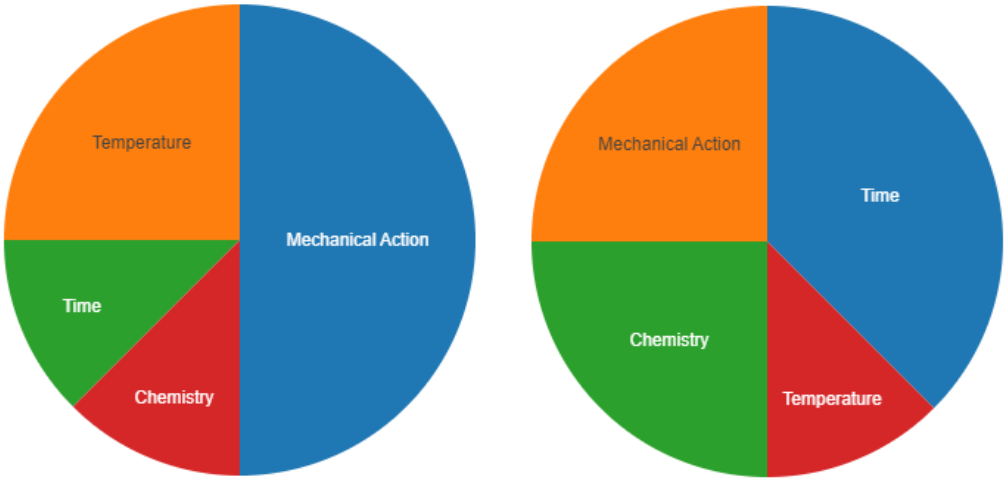


Figure 1.1: Sinner's Circle for handwash (left), dishwasher (right).

but the CPI should still be good enough. On the other hand, in an intensive program for heavily soiled pots and pans, the CPI and DPI must be perfect, but EC, WC, and TD can be sacrificed. To achieve the desired performance values in the cleaning cycle design, the designer needs to adjust the Sinner's Circle factor values in the program flow which will be the primary goal of this thesis.

Once a new cleaning cycle has been designed, it is important to verify the results. The traditional and widely accepted method of verification involves conducting experiments, which can be both costly and time-consuming. However, a predictive model that predicts the CPI, DPI, EC, WC, and TD would significantly reduce the need for experiments during the design phase of new cycles, making it a highly effective solution.

In the literature, there are various studies about modelling the dishwashers. These studies are fundamentally modelling the physical system by the components. The study in [47] describes the development of a simulation model for the hydraulic system of a commercial dishwasher. In [53], the study focuses on developing an integrated model to predict performance. The aim of this study is to assess the effectiveness of various cleaning agents in removing soil from soil surfaces, using a Fluid Dynamic Gauge. The paper [63] focuses on developing a model that can predict the energy, water usage, and duration of automatic dishwashing machines used in private households across Europe. On the other hand, with the improvement of computer process capacities, statistical learning approaches have begun to be used in the industrial

area. A statistical supervised learning problem can be defined to predict the outputs of a cleaning cycle. In the literature, predictive models are given as a framework in [7, 15, 38, 50, 53]. Instead of doing expensive and lengthy experiments, the designer can predict the result of the designed cleaning cycle using the prediction models. In general, the predictive models are the supervised learning models, and they are based on the experiment results that have been done up to today.

The predictive models use the designed cleaning cycle's steps as independent variables. Typically, a dishwasher cleaning cycle comprises ten blocks with different steps. Each step is a dishwasher operation that considers factors from Sinner's Circle, such as temperature value, circulation time, fan, or waiting time. This thesis aims to construct a framework to create a statistical model that predicts the dependent variables (CPI, DPI, EC, WC, TD) of a cleaning cycle from the independent variables, which are the designed cleaning cycle steps. The proposed learning methodology is a supervised learning since the data contain outputs of the performed experiments for the specific cleaning cycles. Underlying learning data in this study consist of 154 cleaning cycles (number of samples), each with more than 450 steps, in where these steps are the independent variables of the model. As the number of independent variables exceeds the number of observations, a least-square methodology may not be effective in predicting the outputs. This is called as a high-dimensional problem [37, 54]. To overcome this issue, we will take an advantages of feature selection methods [14], which are applied in various applications, such as in [52] for reservoir characterisation, in [3, 67] for behaviour of tourists, in [51] for financial time series, in [1] for bankruptcy, and in [75] for energy.

After selecting the essential features, (non)-linear regression models [37] can be studied to predict the outputs of the cleaning cycle. Linear regression with different types of regularization, such as lasso and ridge, can be used if the high dimensionality problem exists. As a non-linear model, k-nearest neighbors and regression trees [15] are good choices, while random forests and boosting methods are also prevalent nowadays as an ensembling method. The critical point is the selection of the model among these alternatives. A goodness of fit measurement based on, for instance, mean absolute error (MAE) or R-squared, can be used to select the model. In addition, all the scores are calculated by cross-validation to detect overfitting.

The prediction model can be considered as a digital twin of the performance laboratory that performs the experiments according to EN 60436 standard. By creating a digital twin of the laboratory, there will be great effectiveness in the time and cost of designing new dishwashers. According to the data provided by Arçelik A.Ş. [4], a new dishwasher cleaning cycle needs approximately ten experiment sets with 5 repetitions for both the test of the new dishwasher and the reference dishwasher with a 350-hours of time and 3 person-months cost. While a digital twin of a performance laboratory can save this amount of time and cost, also the time to market for the new designs is shortened.

After having statistical models to predict the CPI, DPI, EC, WC, and TD of a cleaning cycle, we need to find an answer the following question: Can we use these models to achieve the best, preferred cleaning cycle?

Up to now, the dishwasher cleaning cycles have been designed according to the knowledge of domain experts by trial and error with lots of experiments. The design of the dishwasher cleaning cycle actually needs simultaneous optimization of CPI, DPI, EC, WC, and TD in the incomparable units and conflict among them. Such kinds of problems are called as multi-objective optimization problems. In a multi-objective optimization problem, as one can easily understand, there are multiple optimum solutions, called as Pareto optimal solutions or non-dominated solutions [22]. However, in the real life, only one solution must be selected for the implementation. In the multi-objective optimization problems, the decision maker (DM) chooses the most preferred solution out of this set after finding a set of Pareto optimal solutions. The importance of the designer (also called the decision maker) is now in play. If the designer is looking forward to a cleaning cycle with an ecological interest, the DM will choose low EC and WC and enough CPI and DPI with tolerable TD. However, if DM is looking forward to a short program, the DM will choose low TD and enough CPI, DPI with tolerable EC and WC. This is an exact definition of a multi-objective problem. In this thesis, we will construct a framework to solve a multi-objective optimization problem designed by using statistical models for CPI, DPI, EC, WC, and TD as objective functions.

Multi-objective optimization problems (MOOPs) can be solved in the literature by

using two basic methods: multi-criteria decision making (MCDM) [35] and evolutionary multi-objective optimization (EMO) [12, 49]. MCDM is a mathematical programming technique, which is based on the DM interaction with the solution, whereas EMO, a population-based approach, finds an approximation of the whole Pareto front in one run [22, 45].

Overall in this thesis, we aim to design an effective cleaning cycle for the dishwasher based on the statistical approaches and multi-objective optimization techniques. The prediction models with feature selection are used in a digital twin laboratory that helps the designer to develop the new cycles. Then, the predicted models and selected features are used as objective functions and unknown variables, respectively, in the MOOPs to find the best cleaning cycle according to DM's preferences. This framework can be used in all types of household devices programs, and it can be an initial step to intelligent household products. The schematic structure of the proposed workflow is given in Figure 1.2.

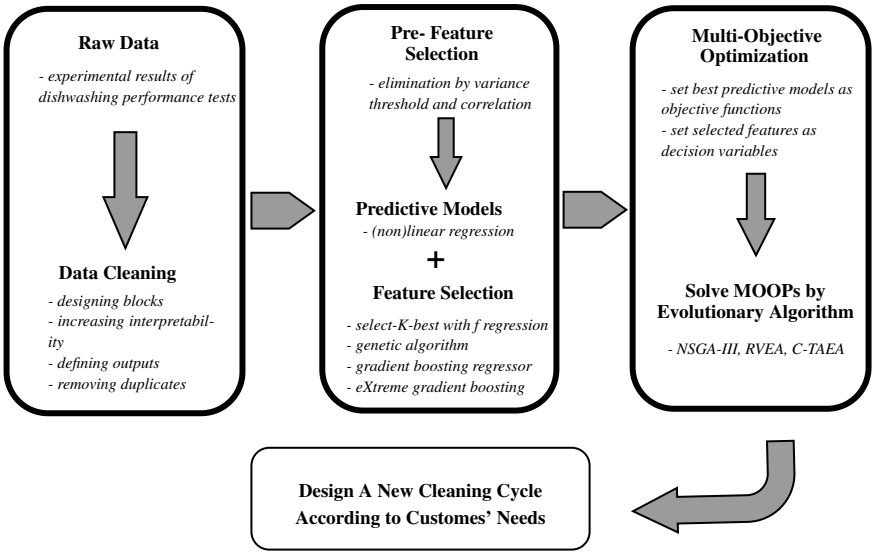


Figure 1.2: Workflow of the proposed framework.

The rest of the thesis is given as follows:

- In Chapter 2, the preliminaries related to prediction models and multi-objective optimization are given.
- In Chapter 3, the statistical modelling problem is solved, the results are given in a systematic order, and a novel framework is developed to be used in the

similar problems. Also, a digital twin of the performance laboratory is studied in this chapter.

- In Chapter 4, the multi-objective optimization problem is solved, and the results are given. A systematic methodology is provided to design a cleaning cycle. As a case study, a design of a cleaning cycle with reduced EC is investigated.
- In Chapter 5, concluding remarks are given with possible future work.

CHAPTER 2

PRELIMINARIES

In this thesis, we try to design a new cleaning cycle for a dishwasher. The idea is first to predict the performance of the cleaning cycle and then, by using the prediction models as the objective functions, to design a new cycle by solving the corresponding optimization problem. Before all these discussions, some preliminaries needed for the rest of the thesis will be presented in this chapter. We start with discussing the dishwashers' fundamental principles. After that, predictive models will be introduced and discussed with the details. Last, the multi-objective optimization problems are stated and the numerical algorithms in the literature to solve them are clearly presented and compared.

2.1 Fundamentals for a Dishwasher

A dishwasher is a mechanical system that cleans dishes using water and detergent. International standards and regulations are based on some principles, and companies have to obey these rules. In this section, we introduce main principles of a dishwasher.

2.1.1 Cleaning Cycles

A standard dishwasher can be seen in Figure 2.1. The dishwashers have three fundamental functions: cleaning, rinsing, and drying. The dishwasher sprays the water with detergent on the dishes for cleaning purposes, and rinsing is obtained by spraying cold or hot water with rinse-aid on the dishes. Drying is removal of residual moisture

from dishes through heating, evaporating, ventilating, and cooling.

Figure 2.1: A standard dishwasher [9].

Geetha and Tyagi [32] report that the demand for sustainable laundry and dishwashing products is rapidly increasing due to the current environmental pressures, rapid urbanization, and escalating prices of petrochemical feedstock. The number of dishwashers used in households is also on the rise, with around 30 million dishwashers being sold annually. Geetha and Tyagi further emphasize the importance of low energy and water consumption, as well as high-performance values when it comes to purchasing automatic washing appliances, as these factors are crucial for consumers. In general, the consumption and performance values of the dishwashers depend on their program algorithms. Dishwashers are run according to the pre-designed programs that manage the cleaning cycle steps one by one.

As given in Figure 2.2, the cleaning cycle consists of six master blocks out of ten. They are pre-wash block, main wash block, cold rinse block, two hot rinse blocks, and drying block. Each block also consists of several steps. Each step is an operation like heating, water inlet, drying, and circulation with an attribute of operation like temperature, duration, rpm of the pump, and amount of water. A designer changes these steps and develops a program to satisfy customer needs on the outputs.

We can define the crucial outputs of a cleaning cycle by analyzing today's trends. One of them is the pandemic's effect; cleaning and hygiene are the customers' shared and topmost priorities in household dishwashing. Environmental awareness, sustain-

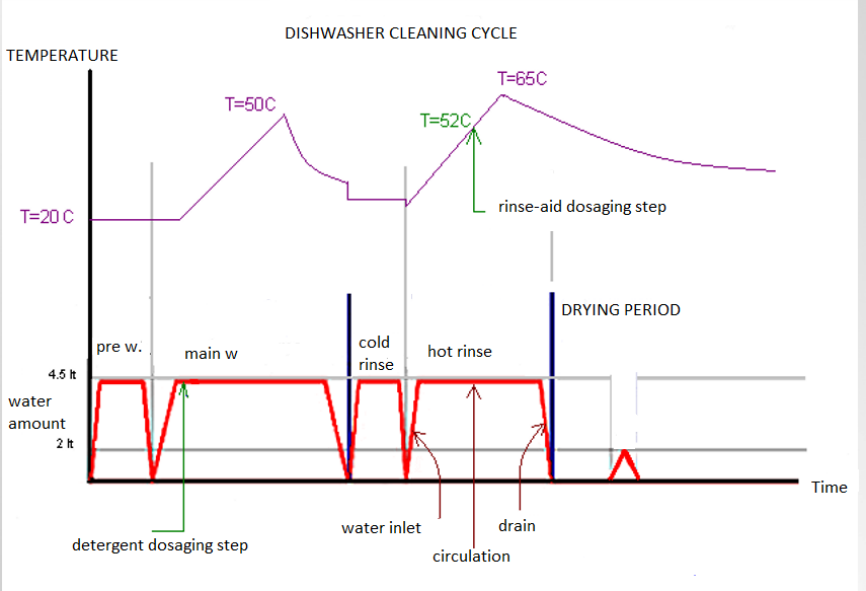


Figure 2.2: A typical dishwasher cleaning cycle [5].

ability, and time are also in the focus of the customers. Hence, overall the outputs of a dishwasher can be listed as follows:

- cleaning performance index (CPI),
- drying performance index (DPI),
- energy consumption (EC),
- water consumption (WC),
- duration of the cleaning cycle (TD).

Right now, the critical issue is how we can measure these outputs.

2.1.2 International Standards

The performance measurements are done by experiments defined according to international standards published by the European Standards (EN) and the International Electrotechnical Commission (IEC). EN is a technical standard drafted and maintained by the European Committee for Standardization (CEN), the European Committee for Electrotechnical Standardization (CENELEC), and the European Telecommunications Standards Institute (ETSI). On the other hand, IEC is an international

standard organization that prepares and publishes international standards for all electrical, electronic, and related technologies [41].

The dishwashers are designed to satisfy the requirements of the Commission Regulation (EU) 2019/2022 of 1 October 2019 laying down ecodesign requirements for household dishwashers under Directive 2009/125/EC of the European Parliament and of the Council amending Commission Regulation (EC) No 1275/2008 and repealing the Commission Regulation (EU) No 1016/2010 [73]. The object is to state and define the principal performance characteristics of electric dishwashers.

The designers of the dishwashers should obey these standards and regulations before presenting the new design to the market; due to these regulation, verification of a new design becomes more challenging. Doing many expensive and lengthy experiments, the current approach in this industry, is necessary and resulting in a long design-to-market time with increasing design costs. The remedy is to model the cleaning cycle of dishwasher and predict the output more efficiently.

2.2 Predictive Modelling

Mathematical models are constructed using mathematical concepts such as functions and equations to represent real-world phenomena. The construction of mathematical models involves moving from the concrete world into the abstract world of mathematical concepts. These models are created to help understand, analyze, and predict real-world phenomena [24]. Similarly, we can mathematically model a dishwasher's cleaning cycle and can predict the cycle's output. Using mathematical models, we can obtain the result of the new, manipulated cycles without doing experiments.

While making modelling, the critical issue is choosing a complex and non-interpretable model or a simple and interpretable one. This phenomenon is a trade-off between a white-box model and a black-box model. Accurate black-box models, such as neural networks and gradient boosting models, have excellent accuracy with low interpretability; however, in traditional statistics models, simpler models such as linear regression and decision trees have the less predictive capacity with clear interpretability. In general, while making predictions, there is a conflict between accuracy and sim-

plicity; see, e.g., [15] for more discussion.

Predictive modelling is a conceptual framework to construct a statistical model from the data to predict the system's future behaviour. It consists of exploring and cleaning the data, selecting essential features from the data, training and evaluating models, and selecting the best model among models according to evaluation criteria [72].

2.2.1 Explore and Clean the Data

In order to create accurate predictive models, the data needs to be in a specific format and ready for use. However, in real-life scenarios, the data is often raw and problematic. This can impact the performance of the predictive model. To address this issue, the necessary steps for improving the quality of data for prediction modeling will be discussed in the following sections.

2.2.1.1 Obtaining a Tidy Data Format

As a first step, the data must be in a tidy data format [70]. In this format, each variable forms a column, each observation forms a row, and each type of observational unit forms a table.

2.2.1.2 Elimination of Duplicate Data

Many datasets contain duplicate samples or rows. To ensure accuracy, it is necessary to analyze the data and remove duplicates by keeping only one instance.

2.2.1.3 Elimination of Outlier/Anomaly Data

Various techniques exist for identifying outliers in the literature [10]. These methods usually employ distance measures and clustering, and it is essential to understand the samples' outlier behaviour better before modelling. Outlier detection algorithms are unsupervised, like one-class support vector machine (SVM) [2], and long short-term memory (LSTM) [27].

2.2.1.4 Elimination of Data with Low Variance

Independent variables with less than a given threshold variance value is dropped in the statistical modelling to reduce the computational complexity. Variance is a metric that shows the average distance between the individual points and the points' mean, formulated as follows:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2,$$

where $\bar{x}$ is the mean and n is the number of points. According to the data's characteristics, the variance threshold value is determined.

2.2.1.5 Elimination of Data with Collinearity Problem

According to [43], the collinearity problem is a high correlation between independent variables and affects the quality of the fitting. The primary way to avoid this issue is to leave one feature and drop others among correlated independent variables. If the Pearson correlation [34], a measure of linear correlation between two datasets, is greater than 0.9 between two independent variables, one should be dropped.

2.2.1.6 Improvement on Imbalance Problem

Generally, the outputs must be in a normal distribution for a suitable fit quality; see, e.g., [14] for more details. The dependent variables can be checked for normality by the Anderson-Darling Normality test [65], which is a statistical test of whether a normal distribution can describe a given sample of data. If the set is not normal, Box-Cox transformation [14], a methodology to transform non-normal dependent variables into a standard shape, can be applied.

2.2.2 Feature Selection from the Data

Feature engineering is the most critical step in predictive machine learning. In educational issues, feature engineering has been completed in most datasets, and most of

the attention is on the methods. However, in real-life problems, datasets are raw and need feature engineering.

In a predictive modelling problem, the sample size, n, and the number of independent variables, p, play a significant role in the model's accuracy. If $n > p$, prediction accuracy will be suitable for linear regression models. If $n = p$, then the results will not be suitable for least-squares. If $p > n$, there will no longer unique least-square coefficient estimate [54]. Various fields are increasingly using these kind of data such as genetic microarrays, chemometrics, medical imaging, text and face recognition, and finance [28]. One of the methods to overcome the issue caused by the number of samples is feature selection. Next, we review some feature selection methods by following [37].

2.2.2.1 Subset Selection Methods

Subset selection methods determine a subset of p predictors (independent variables) that best fit the response, probably smaller than the number of observations (sample size) n. In the literature, there are different ways to determine the subsets. Some of them are as follows:

- Best Subset Selection: A linear regression model is studied for each possible combination of the p predictor. The best combinations are chosen based on metrics like mean absolute error and R-squared. Here, there are 2^p possibilities, and for computational reasons, it is not efficient.
- Forward Stepwise Selection: Starting with a model that contains no predictors, forward stepwise selection adds predictors one at a time until all predictors are in the model. At each step, the variable that improves the model fit the most is added.
- Backward Stepwise Selection: The model begins with the full least squares model containing all p predictors. Then, model iteratively removes the least helpful predictor, one at a time.

Especially, backward stepwise selection method is widely used in the feature selec-

tion.

2.2.2.2 Shrinkage Methods

Shrinkage methods contain all p predictors and regularize the coefficient values towards zero. The well-known methods are ridge, lasso, and elastic net regression.

- In ridge regression [43], one try to estimate the coefficients of multiple-regression models in which the independent variables are highly correlated, and the problem is ill-posed. In the classical least-squares method, the residual sum of squares (RSS) value is minimized by estimating βs as in

$$RSS = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2,$$

  where y is the response and x is a set of variables. On the other hand, ridge regression is like least-squares, except that the coefficients βs are estimated by minimizing the quantity

$$RSS + \lambda \sum_{j=1}^{p}\beta_j^2,$$

  where $\lambda \geq 0$ is a tuning parameter to be determined separately.

- Lasso (least absolute shrinkage and selection operator) is a regression analysis method in the statistics and machine learning that performs both variable selection and regularization to enhance the prediction accuracy and interpretability of the resulting statistical model [37]. In the ridge regression, all the p predictors are in the final model. Lasso overcomes this disadvantage by minimizing

$$RSS + \lambda \sum_{j=1}^{p}|\beta_j|.$$

  Using the l_1 penalty, some coefficients in the lasso become exactly zero, and by doing this, lasso performs variable selection, and the result becomes much more interpretable.

- In elastic net regression, the model tries to find a linear relationship between the input features and the outcome variable by minimizing the sum of squared

errors between the predicted and actual values, while also adding a penalty term to control for overfitting. The penalty term consists of two parts: the l_1 penalty (lasso regression) and the l_2 penalty (ridge regression), which control the number of non-zero coefficients and their magnitudes, respectively. The amount of regularization applied can be controlled by a hyperparameter called the elastic net mixing parameter (ratio), which determines the balance between the l_1 and l_2 penalties.

2.2.2.3 Dimension Reduction Methods

Dimension reduction methods are approaches that transform the independent variables and then fit a least squares model using these transformed variables. In these methods, the predictors/independent variables are not original ones. Principal component regression and partial least-squares are the well-known dimension reduction methods.

- Principal components regression (PCR) approach involves calculating the first M principal components and using them as predictors in a linear regression model [14, 37].
- Partial least squares (PLS) is a statistical technique that is used for both predictive modeling and dimension reduction. This method is particularly useful when there are many independent/predictor variables, and only a few observations. PLS identifies latent variables, which are linear combinations of the original predictor/independent variables, and utilizes them to predict the dependent/response variable. The selection of these latent variables is based on their ability to explain as much variation in the predictor variables as possible, while remaining highly correlated with the response variable.

2.2.2.4 Genetic Algorithm

Genetic optimization [40] is a type of optimization algorithm inspired by natural selection and genetics. In the context of feature selection for a regression model, ge-

netic optimization involves the use of a population of candidate solutions (i.e., sets of features) that undergo selection, crossover, and mutation to produce new candidate solutions in a manner analogous to the way genes are inherited and modified in the biological evolution.

The genetic optimization process starts with creating an initial population of candidate solutions, which are representing a set of features that can be used to train a regression model. The fitness of each candidate solution is evaluated by training a regression model using the selected features and measuring its performance on a validation set.

The selection process involves choosing the fittest individuals from the population to be used as parents for the next generation of candidate solutions. This is typically done using a fitness-proportionate selection scheme, where individuals with higher fitness scores are more likely to be selected as parents.

Next, crossover involves combining the selected parents to produce new candidate solutions. This is done by randomly selecting a crossover point in the parent solutions and by exchanging the features before and after that point to produce two new candidate solutions.

Last, mutation involves randomly modifying one or more features in the candidate solutions to introduce new variations. This can help explore new feature space areas and avoid getting stuck in a local optima. The selection, crossover, and mutation processes are repeated for a fixed number of generations or until a stopping criterion is met.

At the end of the genetic optimization process, the best candidate solution (i.e., the set of features that produced the highest fitness score) is selected as the final set of features for the training the regression model. This can lead to improved model performance and better generalization to new data.

2.2.3 Explanatory Data Analysis and Feature Selection in Python

Up to now, the methods of explanatory data analysis and feature selection methods are introduced. Now, libraries in Python are reviewed that are used in the numerical solutions of the high dimensional problems: filter, wrapper, and embedded methods [61].

2.2.3.1 Filter Methods in Python

In the filter methods, one sets all features, selects the best subset, and performs a learning algorithm. Some of the filter methods are summarized below.

- Variance threshold feature selection method selects a feature with a higher variance than a threshold value to prevent the model from being biased.
- Univariate feature selection with *SelectKBest* method [44, 59] is based on the univariate statistical test, e.g., chi2, Pearson correlation, etc. The base of *SelectKBest* combines the univariate statistical test with selecting the k-number of features based on the statistical result between the independent and dependent variables.

2.2.3.2 Wrapper Methods in Python

In the wrapper methods, one sets all features, generates a subset, performs a learning algorithm, and iterates until finding the best subset.

- Recursive feature elimination (RFE) [19, 57] is a feature selection method utilizing a machine learning model to select the features by eliminating the least important feature after the recursive training. That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached.
- Sequential feature selection (SFS) [29, 58] is a feature selection method to find the best features, either going forward or backwards, based on the cross-

validation score of an estimator. SFS-Forward starts with zero features, whereas SFS-Backward starts with all the features.

2.2.3.3 Embedded Methods in Python

In the embedded methods, learning algorithms have an inherent feature selection. Popular embedded methods are decision tree-based algorithms (e.g., random forest, gradient boosting [60]) and regularisation methods (e.g., lasso, ridge, and elastic [62]).

2.2.3.4 Advance Feature Selection Techniques in Python

In the *atom-ml* [6], the *FeatureSelector* class provides tooling to select the relevant features from a dataset. The feature selection also consists of advanced feature selection methods, a collection of nature-inspired optimization algorithms that maximize an objective function to select the relevant feature. Some of them are given briefly below.

- Particle Swarm Optimization (PSO) is an algorithm that simulates swarm behavior to find the optimal solution to a problem [69]. Each potential solution is represented as a particle, and the population of particles is used to search the problem space. PSO can be also used for feature selection in machine learning by identifying a subset of features that optimize model performance. The algorithm initializes a population of particles representing feature subsets and updates their positions until the optimal subset is found.
- Harris Hawks Optimization (HHO) is a meta-heuristic algorithm that simulates the hunting behaviour of Harris hawks to find the optimal solution for a given problem [68]. In the context of feature selection for a regression model, HHO can be used to identify a subset of features that optimize the model's performance.
- Grey wolf optimization (GWO) mimics the leadership hierarchy and hunting mechanism of grey wolves in nature [25].

- Dragonfly optimization (DFO) algorithm originates from static and dynamic swarming behaviours [36].

Finally, we can summarize the data preparation procedure in the Figure 2.3.

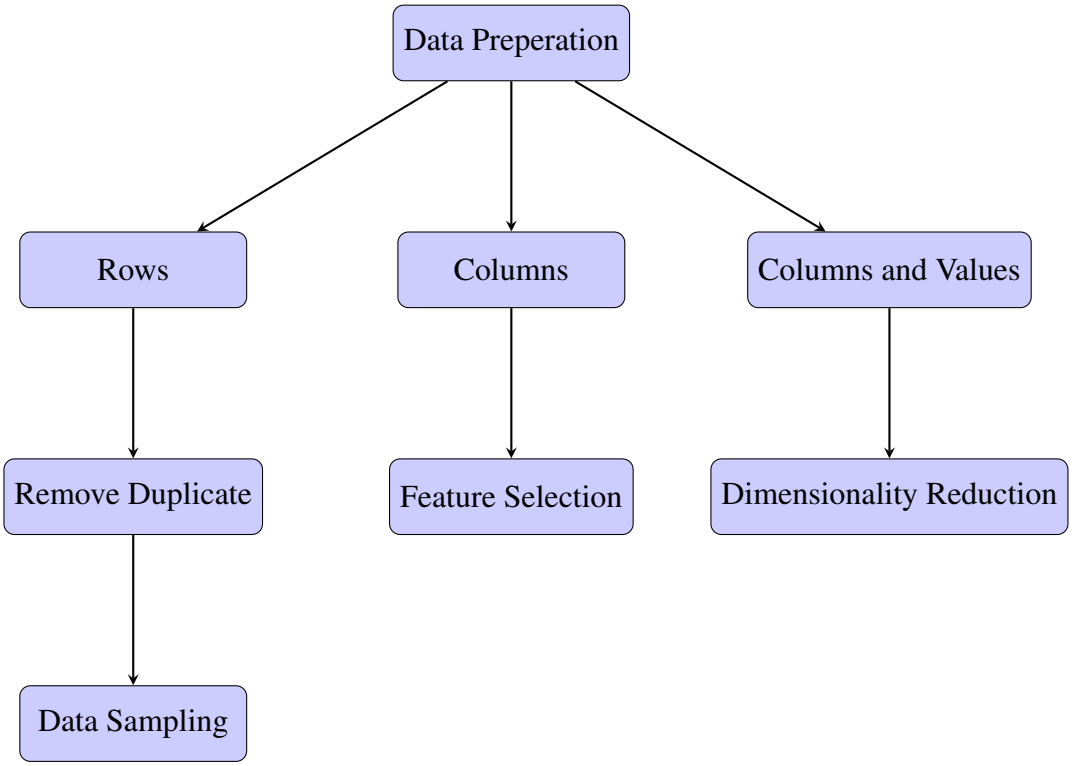


Figure 2.3: Data preparation framework.

2.2.4 Train the Model

Prediction problems are classified into two categories according to the dependent variables. If the dependent variables are numeric, then the prediction problem will be regression. In the other case, the prediction problem will be classification if the dependent variables are discrete. In this thesis, the regression models are in focus.

2.2.4.1 Multiple Linear Regression

Multiple linear regression is a statistical model that predicts the dependent variables using the independent variables [43]. The technique models the linear relationship

between the independent variables and dependent variables. This idea is based on the ordinary least-squares method and can be shown as follows:

The defined problem has multiple outputs $Y_1, Y_2, \ldots, Y_k$ that one tries to predict from inputs $X_1, X_2, \ldots, X_p$. Then, a linear model for each output is given by

$$\begin{aligned} Y_k &= \beta_{0k} + \sum_{j=1}^{p} X_j \beta_{jk} \ + \epsilon_k, \\ &= f_k(x) + \epsilon_k, \end{aligned}$$

where ϵ_k is the error of prediction and $f : X \in \mathbb{R}^p \rightarrow Y \in \mathbb{R}^k$. When we have N training cases (observations), K dependent variables (output), and p independent variables (input), the model is expressed in the matrix notation as

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E},$$

where

$$\mathbf{Y} \in \mathbb{R}^{N \times K} \ , \ \mathbf{X} \in \mathbb{R}^{N \times (p+1)} \ , \ \mathbf{B} \in \mathbb{R}^{(p+1) \times K} \ , \ \mathbf{E} \in \mathbb{R}^{N \times K}.$$

If $\mathbf{X}^T$ is invertible, the least squares estimate has the form

$$\mathbf{B} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} + \mathbf{X}^T\mathbf{E}.$$

In this case, the multiple outputs do not affect one another's least-square estimates. The least-squares estimates often have low bias but large variance. Overall to achieve the desired results, some assumptions are needed for multiple linear regressions that can be summarized [37] :

- There should be a linear relationship between the dependent and independent variables.
- The independent variables are not too highly correlated with each other.
- Residuals should be normally distributed with a mean of 0 and variance σ.

2.2.4.2 K-Nearest-Neighbor Regression (K-NN)

K-NN algorithm is a supervised machine learning algorithm, one of the non-parametric regression methods. Parametric methods such as linear regression have disadvantages

when the information is derived from data. On the other hand, K-NN can solve the problem by being a more flexible approach for the regression [37].

K-NN uses similarity measures like distance or closeness. The distance between a pair of data points, for instance, (p, q), can be calculated in three ways: Euclidean distance, Manhattan distance, and Minkowski distance. Euclidean distance represents the shortest distance between two points

$$D_e = \left(\sum_{i=1}^{n} (p_i - q_i)^2 \right)^{\frac{1}{2}},$$

whereas Manhattan distance is the sum of absolute differences between points across all the dimensions

$$D_m = \left(\sum_{i=1}^{n} |p_i - q_i| \right).$$

On the other hand, Minkowski distance is the generalized form of Euclidean and Manhattan distances

$$D = \left(\sum_{i=1}^{n} (p_i - q_i)^r \right)^{\frac{1}{r}}.$$

The basic idea of this approach is that for a fixed value of k, the predicted response for the ith-observation is the average of the observed response of the k-closest observations:

$$y_n = \frac{1}{k} \sum_{i=1}^{k} y_{n,i}.$$

Finding the optimum value of k is a crucial issue when working with k-nearest neighbor algorithm. To determine the best k value, we can divide the data into train, validation, and test sets. Starting with $k = 1$, we can calculate the accuracy of both validation and test set. Then, we can increase the value of k by 1 and plot the error graph using validation and test data. By analyzing the graph, we can identify the optimal value of k where validation and test errors are close to each other.

2.2.4.3 Support Vector Regression

The article [74] states that support vector regression (SVR) is a supervised machine learning technique for the regression problems. SVR is an extension of the support vector machine (SVM) algorithm which is used for classification problems. SVR

balances model complexity and prediction error and is good with high-dimensional data. SVR can define how much error (ϵ) is acceptable in the model and find an appropriate line (or hyperplane in higher dimensions) to fit the data. In the SVR, the objective function and constraints can be defined as

$$\begin{aligned} \min_{\beta} \quad & \frac{1}{2}|| \beta ||_2^2 \ , \\ \text{s.t.} \quad & |y_i - \beta_i x_i| \leq \epsilon, \end{aligned}$$

where ϵ is the accepted error and β is the coefficient vector. Further, using a kernel, SVR can efficiently handle a nonlinear regression problem by projecting the original feature into a kernel space where data can be linearly discriminated.

2.2.4.4 Decision Tree Regression

Decision tree regression is a supervised machine learning technique used when the interpretation is essential. The tree-based structure makes the model human-readable, and models explain which attributes are used and how the attributes are used to reach the predictions. The disadvantage is that the accuracy could be more competitive with other methods. With the help of bagging, random forests, and boosting, which are reviewed later, the accuracy of decision trees can be improved.

For a given dataset consisting of p inputs and n observations, that is, (x_i, y_i) for $i = 1, 2, \ldots, n$ with $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$, the first step in the decision tree algorithm [43] is to decide on the splitting variables and split points. Starting with M regions $R_1, R_2, \ldots, R_M$, we model the response as a constant c_m in each region as follows:

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m).$$

Then, we minimize the sum of squares of the error

$$\sum_{i=1}^{n} (y_i - f(x_i))^2.$$

As we model the response as a constant c_m in each region, the constant c_m should be just the average $\big(ave(\cdot)\ \big)$ of related response data points y_i in region R_m as

$$c_m = ave(y_i : x_i \in R_m).$$

To find the best binary partition in terms of the minimum sum of squares of errors, starting with all of the data, consider a splitting variable j and split point s, and define the following pair of half-planes

$$R_1(j,s) = \{X : X_j \leq s\} \text{ and } R_2(j,s) = \{X : X_j > s\}.$$

Then, the splitting variable j and split point s are calculated by solving

$$\min_{j,s} \left[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right].$$

For any j and s, the inner minimization is solved by

$$c_1 = ave(y_i : x_i \in R_1(j,s))$$

and

$$c_2 = ave(y_i : x_i \in R_2(j,s)).$$

Efficiently determining the split point s for each variable is possible. Therefore, by scanning through all inputs, we can find the best pair (j,s). Once we have identified the best split, we can partition the data into two regions and repeat the splitting process on each region. This process is then repeated for all resulting regions.

This process ends with a large tree, and it can overfit the data. In this case, the remedy is to stop the splitting according to the node size and, afterwards, prune the tree using cost-complexity pruning.

2.2.4.5 Random Forest Regression

Random Forest is a tree-based supervised machine learning algorithm. The algorithm uses ensemble learning by combining multiple decision trees to determine the final output rather than looking at individual decision trees.

Random Forest [15] is a machine learning algorithm that is generally based on a technique called bagging or bootstrap aggregating, which aims to improve the stability and accuracy of the model, reduce variance, and prevent overfitting. The bagging technique consists of two steps. In the first step, bootstrap sampling is performed

on the training data to obtain subsets of the data chosen randomly with replacement. These subsets are then used to train decision trees. In the bagging process, n decision trees are constructed using bootstrap sampling. This results in an ensemble of different models [72]. In the second step, aggregation is performed. The outputs from all the separate models are aggregated into a single prediction, which is simply the average of predicted outcome values. This process helps to further improve the accuracy and reliability of the model.

In the random forest [43], different from bagging, a random sample of m predictors is chosen as split candidates from the full set of p predictors when building decision trees. This prevents one strong predictor from dominating all decision trees and producing similar models. Overall, the random forest algorithm is a powerful tool that can be used in a variety of applications, such as classification and regression problems.

2.2.4.6 XGBoost Regression

XGBoost is the first of **The Big Three** gradient boosting frameworks, released in 2014. The other two are LightGBM by Microsoft [46], launched in 2016, and CatBoost [23] by Yandex, launched in 2017. These frameworks are well-known tools for regression or classification problems [16].

The XGBoost regression method is a robust machine-learning algorithm that uses an ensemble of decision trees to make predictions. The algorithm builds a series of decision trees sequentially, where each subsequent tree corrects the errors of the previous one.

The primary mathematical principles behind XGBoost regression are gradient boosting and decision trees. Gradient boosting is an optimization algorithm that seeks to minimize the loss function by iteratively adding new models trained to correct the residuals (the difference between the predicted and actual values) of the previous model. Decision trees, on the other hand, are a supervised learning technique that recursively partitions the data into smaller subsets based on the values of the input features.

2.2.4.7 Neural Network Regression

Neural network regression is a type of machine learning algorithm that uses an artificial neural network to learn the relationship between input features and output values [17]. In the regression tasks, the network is trained to predict a continuous output variable given a set of input features. The network architecture typically consists of one or more hidden layers of neurons that perform nonlinear transformations on the input data, followed by an output layer that produces the final prediction. The weights of the network are adjusted during training using backpropagation to minimize a loss function that measures the difference between the predicted and actual output values.

One advantage of neural network regression is its ability to model complex nonlinear relationships between the input features and output variables. Neural networks can learn to extract relevant features from high-dimensional input data and can capture subtle interactions between variables that may be difficult for other algorithms to detect. Neural networks are also highly flexible and can be adapted to a wide range of regression tasks by adjusting the number of layers, neurons, and activation functions used. However, neural networks can be computationally expensive to train and may require large amounts of data to achieve good performance. Overfitting is also a potential issue with neural networks, especially when the model is large and the training data is limited. Regularization techniques such as, dropout and weight decay, can help prevent overfitting, but they can also add to the complexity and computational cost of the model.

2.2.5 Evaluate the Model

One of the most common ways to evaluate the performance of a model is based on accuracy [55]. Error metrics measure the dissimilarity between the actual solution y and the predicted solution $\widehat{y}$. To quantify the error, there are different metrics in the literature [15, 37, 43], which will be reviewed next.

2.2.5.1 Mean Absolute Error (MAE)

Mean absolute error, mean of the error for all samples, is

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|,$$

where n is the number of samples. This measure can be used to interpret the range of predictions directly. For instance, if the MAE is 10 and the model predicts 100, the prediction ranges between 90 and 110. This means the error of the model is ± 10. MAE is unsuitable when the deviation of predicted output is large and does a poor job when the scale of the data is large.

2.2.5.2 Mean Squared Error (MSE)

Mean squared error computes a risk metric corresponding to the expected value of the squared error. For given n samples, it is defined by

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2.$$

Although squaring emphasizes enormous differences, it is more convenient than MAE since it effectively penalizes the outliers. For instance, if the error is 0.1, its magnitude effect as a whole will be 0.01, we can say it is negligible, and if the error is 10, its magnitude effect will be 100, which is known as penalizing the outliers. Since we want to penalize the outliers, MSE fulfills the desired property of regression models.

2.2.5.3 Maximum Error

Maximum error refers to the highest possible difference between the predicted value and the true value. This metric is crucial in situations where accuracy is critical, such as the medical industry. In a single output regression model that is perfectly fitted, the maximum error would be 0 on the training set. However, this scenario is highly unlikely in the real world. Generally, the maximum error indicates the level of error that the model encountered during the fitting process. Mathematically, it is formulated as follows

$$MaxError(y, \widehat{y}) = \max_{1\leq i\leq n}|y_i - \widehat{y_i}|.$$

Maximum error seeks to identify the highest level of error the model produces for a single sample. It assists in determining the optimal model that encompasses all the samples.

2.2.5.4 Root Mean Squared Error (RMSE)

Root mean squared error, the square root of MSE, is mainly used to scale the MSE down. RMSE is a measure of how to spread out residuals. In other words, it shows how concentrated the data is around the line of best fit.

2.2.5.5 Explained Variance Score

Explained Variance Score is calculated as follows

$$Explained\,Variance(y, \hat{y}) = 1 - \frac{Var(y, \hat{y})}{Var(y)}.$$

Here, 1 is the best evaluation score possible for a model, and a negative value is considered to be not correctly trained models. This score gives information about the variance of the whole model.

2.2.5.6 R-Squared Score

R-squared (R^2) is a metric used to evaluate the performance of a model. It calculates the coefficient of determination, which represents the percentage of variance in the actual solution that is explained by the independent variables in the model. R-squared indicates the goodness of fit of the model and measures how well it can predict unseen test data by explaining the proportion of variance. However, it's important to note that the variance is highly dependent on the dataset, so R-squared scores should not be compared across different datasets. Over n samples, R-squared can be calculated as,

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\widehat{y_i} - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2},$$

where $\widehat{y_i}$ is the predicted values and $\bar{y}$ is the average of the real values. Here, the best possible score is 1. The R-squared value can also be negative (since the model can be arbitrarily worse). A constant model that always predicts the expected value of y, disregarding the input features, would get an R-squared score of 0.

2.2.5.7 Cross Validation

After training a model, a numerical estimate of the difference between the predicted and initial responses is done - this is known as the evaluation of the residuals [55]. This evaluation method is called the training error and only provides insight into how well the model performs on the training data. However, this does not guarantee that the model will perform well on new, unseen data, as it may be either underfitting or overfitting the data. To address this issue, cross-validation is used to determine how well the model will generalize to unseen data.

Cross-validation is a technique used in the regression modelling to assess the performance and generalizability of the model. It involves partitioning the available data into subsets, typically called "folds". The underlying model is trained on a subset of the data, known as the training set, and then evaluated on the remaining subset, known as the validation set. There are various cross-validation techniques in the literature; such as the holdout method, k-fold cross-validation, or leave-p-out cross-validation [43].

2.2.6 Accept the Model

When considering whether to accept a statistical regression model, it is essential to take several steps. These include evaluating the model's assumptions, assessing the goodness of fit, and interpreting the results. We check for linearity, independence, homoscedasticity, and normality to evaluate the assumptions. Diagnostic plots can help with this. Assessing the goodness of fit involves measuring how well the model fits the data, which can be done using metrics such as R-squared and RMSE. Finally, interpreting the model's results is done by a domain expert. The regression model can be accepted if the assumptions are met, the goodness of fit is acceptable, and the

results are interpretable.

2.3 Optimization

In general, optimization is finding feasible solutions corresponding to extreme values of one or more objectives. We can classify optimization problems as single or multi-objective optimization according to the number of objectives. If there is only one objective in the scenario, then the optimization will be single-objective; if we have more than one objective, the optimization will be multi-objective, which will be our interest.

2.3.1 Multi-Objective Optimization Problems

Some problems in engineering require the simultaneous optimization of several objectives in incomparable units and conflict among them [45]. These problems are called as multi-objective optimization problems (MOOPs) in the form of:

$$\begin{aligned} minimize\ or\ maximize\ f_m(x), \qquad & m = 1, 2, ..., M, \\ subject\ to\ g_j(x) \geq 0, \qquad & j = 1, 2, ..., J, \\ h_k(x) = 0, \qquad & k = 1, 2, ..., K, \\ x_i^l \leq x_i \leq x_i^u, \qquad & i = 1, 2, ..., n, \end{aligned}$$

where $f_m,\ m = 1, \ldots, M$, are the objective functions, whereas $g_j,\ j = 1, \ldots, J$ and $h_k,\ k = 1, \ldots, K$ are inequality and equality bounds, respectively. In addition, x_i^l and x_i^u are box constraints for the unknown variables for $i = 1, \ldots, n$. A solution $x \in R^n$ is a vector of n decision variables $x = (x_1, x_2, ..., x_n)^T$. The solutions satisfying the constraints and variable bounds constitute a feasible decision variable space $S \subset R^n$. The objective functions $f_m,\ m = 1, \ldots, M$, constitute a multi-dimensional space called the objective space $Z \subset R^M$. For each solution x in the decision variable space, there exists a point $z \in R^M$ in the objective space and denoted by $f(x) = z = (z_1, z_2, ..., z_M)^T$. To define the optimal solutions in the multi-objective optimization problem, we need to set some definitions [22].

Definition 2.3.1. *(Pareto Dominance Relation) A vector z^1 Pareto dominates vector*

z^2, *denoted by* $z^1 \prec_{pareto} z^2$, *if and only if*

$$\forall i \in \{1, ..., k\} : z_i^1 \leq z_i^2,$$

and

$$\exists i \in \{1, ..., k\} : z_i^1 < z_i^2.$$

All points not dominated by any other set member are called as the non-dominated points. In Figure 2.4, vector z^3 is strictly less than z^2 in both objectives; therefore $z^3 \prec_{pareto} z^2$. Vector z^3 also Pareto dominates z^1 since with respect to f_1 those vectors are equal, but in f_2, z^3 is strictly less than z^1. Since $\prec_{pareto}$ is not a total order some elements can be incomparable like the case with z^1 and z^4, i.e., $z^1 \not\prec_{pareto} z^4$ and $z^4 \not\prec_{pareto} z^1$. Similarly, $z^3 \prec_{pareto} z^4$, $z^1 \prec_{pareto} z^2$, and $z^4 \prec_{pareto} z^2$ [49].

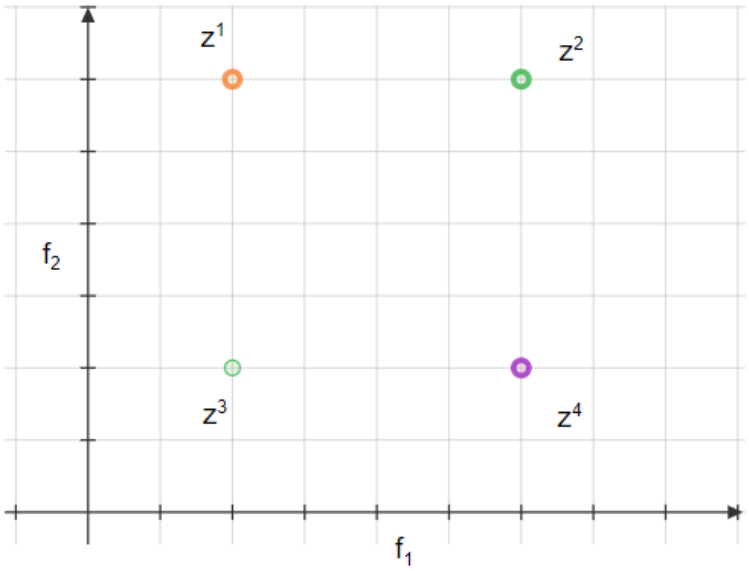


Figure 2.4: Pareto dominance relation sample [49].

Definition 2.3.2. *(Pareto Optimality) A solution* $x^* \in X$ *is Pareto optimal if there does not exist another solution* $x \in X$ *such that* $f(x) \prec_{pareto} f(x^*)$.

Definition 2.3.3. *(Weak Pareto Optimality) A solution* $x^* \in X$ *is weakly Pareto optimal if there does not exist another solution* $x \in X$ *such that* $f(x) < f(x^*)$ *for all* $i = 1, ..., k$.

Definition 2.3.4. *(Pareto Optimal Set) The Pareto optimal set,* P^*, *is defined as:*

$$P^* = \{x \in X : \nexists y \in X \text{ such that } f(y) \preceq f(x)\}.$$

Definition 2.3.5. *(Pareto Front) For a Pareto optimal set P^*, the Pareto front PF^*, is defined as:*

$$PF^* = \{(f(x) = f_1(x), ..., f_k(x)) : \ x \in P^*\}.$$

A graphical representation of Pareto optimal and Pareto front is also illustrated in Figure 2.5.

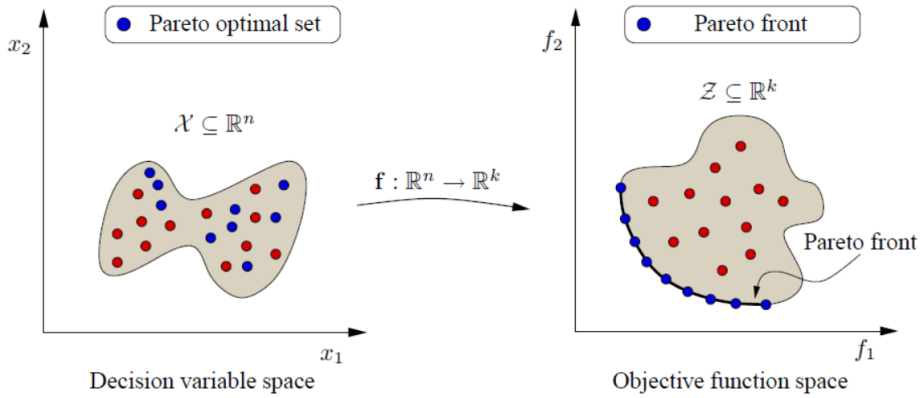


Figure 2.5: Pareto optimal (left) and Pareto front (right) [49].

Definition 2.3.6. *(Ideal and Nadir Points) The ideal point represents the lower bounds of the Pareto front and is defined by $z_i^* = \min_{z \in Z} z_i$ for all $i = 1, ..., k$. In turn, the upper bounds of the Pareto front are defined by the nadir point, which is given by $z_i^{nad} = \max_{z \in Z} z_i$ for all $i = 1, ..., k$.*

The points on the non-domination front are Pareto optimal points that are Pareto-optimal solutions. The solutions are need:

- to lie on the Pareto optimal front, and
- to be diverse enough to represent the entire range of the Pareto optimal front.

MOOPs can be solved basically by using two techniques: mathematical programming techniques and evolutionary algorithms. Mathematical programming techniques are classified regarding how and when to incorporate preferences from the decision maker (DM) into the search process. A critical issue is when the DM is required to provide preference information. There are three ways of doing this [49]: a priori approaches (e.g., goal programming [18], goal attainment method [42], lexicographic method

[45]), posteriori approaches (e.g., linear combination of weights [49], normal boundary intersection [21], ϵ-constraint method [45], and method of weighted metrics), and interactive approaches (e.g., method of Geoffrion-Dyer-Feinberg [33], Tchebycheff method [66], reference point methods [71], and light beam search [49]). However, these traditional techniques have several limitations to solve MOPs:

- they are needed to run many times to find the elements of Pareto optimal set.
- they can require domain knowledge in advance.
- they can sensitive to shape or continuity of the Pareto front.

Aforementioned reasons, we will focus on evolutionary algorithms to solve MOOPs in this thesis.

2.3.2 Evolutionary Algorithms

Evolutionary multi-objective optimization (EMO) is an approach to solving multi-objective optimization problems. An evolutionary algorithm is a stochastic direct search algorithm that, in some sense, mimics natural evolution [8]. To achieve a single solution in the mathematical programming techniques, DM preferences play an important role in selecting the solution from the Pareto front [22]. On the other hand multi-objective evolutionary algorithms (MOEAs) do not guarantee to find the actual Pareto optimal set; in general, it finds a close approximation of the optimal set in a single run. This issue is critical regarding computational time, and the MOEAs use a reasonable time to find the approximate solution. In the EMO approach, finding an approximation to the Pareto front has two aims [49]. One is minimizing the distance of the solution to the actual Pareto front, and the second is maximizing the diversity of the achieved Pareto front approximation. Diversity is critical since the solution can be located in a narrow region of the actual Pareto front in MOEAs.

An EMO contains four steps: initialization, selection, genetic operators (crossover and mutation), and termination. Initialization is creating the initial population of solutions which is a random process. If there is knowledge of the task, the initial

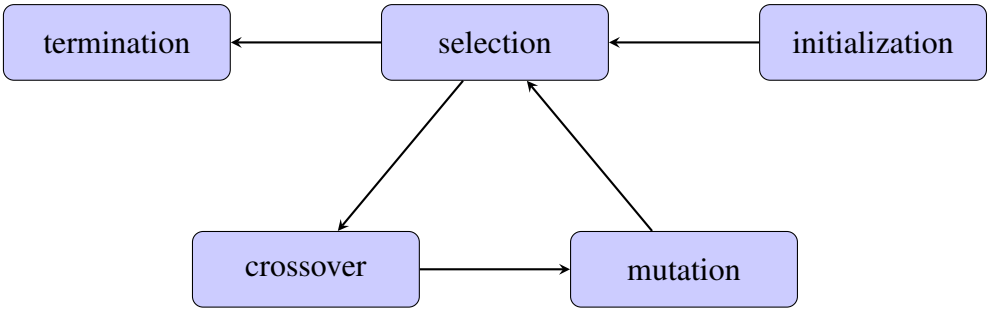


Figure 2.6: Flowchart of evolutionary algorithms.

population will be roughly centered around what is considered ideal. After the initialization, the population selection is in charge, and a fitness function evaluates the members. The fitness of all members is calculated, and their selection is made according to the top scores. Non-dominated sorting and rake selection are efficient tools in the selection step. The selected population is then used to create the next generations by genetic operators. In this step, the mutation prevents the method from getting stuck in the local extrema. Eventually, the algorithm will end with either the algorithm reaches maximum runtime or the algorithm reaches some performance threshold. The flowchart of the EMO is given in Figure 2.6.

In general, evolutionary algorithms can be considered effective approaches to solve multi-objective optimization problems due to the several reasons, which are summarized as follow:

- **Pareto Optimality:** MOOPs involve optimizing multiple conflicting objectives simultaneously, which often results in a set of solutions known as the Pareto front or Pareto set. Evolutionary algorithms are well-suited for finding and approximating this set since they maintain a diverse population of solutions and can explore different regions of the search space.

- **Global Search:** Evolutionary algorithms, such as genetic algorithms, have inherent global search capabilities. They explore the search space by maintaining a population of candidate solutions and applying evolutionary operators, such as selection, crossover, and mutation. This enables them to search for solutions in different regions of the objective space, helping to find a diverse set of Pareto optimal solutions.

- **Handling Nonlinearity and Non-Convexity:** MOOPs often involve nonlinearity and non-convexity due to the presence of multiple conflicting objectives. Evolutionary algorithms are generally robust and can handle such complexities without requiring specific problem formulations or assumptions about the objective functions. They can adaptively search for solutions in complex and irregular Pareto fronts.

- **Multi-Modality:** Evolutionary algorithms can effectively handle MOOPs with multiple modes or multiple Pareto optimal fronts. They are capable of maintaining multiple subpopulation or niches, allowing them to explore and converge to different Pareto-optimal solutions simultaneously. This ability to handle multi-modality is especially useful when the Pareto front is fragmented or has disconnected regions.

- **Flexibility and Adaptability:** Evolutionary algorithms offer flexibility in terms of problem representation, allowing various types of decision variables and constraints to be incorporated. They can handle both continuous and discrete variables, making them applicable to a wide range of MOOPs. Additionally, they can adapt their search strategy over time by adjusting parameters or operators, enhancing their convergence and exploration capabilities.

- **Interactive Decision-Making:** Evolutionary algorithms can be combined with interactive decision-making methods to involve human preferences in the optimization process. Techniques like interactive evolutionary multi-objective optimization allow decision-makers to provide feedback on solutions and guide the search towards their preferred regions of the Pareto front.

Overall, the combination of global search, ability to handle complexity, multi-modality, flexibility, and potential for interactive decision-making makes evolutionary algorithms a powerful and widely used approach for solving MOOPs.

Next, we will review some well-used multi-objective evolutionary algorithms.

2.3.2.1 Multi-Objective Genetic Algorithm

Multi-objective genetic algorithm (MOGA), developed by Fonseca and Fleming [31], is a variant of a genetic algorithm that is used in the MOOPs. The MOGA consists of the same steps as given in Figure 2.6 with differences in the selection step. In the MOGA, selection is based on the multiple objective functions, and the goal is to maintain a diverse population that contains Pareto optimal solutions for all objective functions. The flowchart of the MOGA is also given in Figure 2.7.

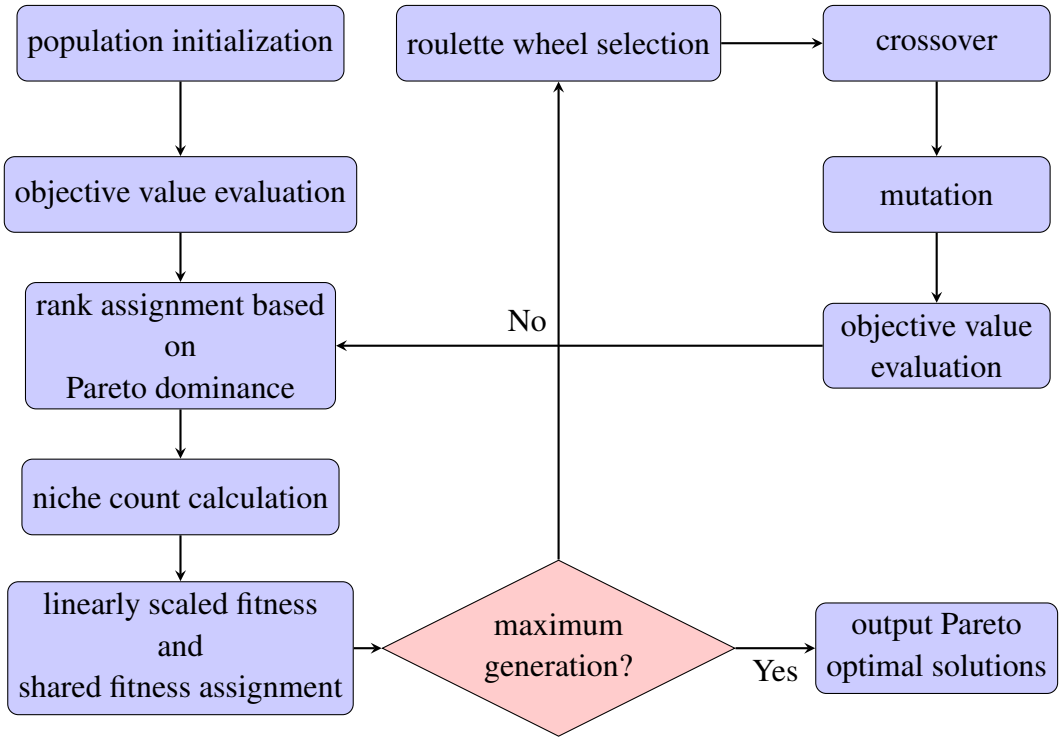


Figure 2.7: Flowchart of MOGA.

2.3.2.2 Nondominated Sorting Genetic Algorithm II & III (NSGA-II & NSGA-III)

An algorithm was developed by Kalyanmoy Deb and Himanshu Jain, which is described in [12]. The algorithm is based on genetic algorithms but with some adjustments to mating and survival selection. In NSGA-II, individuals are selected frontwise to ensure that not all individuals survive. Solutions in the splitting front are chosen based on their crowding distance, calculated as the Manhattan Distance in the

objective space. However, the extreme points are always kept in every generation and assigned a crowding infinity distance. NSGA-II also uses a binary tournament mating selection to increase selection pressure. Each individual is first compared by rank and then by crowding distance to ensure the best possible outcome.

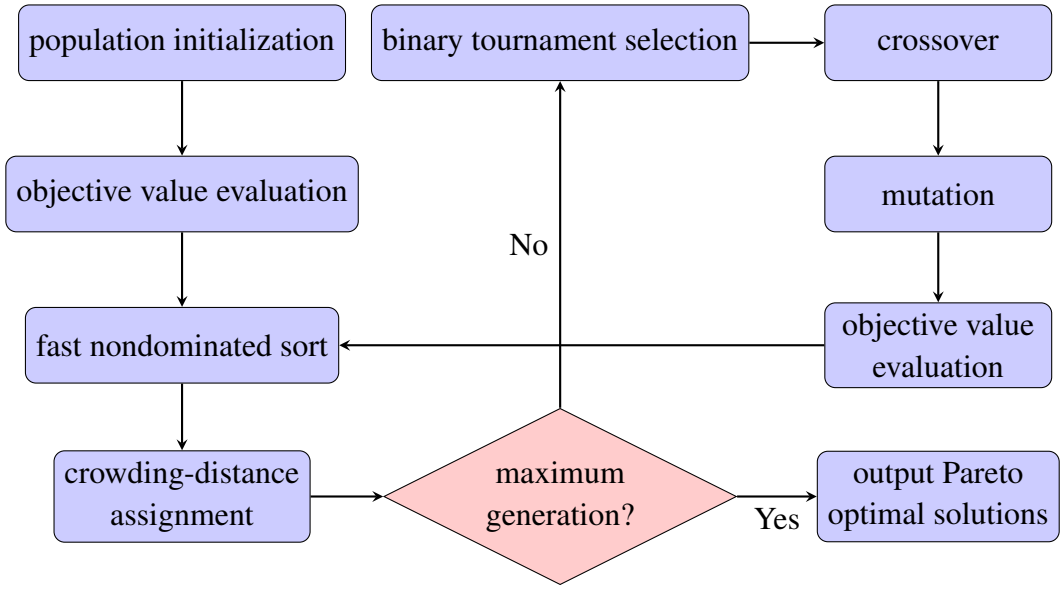


Figure 2.8: Flowchart of NSGA-II & NSGA-III.

While using NSGA-III, it is crucial to start the algorithm with reference directions. The survival process involves non-dominated sorting, like in NSGA-II. The next step is selecting solutions from the splitting front and prioritizing unrepresented reference directions. If no solutions are assigned to a reference direction, the solution with the smallest perpendicular distance in the normalized objective space is chosen. If a second solution is added to a reference line, it is assigned randomly. The goal for each reference line is to find a representative non-dominated solution. The flowchart of NSGA-II & NSGA-III is given in Figure 2.8.

2.3.2.3 Reference Vector Guided Evolutionary Algorithm (RVEA)

Reference Vector Guided Evolutionary Algorithm (RVEA) is a multi-objective optimization algorithm that aims to solve problems with multiple conflicting objectives [20]. It combines concepts from evolutionary algorithms and reference vectors to guide the search process.

In the RVEA, a set of reference vectors is defined in the objective space to represent different trade-offs between the objectives. These reference vectors are evenly distributed throughout the objective space and act as a guidance for the evolution process. Each reference vector represents a potential solution that achieves a balanced compromise between the objectives.

The algorithm starts with an initial population of candidate solutions, which are typically represented as a set of chromosomes or individuals. These individuals are evaluated based on their objective values and their distances to the reference vectors. The distance metric is used to determine the individuals' positions relative to the reference vectors and to guide the search towards a diverse and well-distributed set of solutions.

During the evolution process, the algorithm applies variation operators such as mutation and crossover to create new offspring solutions. These offspring solutions are then evaluated and compared to the current population and the reference vectors. The algorithm selects the individuals with the best objective values and distances to the reference vectors to form the next generation.

By iteratively repeating the selection, variation, and evaluation steps, the RVEA explores the search space and converges towards a set of solutions that represents a good compromise between the conflicting objectives. The final result is a diverse and well-distributed Pareto front, which represents the optimal trade-off solutions for the given multi-objective optimization problem.

2.3.2.4 Constrained Two-Achieve Evolutionary Algorithm (C-TAEA)

Different from the other evolutionary algorithms, there are two archives in C-TAEA to manage the solutions: the convergence-oriented archive (CA) and the diversity-oriented archive (DA) [48]. These play different roles in driving the optimization process towards the Pareto front and maintaining population diversity.

Convergence-Oriented Archive (CA) is responsible for promoting convergence towards the Pareto front, which represents the optimal trade-off solutions for the objectives. It stores solutions based on their objective values and aims to preserve the best solutions found so far. It acts as a driving force to push the population towards

the Pareto front by favoring solutions with better objective values. On the other hand, diversity oriented archive (DA) focuses on maintaining population diversity, ensuring that the solutions cover a wide range of the Pareto front. It stores solutions based on their diversity measures; such as distance or spread. The DA helps prevent premature convergence by encouraging exploration of the search space and preserving diverse solutions that represent different regions of the Pareto front.

By combining the CA and DA, C-TAEA balances convergence towards the Pareto front and maintaining diversity. This helps the algorithm to explore the search space effectively and to discover a diverse set of high-quality solutions that are both Pareto-optimal and feasible.

2.3.2.5 Multi-Objective Selection Based on Dominated Hypervolume Evolutionary Algorithm (SMS-EMOA)

SMS-EMOA is indeed a variant of evolutionary algorithms that incorporates the dominated hypervolume as a selection criterion for multi-objective optimization [39]. In the SMS-EMOA, the dominated hypervolume is used as a measure to assess the quality of the solutions and guide the selection process. The algorithm aims to maximize the hypervolume metric, which represents the volume of the objective space covered by the non-dominated solutions (Pareto front). By maximizing the hypervolume, SMS-EMOA encourages the generation of a diverse set of solutions that cover a large portion of the Pareto front.

2.3.3 Hypervolume Metric to Compare the Methods

After summarizing the algorithms, we need a metric to compare them. Hypervolume metric is a performance measure used in the evolutionary algorithms to evaluate and compare the quality of different solutions generated by the evolutionary algorithms. It is particularly useful for the multi-objective optimization problems, where the goal is to optimize multiple conflicting objectives simultaneously.

In the multi-objective optimization, the solutions generated by an evolutionary algo-

rithm form a set known as the Pareto front. The Pareto front represents a trade-off between different objectives, where improving one objective generally leads to a deterioration in another. The hypervolume metric quantifies the extent of the solution space covered by the Pareto front. The hypervolume metric works as follows:

- **Reference Point:** The hypervolume metric requires a reference point, which serves as the origin of the objective space. The reference point should be set based on the problem's domain knowledge and the desired trade-off between objectives.
- **Dominance:** Solutions in the Pareto front are compared using dominance relations. A solution A is said to dominate another solution B if A is better than B in at least one objective and not worse than B in any other objective. Dominance determines the inclusion or exclusion of solutions from the hypervolume calculation.
- **Calculation:** The hypervolume is calculated by computing the volume of the portion of the objective space that is dominated by the Pareto front. It represents the area between the Pareto front and the reference point. Various algorithms and techniques; such as Monte Carlo sampling or grid-based approaches, can be used to estimate this volume. In this thesis, we will use a grid-based approach due to its simplicity and cheapness. It is noted that a finer grid yields a more accurate approximation but it increases computation cost.
- **Interpretation:** A higher hypervolume value indicates a better performance of the evolutionary algorithm. It signifies a greater coverage of the objective space by the solutions in the Pareto front, indicating a diverse set of non-dominated solutions that provide a range of trade-off options [12].

CHAPTER 3

STATISTICAL MODELLING OF A DISHWASHER CYCLE

In this chapter, we use a supervised learning method to develop statistical models to predict the outputs of a dishwasher cleaning cycle. In real-life applications, the data is not ready, and therefore a significant effort will be necessary for data preparation, explanatory data analysis, and feature selection.

The chapter starts with obtaining the essential features by using statistical feature selection methods, then (non)linear regression models are constructed. After determining the features and models, a dishwasher cleaning cycle's performance can be predicted without doing experimental tests. The chapter ends with designing the digital twin laboratory and a case study about designing the best cleaning cycle.

3.1 Dishwasher Cleaning Cycle Program

In this study, our data is the dishwasher cleaning cycles and the outputs of them. From here on out, we will refer to these cleaning cycles also as "the programs". These programs are carefully crafted to meet the specific requirements and needs of the customers. Additionally, the programs are designed based on the equipment's type and level of soil.

Programs can be customized by adjusting the program steps and parameters like water amount, temperature, time, pressure, and water flow rate to achieve desired performance goals. The designer selects them for optimal output. It is important to note that these values may require adjustment and fine-tuning for optimal results. The physical

properties of the dishwasher are also crucial in the program steps. Tub dimensions of the dishwasher, such as compact size, slim size, and standard size tubs, change the number of dishes that can be cleaned in the dishwasher (loading capacity) and the dishwasher's cleaning and energy transfer characteristics. Moreover, the dishwasher can be designed as freestanding or built-in, affecting the heat transfer characteristics and drying performance design. Overall, the designer has to optimize the cleaning cycle program according to all these complex systems.

The following summarizes the main categories affecting a dishwasher program design.

3.1.1 Definition of the Program

Basically, a program is characterized with the following properties:

- Size of the Dishwasher: slim size, normal size, tall tub size.
- Type of the Dishwasher: free standing, built-in.
- Capacity of the Dishwasher: 8 to 16 place setting.
- Type of the Program: There are more than 15 program types today. These are the preferences of the user or the aims of the programs. The program aims, for example, minimum energy, minimum water usage, or to be fast. Types can be exemplified as normal wash, eco, heavy duty, quick wash, express wash, rinse only, auto or sensor wash, pots & pans, sanitize, half load, glassware cycle throughout the world.

For example, a program can be an eco program for a slim size dishwasher and it is coded with a program description like DWECO050 which is a unique definition. This thesis uses the data of all type dishwashers; however the analysis is grouped by size of the dishwasher and the studies are performed for normal & standard size dishwashers.

3.1.2 Program Flow

The programs consist of blocks with a step by step operation flow. The ten program blocks are:

- Pre-Wash (1PW): Pre-wash aims to remove coarse dirt from the system. In some pre-wash programs, there can be heating and detergent use. Pre-wash is an optional block, and some programs have no pre-wash.
- Main-Wash (2MW): A cleaning cycle block is mainly responsible for the cleaning. Temperature, mechanical, and chemical components of the Sinner Circle are in this block.
- Micro-Filter Cleaning (3MF): This is also an optional block. If the dishwasher has a function of micro-filter cleaning, the system washes the micro-filter in this block.
- Rinse Steps:
  - Cold Rinse (4CR): It is a cleaning cycle block that removes the excessive detergent and soil residuals from the surface of the dishes. There is no heating step in this block.
  - Extra Rinse (5ER): The optional cleaning cycle block increases rinsing efficiency.
  - First Hot Rinse (6HR): It is also an optional cleaning cycle block for additional rinsing by high temperature.
  - Last (Second Hot) Rinse (7RS): Rinse aid is used in this step for better drying performance.
- Water Storage Unit Filling (8DS): If there is a water storage unit option in the dishwasher, this step is charged with filling the water tank.
- Resin Wash (9RY): This block is responsible for regenerating a water softener resin in a dishwasher.
- Drying (10DY): Drying dishes are mainly obtained in this block. After rinse blocks, the dishware is relatively hot, and this block aims for natural or forced

convection. Some options in this block are natural, automatic door opening, or fan drying.

The aforementioned blocks compose the cleaning cycles. All blocks consist of steps and operations. Next the details of the program structure will be discussed.

Table 3.1: Input data of slim size dishwasher's eco program DWECO050.

Block No	Block	Step	Value	Spray Arm	Pump	Heating	Target
1	Pre-wash 1PW	NA					
2	Main-wash 2MW	water-inlet	3,40				
2	Main-wash 2MW	circulation	8,00	2-2 mix	2600	0	
2	Main-wash 2MW	detergent	18,00				
2	Main-wash 2MW	circulation	3,00	3-0 lower	2600	0	
2	Main-wash 2MW	circulation	2,00	0-2 upper	2600	0	
2	Main-wash 2MW	circulation	4,00	2-2 mix	2400	0	
2	Main-wash 2MW	circulation	16,00	2-2 mix	2400	1	53
3	Micro-filter cleaning 3MF	NA					
4	Cold-rinse 4CR	water-inlet	2,60				
4	Cold-rinse 4CR	circulation	8,00	2-2 mix	2200	0	
5	Extra-rinse 5ER	NA					
...	...	...	...	...	...	...	...
6	First hot rinse 6HR	NA					
7	Second hot rinse 7RS	water-inlet	2,80				
7	Second hot rinse 7RS	circulation	10,00	2-2 mix	2200	0	
7	Second hot rinse 7RS	circulation	1,00	2-2 mix	2200	1	35
7	Second hot rinse 7RS	rinse-aid	6,00				
7	Second hot rinse 7RS	circulation	4,50	2-2 mix	2200	1	37
7	Second hot rinse 7RS	circulation	10,00	2-2 mix	2200	1	55
8	Water tank drain 8DS	NA					
9	Resin wash 9RY	NA					
10	Drying 10DY	waiting	60,00				

3.1.3 Program Operations

The basic operations of the blocks are water-inlet, circulation, detergent and rinse-aid dosing, heating, and drying. All operations, also called program inputs, have different properties based on their charge, and affect the outputs at different levels. A sample of operation list for eco program DWECO050 is displayed in Table 3.1. We note that NA stands for "Not Applicable" in the Table 3.1.

3.1.4 Program Outputs

We have mainly five program outputs, whose descriptions are summarized below:

- Cleaning Performance Index (CPI): CPI values are $0 <$ CPI < 5,
- Drying Performance Index (DPI): DPI values are $0 <$ DPI < 100,
- Energy Consumption (EC): kilowatt hour (kWh),
- Water Consumption (WC): litre (l),
- Time Duration (TD): minute (min).

Table 3.2: Output data of slim size dishwasher's eco program DWECO050.

Program	**CPI**	**DPI**	**TD**	**EC**	**WC**
DWECO050	3.35	82	205	0.9	13.17

A sample for program outputs is displayed in Table 3.2. After introducing the input and output of programs, we can continue with the data obtained from the physical experiments.

3.2 The Data

The prediction model takes the program's operations as input data and predicts the program's outputs (CPI, DPI, EC, TD, and WC). We have 2472 experimental data in 154 different programs, and approximately 16 repeated experiments have been done for a program. The output is an average of these 16 repeated experiments for the learning data. A sample of data is given in Table 3.3.

Table 3.3: Data of slim size dishwasher's eco program DWECO050.

	OUTPUTS					**INPUTS**			
Program	**CPI**	**DPI**	**TD**	**EC**	**WC**	**I1**	**I2**	...	**I448**
DWECO050	3.35	82	205	0.9	13.17	DRAIN	WATER	...	0

3.2.1 Input Data

The input data obtained from the programs consists of 10 blocks and 448 steps. Every step is a feature/predictor/independent variable in the program. The feature vector between I1 and I112 values are functions and categorical variables. The feature vector between I113 and I224 are values of the functions and continuous data. The feature vector between I225 and I336 are valve positions, and they are categorical variables. The feature vector between I337 and I448 are pump speed in rpm values which are continuous data. In general, the underlying data can be stated as a raw data. A sample feature table is given in Table 3.4.

Table 3.4: Sample features of a dishwasher program.

Predictor No	FUNCTION	Predictor No	VALUE	Predictor No	POS_OF_VALVE	Predictor No	CIRC_RPM
I1	DRAIN	I113	0	I225	NO_ENERGY	I337	0
I2	WATERINLET	I114	0	I226	NO_ENERGY	I338	0
I3	CIRCULATION	I115	0	I227	NO_ENERGY	I339	0
I4	WAIT	I116	0	I228	NO_ENERGY	I340	0
I5	CIRCULATION	I117	0	I229	NO_ENERGY	I341	0
I6	CIRCULATION_HEATER_DETERGENT	I118	0	I230	NO_ENERGY	I342	0
I7	CIRCULATION_HEATER	I119	0	I231	NO_ENERGY	I343	0
I8	CIRCULATION	I120	0	I232	NO_ENERGY	I344	0
I9	CIRCULATION	I121	0	I233	NO_ENERGY	I345	0
I10	CIRCULATION	I122	0	I234	NO_ENERGY	I346	0
I11	CIRCULATION_HEATER	I123	0	I235	NO_ENERGY	I347	0
I12	CIRCULATION	I124	0	I236	NO_ENERGY	I348	0
...	...	...	...	...	...	...	...
I102	WAIT	I214	0	I326	NO_ENERGY	I438	0
I103	WAIT	I215	0	I327	NO_ENERGY	I439	0
I104	WATERINLET	I216	0	I328	NO_ENERGY	I440	0
I105	WAIT	I217	0	I329	NO_ENERGY	I441	0
I106	WAIT	I218	0	I330	NO_ENERGY	I442	0
I107	WAIT	I219	0	I331	NO_ENERGY	I443	0
I108	DRAIN	I220	0	I332	NO_ENERGY	I444	0
I109	DOOROPENING	I221	0	I333	NO_ENERGY	I445	0
I110	WAIT	I222	0	I334	NO_ENERGY	I446	0
I111	WAIT	I223	0	I335	NO_ENERGY	I447	0
I112	DRAIN	I224	30	I336	NO_ENERGY	I448	0

The categorical input variables are need to be encoded, and after encoding, the prediction model can be stated as $f : \mathbb{R}^{448} \rightarrow \mathbb{R}^{5}$. The interpretability of the input data with 448 features needs to be clarified. To increase the interpretability of the input data, every block is analyzed, and necessary information is derived from these blocks. Necessary attributes to define blocks, which are 16 in total, are listed in Table 3.5 according to the knowledge of the domain expert.

Now, we can set our independent variables by 10 blocks times 16 attributes for each block equal to 160 independent variables totally; see Table 3.6. A sample of the independent variables from the set of 160 are also given in Table 3.7.

Table 3.5: Cleaning cycle blocks attributes.

Abbreviation	**Name**	**Unit**
Closed_Cycle_Period	Total Period of Closed Cycle Heating	minutes
Period	Total Period of Block	minutes
Circulation_Period	Total Period of Circulation	minutes
WaterInlet	Total Amount of Water Inlet	litres
RPM	Maximum Motor Speed	RPM
Temperature	Maximum Water Temperature	C
Lower_Spray_Circulation_Period	Total Period of Lower Spray Arm Circulation	minutes
Upper_Spray_Circulation_Period	Total Period of Upper Spray Arm Circulation	minutes
Top_Spray_Circulation_Period	Total Period of Ceiling Spray Arm Circulation	minutes
Zone_Spray_Circulation_Period	Total Period of Zone Spray Circulation	minutes
Waiting	Total Period of Waiting	minutes
Fan	Total Period of Fan Operation	minutes
Door_Openening	Total Period of Door Opening	minutes
Fan_Flap	Total Period of Fan and Valve	minutes
Extra_Heater_Offset	Total Period of Gaudi Offset Heating	minutes
Extra_Heater_Nonoffset	Total Period of Gaudi Non-Offset Heating	minutes

Table 3.6: 160 independent variables in terms of blocks and attributes.

INDEPENDENT VARIABLES	1PW	2MW	3MF	4CR	5ER	6HR	7RS	8DS	9RY	10DY
Closed_Cycle_Period	1	2	3	4	5	6	7	8	9	10
Period	11	12	13	14	15	16	17	18	19	20
Circulation_Period	21	22	23	24	25	26	27	28	29	30
WaterInlet	31	32	33	34	35	36	37	38	39	40
RPM	41	42	43	44	45	46	47	48	49	50
Temperature	51	52	53	54	55	56	57	58	59	60
Lower_Spray_Circulation_Period	61	62	63	64	65	66	67	68	69	70
Upper_Spray_Circulation_Period	71	72	73	74	75	76	77	78	79	80
Top_Spray_Circulation_Period	81	82	83	84	85	86	87	88	89	90
Zone_Spray_Circulation_Period	91	92	93	94	95	96	97	98	99	100
Waiting	101	102	103	104	105	106	107	108	109	110
Fan	111	112	113	114	115	116	117	118	119	120
Door_Openening	121	122	123	124	125	126	127	128	129	130
Fan_Flap	131	132	133	134	135	136	137	138	139	140
Extra_Heater_Offset	141	142	143	144	145	146	147	148	149	150
Extra_Heater_Nonoffset	151	152	153	154	155	156	157	158	159	160

After the strategy mentioned above, we reduce the 448 input attributes to 160 independent variables. All of the attributes are now continuous data, and all of them have clear interpretability. Hence, the model becomes $f : \mathbb{R}^{160} \rightarrow \mathbb{R}^5$. After data preparation, we fully conform to a tidy data format explained in Section 2.2.1.1. Each variable forms a column, and at the same time, each observation/sample program forms a row. This formation is named as DATASET 0 with 160 independent variables, 5 dependent variables, and 154 samples.

Table 3.7: A sample set of 160 independent variables.

Block	Attribute	Independent Variable	Sample Value	Unit
1PW	WaterInlet	1PWWaterInlet	5	l
2MW	RPM	2MWRPM	2800	rpm
2MW	Temperature	2MWTemperature	52	C

3.2.2 Output Data

The output data consists of the cleaning cycles' experimental results which are CPI, DPI, EC, WC, and TD. Next, we will discuss the statistical analysis of the underlying output data.

3.2.2.1 Cleaning Performance Index (CPI)

The CPI value is the cleaning performance index, and is between 0 and 5. After the cleaning cycle ends, the evaluator points to all dishes individually and averages the whole dishes' evaluation according to EN 60436 standard. 0 indicates a lousy cleaning, while 5 indicates a perfect cleaning. The know-how in the dishwasher cleaning technology states that 3.3 is accepted as a good value for customer expectations.

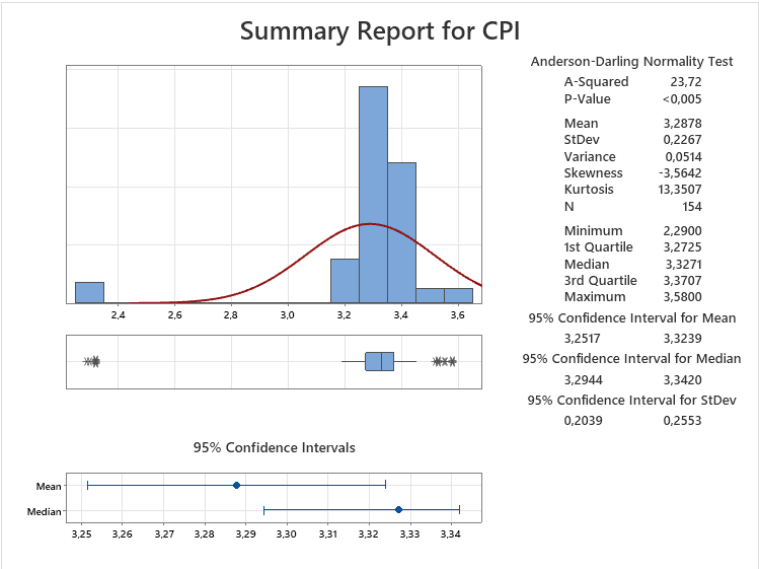


Figure 3.1: Statistical summary of CPI.

In the statistical summary of the CPI in Figure 3.1, the mean is 3.29, and the standard deviation is 0.23. The 75% of the CPI value is between 3.27 and 3.58, caused by

targeting 3.3. The CPI value less than 3.27 is only 25%; these programs are for slightly dirty dishes. According to Anderson Darling Normality Test [56], the CPI is not normally distributed and skewed. The skewness value is -3.56, which means that the tail is on the left side of the distribution. This means that the data is imbalanced, and therefore our regression model may suffer. The statistical summary also shows us that there is no CPI value greater than 3.6.

In the regression model, the acceptable MAE value can be decided from the statistical summary of CPI given in Figure 3.1. The range of the CPI value is 5, and according to the domain experts, a difference of 0.15 is not sensitive to customers. Additional data comes from the standard deviation of the CPI value in the repeated experiments, as it is 0.03. Since there is no ideal method to decide the acceptable MAE value, we can look for an MAE value smaller than 0.05 in the CPI prediction for acceptance.

3.2.2.2 Drying Performance Index (DPI)

The DPI value is the drying performance index, which is also taken from the results of the experiments that are done according to EN 60436. An evaluator gives a point between 0 and 100 to all unique dishes after the cleaning cycle ends, and 100 is the best drying performance.

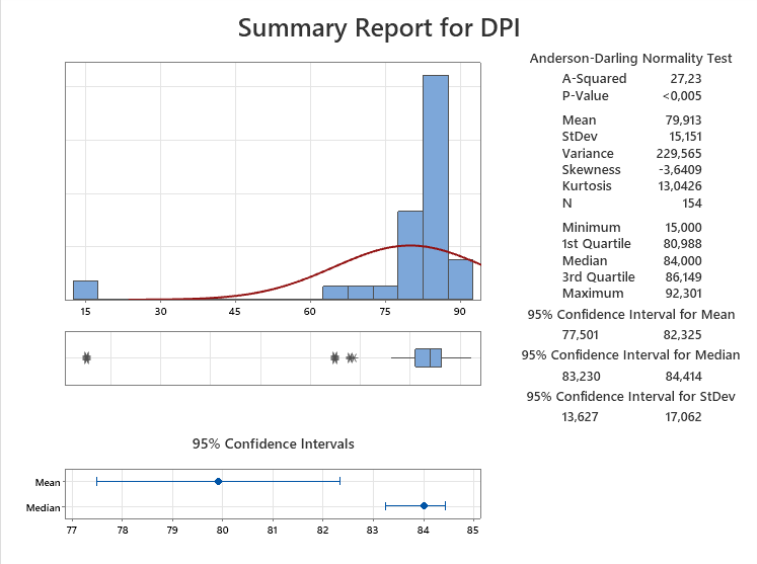


Figure 3.2: Statistical summary of DPI.

According to the know-how in dishwasher technology, 80 is accepted as a good value

for DPI. The analysis of the DPI values for 154 cleaning programs is given in Figure 3.2. The statistical summary of the DPI data's characteristics is very similar to the CPI data's characteristics. The mean of the data is 79.91, and the standard deviation is 15.75. The high standard deviation comes from the short-period programs that do not focus on drying. The 75% of the DPI value is higher than 81. The DPI is not normally distributed and skewed. The skewness value is -3.64, which means that the tail is on the left side of the distribution, like the CPI distribution.

The acceptable value for MAE in the DPI regression model can be studied according to the range of the target variable, which is 100. However, the distribution is between 15 to 92. According to domain expert, 5 can be sensitive to the customer. Moreover, the standard deviation of the repeated experiments in the DPI measurement is 1.45. Using these factors, the acceptable MAE value in the DPI prediction can be decided as 3.

3.2.2.3 Energy Consumption (EC), Water Consumption (WC), and Time Duration (TD)

The EC value is the dishwasher's energy consumption, the WC value is the dishwasher's water consumption, and the TD value is the dishwasher's operating time in the related cycle. The statistical summaries of the related values for 154 cleaning programs are given in Figure 3.3 for EC, and in Figure 3.4 for TD and WC. All

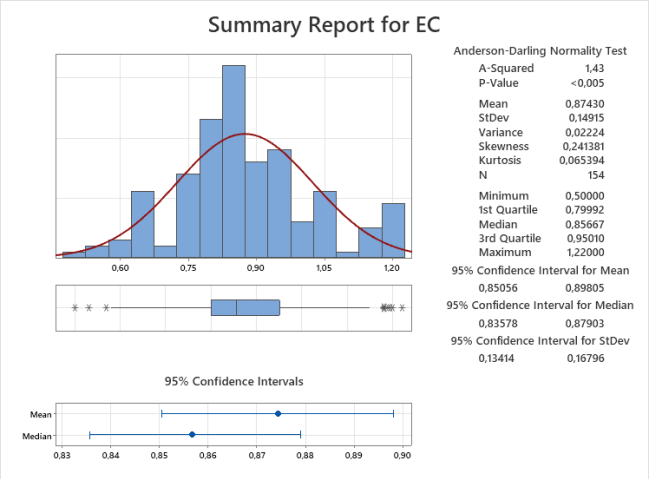


Figure 3.3: Statistical summary of EC.

distributions are close to normal according to the Andersen-Darling Normality Test [56].

The values of WC and TD can be easily calculated by summing up the related features of input data. Therefore, we expect a linear function for WC and TD models. The acceptable MAE value of the EC prediction model can be 0.07. Using the same idea up to now, from the previous studies and domain expert knowledge, the acceptable MAE values for WC and TD can be taken as 0.5 and 5, respectively.

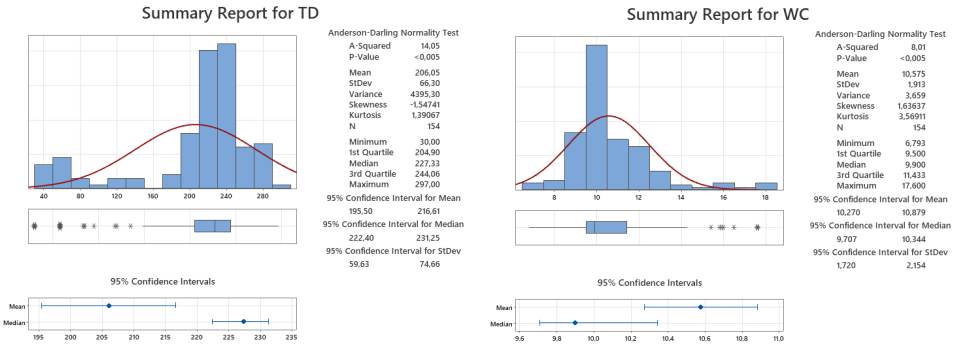


Figure 3.4: Statistical summary of TD (left) and WC (right).

As we figure out the acceptable MAE values, we also look at the R-squared value in the regression models for acceptance. In general, there is no single "acceptable" R-squared value for a regression prediction model, as it can depend on the specific context and application of the model. In general, a higher R-squared value generally indicates that the model is a better fit for the data and has more predictive power. On the other hand, R-squared (close to 1) does not necessarily mean that the model is the best (or optimal) fit for the data, as it could be overfitting the data and not generalizing well to new data. Therefore, it is also important to consider other metrics and diagnostics in addition to R-squared, such as residual plots, to assess the overall quality and validity of the regression model.

In our numerical simulations, we consider 0.70 as the minimum acceptable R-squared value; see Table 3.8 for the overall acceptable thresholds for the dependent variables.

Table 3.8: Acceptable maximum MAE and minimum R-squared values for related outputs.

Dependent Variable	**MAE**	R^2
CPI	0.05	0.70
DPI	3	0.70
EC	0.07	0.70
WC	0.5	0.70
TD	5	0.70

3.3 Data Analysis

We will continue with the data analysis according to the predictive modelling framework explained in Section 2.2. At the beginning, starting with DATASET 0, we have 154 samples of data. First, the data is checked for duplicate samples, and 60 samples are dropped, leaving 94 samples. Then, the independent variables are checked for low variance, and independent variables with a variance value less than 0.05 are dropped, left with 49 independent variables which is our first dataset, DATASET I; see Table 3.9.

The rank of the obtained data matrix is 47 which implies that there is a rank deficiency problem. Rank deficiency means that there is a collinearity problem in the matrix. The inverse of the matrix cannot be computed, making it impossible to obtain unique regression coefficients. One approach to overcome this issue is to remove one or more correlated variables. The independent variables are checked for the collinearity, and one of the independent variables with a Pearson correlation [13] higher than 0.9 among correlated independent variables are dropped, left with 37 independent variables. In general, the dropped column is chosen by a domain expert according to the model's interpretability. But right now, we will continue choosing the dropped column randomly. We note that the dropped variables are checked and approved by the domain expert. We end up with 94 samples, and the data have 5 dependent variables and 37 independent variables, which generates our second data set, named as DATASET II; see Table 3.9.

The properties of the DATASET II matrix can be summarized by the rank value of 37, the determinant of 6.14e+107, and the condition number 114393. There is no rank-deficiency problem in the input matrix right now. However, the condition number is

Table 3.9: Independent variables of DATASET I and DATASET II.

No	DATASET I	No	DATASET II	No	DATASET I	No	DATASET II
1	1PWRPM	1	1PWRPM	26	4CRPeriod	20	4CRPeriod
2	1PWTemperature	2	1PWTemperature	27	4CRTop_Spray_Circulation_Period	21	4CRTop_Spray_Circulation_Period
3	1PWLower_Spray_Circulation_Period	3	1PWLower_Spray_Circulation_Period	28	4CRUpper_Spray_Circulation_Period	22	4CRUpper_Spray_Circulation_Period
4	1PWWaiting	4	1PWWaiting	29	4CRWaterInlet	23	4CRWaterInlet
5	1PWCirculation_Period	5	1PWCirculation_Period	30	7RSRPM	24	7RSRPM
6	1PWClosed_Cycle_Period	6	1PWClosed_Cycle_Period	31	7RSTemperature	25	7RSTemperature
7	1PWPeriod	7	1PWPeriod	32	7RSLower_Spray_Circulation_Period	26	7RSLower_Spray_Circulation_Period
8	1PWTop_Spray_Circulation_Period	8	1PWTop_Spray_Circulation_Period	33	7RSWaiting		dropped
9	1PWUpper_Spray_Circulation_Period		dropped	34	7RSCirculation_Period		dropped
10	1PWWaterInlet	9	1PWWaterInlet	35	7RSClosed_Cycle_Period	27	7RSClosed_Cycle_Period
11	2MWRPM	10	2MWRPM	36	7RSPeriod	28	7RSPeriod
12	2MWTemperature	11	2MWTemperature	37	7RSTop_Spray_Circulation_Period		dropped
13	2MWLower_Spray_Circulation_Period	12	2MWLower_Spray_Circulation_Period	38	7RSUpper_Spray_Circulation_Period	29	7RSUpper_Spray_Circulation_Period
14	2MWCirculation_Period		dropped	39	7RSWaterInlet		dropped
15	2MWClosed_Cycle_Period	13	2MWClosed_Cycle_Period	40	8DSRPM	30	8DSRPM
16	2MWPeriod	14	2MWPeriod	41	8DSCirculation_Period		dropped
17	2MWTop_Spray_Circulation_Period	15	2MWTop_Spray_Circulation_Period	42	8DSClosed_Cycle_Period	31	8DSClosed_Cycle_Period
18	2MWUpper_Spray_Circulation_Period		dropped	43	8DSPeriod		dropped
19	2MWWaterInlet	16	2MWWaterInlet	44	10DYWaiting	32	10DYWaiting
20	3MFPeriod	17	3MFPeriod	45	10DYFan	33	10DYFan
21	3MFWaterInlet		dropped	46	10DYFan_Flap	34	10DYFan_Flap
22	4CRRPM	18	4CRRPM	47	10DYExtra_Heater_Offset	35	10DYExtra_Heater_Offset
23	4CRLower_Spray_Circulation_Period	19	4CRLower_Spray_Circulation_Period	48	10DYDoor_Openening	36	10DYDoor_Openening
24	4CRWaiting		dropped	49	10DYPeriod	37	10DYPeriod
25	4CRCirculation_Period		dropped				

relatively high, which means high sensitivity in fitted the parameters to the input data. This may emerge problems in the prediction process.

Now we first construct a standard linear regression (LR) model for all output by using DATASET I & II. Obtained results are provided in Table 3.10. The improvement of the prediction quality can be seen with the evaluation metrics in terms of MAE and R^2. These are calculated by repeated k-fold cross-validation with the number of splits being 5 and the number of repetitions being 3. Unfortunately, for CPI and DPI models, the R^2 values are negative, which means the models predict worse than the mean of the target values. In addition, the MAE values of CPI and DPI predictions are higher than the agreed acceptable values in Table 3.8 which are 0.05 and 3, respectively. The LR models with DATASET II are better however cannot be still acceptable. Insufficient data or nonlinearity of the problem can be reasons for poor prediction results obtained by the linear model. Therefore, we next try to improve the model by selecting important features to overcome the insufficient data problem.

Table 3.10: Results of linear regression model for DATASET I and II.

Dependent Variable	**MAE**	R^2
CPI	0.28	-8.62
DPI	14.25	-13.42
EC	0.16	-6.38
WC	1.39	-1.19
TD	10.38	0.85

(a) DATASET I

Dependent Variable	**MAE**	R^2
CPI	0.13	-0.41
DPI	8.58	-1.55
EC	0.07	0.61
WC	0.73	0.42
TD	6.43	0.97

(b) DATASET II

3.3.1 Improving Prediction Quality by Feature Selection

In general, we need enough samples in the prediction models for each independent variable. However, the number of independent variables and the sample size are very close in our case, and therefore we need to decrease the number of independent variables to solve the underlying problem. Feature selection methods introduced in Section 2.2.2, which are, select k-best with f-regression, sequential backward, and genetic algorithm, are applied to decrease the number of the features in the DATASET II. In Table 3.11, we give the obtained results by using a linear model after the application of the feature selection methods. We note that different features can be selected for each output and U indicates the total number of features belong to all outputs.

Table 3.11: Results of linear regression prediction with feature selection methods.

Method		**CPI**	**DPI**	**EC**	**WC**	**TD**	**U**
Select K Best f-regression	# Features	24	2	30	30	12	36
	MAE	0.09	7.48	0.06	0.56	3.41	
	R^2	0.44	-1.02	0.74	0.81	0.99	
Sequential Backward	# Features	19	18	9	8	5	31
	MAE	0.10	4.48	0.05	0.46	3.32	
	R^2	0.50	0.48	0.84	0.88	0.99	
Genetic Algorithm	# Features	18	20	19	17	9	34
	MAE	0.08	4.01	0.04	0.41	2.90	
	R^2	0.66	0.59	0.89	0.89	1.00	

Although select k-best with f-regression provides acceptable results for EC, TD, and WC, it is not applicable since the R-squared values for CPI and DPI are negative or not enough. Sequential backward feature selector makes improvement in both MAE and R-squared, however R-squared value of CPI and DPI are still low. Last, we apply a genetic algorithm for the feature selection, which yields the best results based on the linear regression model. However, MAE and R^2 values of CPI and DPI are still not enough and require improvement. The selected features with genetic algorithms belong to related output are listed with reference to DATASET II is given in Table 3.12. The eliminated variables by genetic algorithm is assigned with "not included" in Table 3.12.

Overall, the linear regression models provide reasonable results for the outputs EC, WC, and TD since the physical background of their calculations might be linear.

Table 3.12: Selected features by genetic algorithm for linear regression model on DATASET II.

DATASET II	CPI	DPI	EC	WC	TD
1PWRPM	not included	not included	not included	1PWRPM	not included
1PWTemperature	not included	not included	not included	1PWTemperature	not included
1PWLower_Spray_Circulation_Period	not included	not included	1PWLower_Spray_Circulation_Period	not included	not included
1PWWaiting	not included	not included	not included	not included	1PWWaiting
1PWCirculation_Period	not included	1PWCirculation_Period	1PWCirculation_Period	not included	not included
1PWClosed_Cycle_Period	not included	not included	1PWClosed_Cycle_Period	1PWClosed_Cycle_Period	not included
1PWPeriod	1PWPeriod	not included	1PWPeriod	not included	1PWPeriod
1PWTop_Spray_Circulation_Period	not included	1PWTop_Spray_Circulation_Period	not included	not included	not included
1PWWaterInlet	not included	1PWWaterInlet	1PWWaterInlet	1PWWaterInlet	not included
2MWRPM	2MWRPM	2MWRPM	not included	2MWRPM	not included
2MWTemperature	2MWTemperature	2MWTemperature	2MWTemperature	not included	not included
2MWLower_Spray_Circulation_Period	not included	not included	2MWLower_Spray_Circulation_Period	not included	not included
2MWClosed_Cycle_Period	2MWClosed_Cycle_Period	2MWClosed_Cycle_Period	2MWClosed_Cycle_Period	2MWClosed_Cycle_Period	2MWClosed_Cycle_Period
2MWPeriod	not included	2MWPeriod	2MWPeriod	not included	2MWPeriod
2MWTop_Spray_Circulation_Period	not included	2MWTop_Spray_Circulation_Period	not included	not included	not included
2MWWaterInlet	2MWWaterInlet	2MWWaterInlet	not included	2MWWaterInlet	not included
3MFPeriod	not included	not included	not included	not included	not included
4CRRPM	4CRRPM	not included	4CRRPM	not included	not included
4CRLower_Spray_Circulation_Period	not included	4CRLower_Spray_Circulation_Period	not included	not included	not included
4CRPeriod	not included	not included	4CRPeriod	4CRPeriod	not included
4CRTop_Spray_Circulation_Period	4CRTop_Spray_Circulation_Period	not included	4CRTop_Spray_Circulation_Period	4CRTop_Spray_Circulation_Period	not included
4CRUpper_Spray_Circulation_Period	4CRUpper_Spray_Circulation_Period	4CRUpper_Spray_Circulation_Period	4CRUpper_Spray_Circulation_Period	not included	not included
4CRWaterInlet	4CRWaterInlet	4CRWaterInlet	not included	4CRWaterInlet	4CRWaterInlet
7RSRPM	7RSRPM	7RSRPM	not included	7RSRPM	7RSRPM
7RSTemperature	7RSTemperature	7RSTemperature	7RSTemperature	not included	not included
7RSLower_Spray_Circulation_Period	7RSLower_Spray_Circulation_Period	7RSLower_Spray_Circulation_Period	7RSLower_Spray_Circulation_Period	7RSLower_Spray_Circulation_Period	not included
7RSClosed_Cycle_Period	7RSClosed_Cycle_Period	7RSClosed_Cycle_Period	7RSClosed_Cycle_Period	7RSClosed_Cycle_Period	not included
7RSPeriod	not included	not included	not included	7RSPeriod	7RSPeriod
7RSUpper_Spray_Circulation_Period	not included	not included	not included	not included	not included
8DSRPM	not included	not included	not included	not included	not included
8DSClosed_Cycle_Period	not included	not included	not included	8DSClosed_Cycle_Period	8DSClosed_Cycle_Period
10DYWaiting	10DYWaiting	10DYWaiting	10DYWaiting	not included	not included
10DYFan	10DYFan	10DYFan	10DYFan	not included	not included
10DYFan_Flap	10DYFan_Flap	not included	not included	not included	not included
10DYExtra_Heater_Offset	10DYExtra_Heater_Offset	not included	not included	10DYExtra_Heater_Offset	not included
10DYDoor_Openening	10DYDoor_Openening	10DYDoor_Openening	10DYDoor_Openening	not included	not included
10DYPeriod	not included	10DYPeriod	not included	10DYPeriod	10DYPeriod

While a perfect R-squared value for TD should be interpreted cautiously, however in our case the GA chooses the all necessary attributes and TD prediction becomes the sum of them. However, we think that the physical nature of CPI and DPI seems nonlinear. Therefore, we move to nonlinear approaches.

3.3.2 Improving Prediction Quality by Solving Nonlinearity

We aim to improve the prediction models of CPI and DPI by using nonlinear models like k-nearest neighbors with $k = 3$ (3-NN) and like tree-based models such as random forest regression (RFR), gradient boosting regression (GBR), and XGBoost. Obtained results from nonlinear models with DATASET II are given in Table 3.13.

There is a significant improvement for CPI and DPI predictions as expected due to solving the nonlinearity problem. In the CPI prediction, we are very close to our targets for MAE and R^2 values; see Table 3.8 for the target values. In DPI we have acceptable models. As being linear functions, WC and TD the prediction performance become worse in the non-linear models.

In the LR case, we have seen that feature selection with genetic algorithm has significant effect on the prediction. Now, Table 3.14 displays the number of selected

Table 3.13: Results of nonlinear regression predictions with DATASET II.

Method		**3-NN**	**RFR**	**GBR**	**XGB**
CPI	MAE	0.10	0.07	0.06	0.06
	R^2	0.06	0.66	0.64	0.64
DPI	MAE	4.93	3.11	2.89	2.92
	R^2	0.26	0.74	0.79	0.81
EC	MAE	0.07	0.05	0.05	0.04
	R^2	0.60	0.81	0.80	0.80
WC	MAE	0.70	0.63	0.65	0.59
	R^2	0.66	0.77	0.75	0.73
TD	MAE	10.64	7.42	7.32	7.66
	R^2	0.95	0.97	0.98	0.97

features obtained by a genetic algorithm and the values of MAE and R^2 for our outputs in the context of nonlinear regression predictor. It is noted that U indicates the total number of features.

Table 3.14: Results of nonlinear regression predictions with genetic feature selection on DATASET II.

Method		**3-NN**	**RFR**	**GBR**	**XGB**
CPI	# Features	16	8	10	15
	MAE	0.05	0.05	0.05	0.05
	R^2	0.83	0.83	0.85	0.84
DPI	# Features	15	12	7	4
	MAE	2.11	2.56	2.18	2.15
	R^2	0.85	0.79	0.81	0.78
EC	# Features	10	14	8	5
	MAE	0.04	0.05	0.04	0.04
	R^2	0.90	0.84	0.89	0.87
WC	# Features	11	8	9	14
	MAE	0.57	0.54	0.47	0.52
	R^2	0.80	0.83	0.84	0.80
TD	# Features	16	11	9	10
	MAE	5.50	5.38	5.10	5.14
	R^2	0.98	0.99	0.99	0.98
U	# Features	30	28	27	31

The non-linear regression model GBR with GA feature selection achieves the best performance among all models for CPI and DPI. Also for EC, 3-NN model with genetic algorithm feature selection achieves the best performance with minimum number of features.

Table 3.15: Selected prediction models and number of features for each output.

Dependent Variable	**# of Features**	**Model**	R^2	**MAE**
CPI	10	GBR w GA	0.85	0.05
DPI	7	GBR w GA	0.81	2.18
EC	10	3-NN w GA	0.90	0.04
WC	17	LR w GA	0.89	0.41
TD	9	LR w GA	1.00	2.90
U	30			

Table 3.16: Selected features by genetic algorithm for regression models with best regression performances on DATASET II.

DATASET II	**CPI with GBR**	**DPI with GBR**	**EC with 3-NN**	**WC with LR**	**TD with LR**
1PWRPM	not included	not included	1PWRPM	1PWRPM	not included
1PWTemperature	1PWTemperature	1PWTemperature	not included	1PWTemperature	not included
1PWLower_Spray_Circulation_Period	not included	not included	not included	not included	not included
1PWWaiting	1PWWaiting	not included	not included	not included	1PWWaiting
1PWCirculation_Period	not included	not included	not included	not included	not included
1PWClosed_Cycle_Period	not included	not included	not included	1PWClosed_Cycle_Period	not included
1PWPeriod	not included	not included	1PWPeriod	not included	1PWPeriod
1PWTop_Spray_Circulation_Period	not included	not included	not included	not included	not included
1PWWaterInlet	not included	not included	not included	1PWWaterInlet	not included
2MWRPM	2MWRPM	not included	not included	2MWRPM	not included
2MWTemperature	not included	not included	not included	not included	not included
2MWLower_Spray_Circulation_Period	not included	not included	2MWLower_Spray_Circulation_Period	not included	not included
2MWClosed_Cycle_Period	not included	not included	not included	2MWClosed_Cycle_Period	2MWClosed_Cycle_Period
2MWPeriod	2MWPeriod	2MWPeriod	2MWPeriod	not included	2MWPeriod
2MWTop_Spray_Circulation_Period	not included	2MWTop_Spray_Circulation_Period	2MWTop_Spray_Circulation_Period	not included	not included
2MWWaterInlet	not included	not included	not included	2MWWaterInlet	not included
3MFPeriod	3MFPeriod	not included	not included	not included	not included
4CRRPM	4CRRPM	not included	not included	not included	not included
4CRLower_Spray_Circulation_Period	not included	not included	not included	not included	not included
4CRPeriod	4CRPeriod	not included	not included	4CRPeriod	not included
4CRTop_Spray_Circulation_Period	4CRTop_Spray_Circulation_Period	not included	not included	4CRTop_Spray_Circulation_Period	not included
4CRUpper_Spray_Circulation_Period	not included	not included	not included	not included	not included
4CRWaterInlet	not included	not included	not included	4CRWaterInlet	4CRWaterInlet
7RSRPM	7RSRPM	not included	not included	7RSRPM	7RSRPM
7RSTemperature	not included	not included	7RSTemperature	not included	not included
7RSLower_Spray_Circulation_Period	not included	not included	not included	7RSLower_Spray_Circulation_Period	not included
7RSClosed_Cycle_Period	not included	not included	not included	7RSClosed_Cycle_Period	not included
7RSPeriod	not included	not included	not included	7RSPeriod	7RSPeriod
7RSUpper_Spray_Circulation_Period	not included	not included	7RSUpper_Spray_Circulation_Period	not included	not included
8DSRPM	8DSRPM	not included	not included	not included	not included
8DSClosed_Cycle_Period	not included	8DSClosed_Cycle_Period	8DSClosed_Cycle_Period	8DSClosed_Cycle_Period	8DSClosed_Cycle_Period
10DYWaiting	not included	not included	10DYWaiting	not included	not included
10DYFan	not included	10DYFan	not included	not included	not included
10DYFan_Flap	not included	10DYFan_Flap	not included	not included	not included
10DYExtra_Heater_Offset	not included	10DYExtra_Heater_Offset	10DYExtra_Heater_Offset	10DYExtra_Heater_Offset	not included
10DYDoor_Openening	not included	not included	not included	not included	not included
10DYPeriod	not included	not included	not included	10DYPeriod	10DYPeriod

In this problem, we see the effect of feature selection in the regression performance clearly. WC and TD cannot achieve an increase in prediction performance by non-linear models since they are linear functions in nature. The best regression models for WC and TD are linear regression with genetic algorithm feature selection. Overall, the prediction models yielding the best performance in terms of MAE, R-squared, or minimum feature for each dependent variable are given in Table 3.15 . Corresponding features are displayed in Table 3.16. The eliminated variables by genetic algorithm are assigned with "not included" in the Table 3.16. Further, we note that genetic algorithm can select different features for different runs with acceptable results. Differentiation in the selected features across different runs of a genetic algorithm can be attributed to the random initialization, selection pressure, genetic operators, eval-

uation criteria, convergence and exploration dynamics, and the algorithm's inherent stochasticity.

3.4 Verification of the Models

In the previous section, we have developed prediction models based on the experiments. WC and TD outputs are formulated in a linear way, whereas CPI, DPI, and EC are considered in a nonlinear structure. Models are accepted using the metrics MAE, R-squared, and number of features. The acceptance limits of the MAE are also determined by the experimental results and the domain expert's knowledge; see Table 3.8. Moreover, all the tests are done with k-fold cross validation methods.

In the verification step, the whole dataset of the dishwasher experimental data are predicted by the related models, and the corresponding error distributions are analyzed; see Figure 3.5 for EC, WC, and TD and see Figure 3.7 for CPI and DPI.

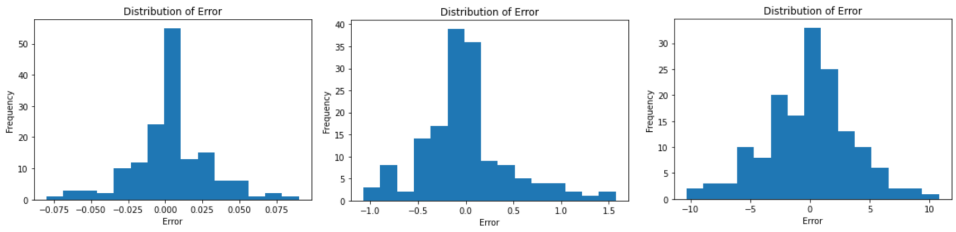


Figure 3.5: Error distributions of EC, WC, and TD models from left to right.

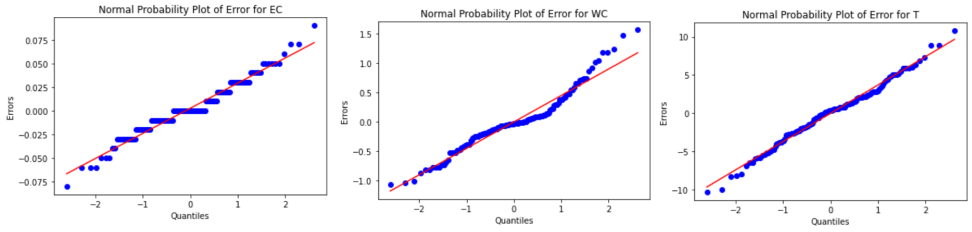


Figure 3.6: Probability graph of EC, WC, and TD models' errors from left to right.

Results show that the error distributions of the modeled outputs are in acceptable level since the mean of the error is close to zero with a close to normal distribution even some outliers exist. Probability graphs of the errors having a normal distribution are also displayed in Figure 3.6 and Figure 3.8. When we fit a regression model to a dataset, we assume that the residuals are normally distributed with a mean of zero

and constant variance. If this assumption is not met, it can indicate that the model is misspecified or that there are outliers in the data that are affecting the model's performance. When we analyse the outliers of CPI, we detect that the outliers are due to the physical experiment errors.

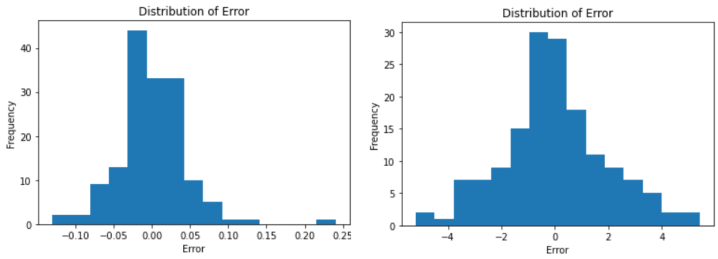


Figure 3.7: Error distributions of CPI and DPI models from left to right.

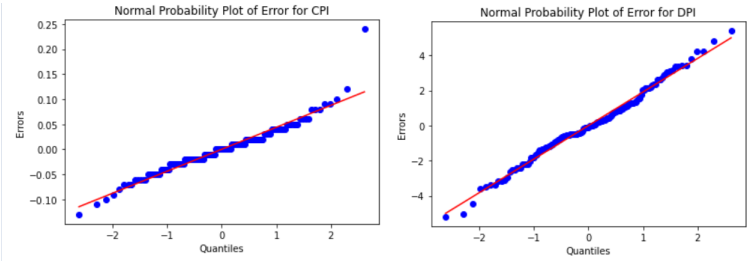


Figure 3.8: Probability graph of CPI and DPI models' errors from left to right.

Next, we extend the prediction models into the real-life case studies.

3.5 Digital Twin of Dishwasher Performance Laboratory

This scenario aims on predicting the outcomes of a new (unseen) dishwasher cleaning cycle program without actually conducting laboratory experiments. This is a smart and cost-effective approach since designing a new cleaning cycle requires a lot of trial and error-based experiments, which can be time-consuming and expensive. To better understand the steps involved in the trial and error-based design; see Figure 3.9. By developing the prediction models, the designers can use them as a digital twin of the performance laboratory, which makes the whole process more efficient.

With the advent of digital twin laboratory technology, designers can now take advantage of the trial-and-error method without the need for traditional physical testing. This has led to a significant reduction in the time to market period, as well as more

efficient use of performance laboratories. In fact, up to 40% fewer tests are required for new program designs in R&D department, resulting in up to 50% reductions in time to market for new cleaning cycle designs. This has also led to a significant reduction in the cost of experiments, with up to 50% savings. With these advancements, designers can now confidently develop and test new products with greater efficiency and accuracy, resulting in improved performance and greater customer satisfaction.

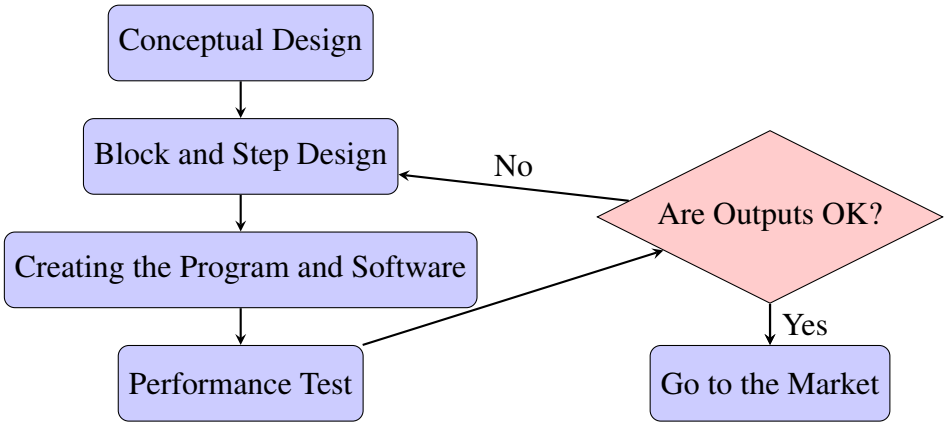


Figure 3.9: Design steps of dishwasher program.

Table 3.17: Results of test data obtained by digital twin performance laboratory using selected features and models of CPI and DPI.

No	CPI_e	CPI_p	Err % CPI	DPI_e	DPI_p	Err % DPI
1	3.27	3.34	2.14%	83.23	83.25	0.02%
2	3.27	3.32	1.53%	65	68.53	5.43%
3	2.32	2.33	0.43%	15	15.46	3.07%
4	3.53	3.4	3.68%	84	84.2	0.24%
5	3.33	3.31	0.60%	91.67	87.01	5.08%
6	3.27	3.32	1.53%	65	68.53	5.43%
7	3.34	3.33	0.30%	85.39	84.38	1.18%
8	3.29	3.32	0.91%	85.67	87.48	2.11%
9	3.3	3.32	0.61%	84.11	86.42	2.75%

We select models with best regression performance and minimum number of features in Section 3.4, based on the results in Table 3.15. We design the digital twin laboratory by using the GBR model with selected 10 features for CPI, and selected 7 features for DPI; by using 3-NN model with selected 10 features for EC; by using LR with selected 9 features for TD and selected 17 features for WC.

Additional to MAE and R^2 values of the selected models and features, the digital twin laboratory is tested by using unseen test data. The experimental results of the digital twin laboratory for the test data are given in Table 3.17 for CPI and DPI and in Table 3.18 for EC, TD, and WC. We note that the subscript p denotes the predicted values, whereas the subscript e corresponds to the physical experimental results.

Table 3.18: Results of test data obtained by digital twin performance laboratory using selected features and models of EC, WC, and TD.

No	EC_e	EC_p	Err% EC	TD_e	TD_p	Err% TD	WC_e	WC_p	Err% WC
1	0.88	0.9	2.27%	240	242	0.83%	8.75	9.57	9.37%
2	1.06	1.02	3.77%	58	56	3.45%	10.6	10.53	0.66%
3	0.64	0.65	1.56%	30	32	6.67%	10.8	10.6	1.85%
4	1.18	1.16	1.69%	148	147	0.68%	17.6	17.69	0.51%
5	0.95	0.92	3.16%	210	218	3.81%	9.68	9.67	0.10%
6	1.06	1.03	2.83%	58	56	3.45%	10.8	10.53	2.50%
7	0.83	0.85	2.41%	181	188	3.87%	11.44	11.56	1.05%
8	0.94	0.97	3.19%	215	218	1.40%	11.59	11.29	2.59%
9	1.05	0.98	6.67%	226	223	1.33%	12.11	11.28	6.85%

The acceptable levels for the prediction performance of unique tests are determined by the sensitivity level of the customer that is explained in Section 3.2.2 and the allowable error percentage of the test institutes. The level can be 10% for EC, WC, and TD, whereas it can be 6% for CPI and DPI. Under these regulations all the results in Table 3.17 and Table 3.18 seem acceptable.

During the design of the digital twin laboratory, another strategies can be using the features selected by domain expert and all features in the dataset (DATASET II); see Table 3.19 for using features in the data sets. All simulations are also proceed by following the same selected models. The results are compared at Table 3.20. As expected, the regression performance of the same models are better with feature selection methods.

As being consistent, the test results of the domain expert features are given at Table 3.21 and Table 3.22 and the test results of the all feature dataset are given at Table 3.23 and Table 3.24.

Table 3.19: Selected features in DATASET II, domain expert dataset, and the dataset producing best regression performance.

UNION ALL	DATASET II	DOMAIN EXPERT DATASET	BEST REGRESSION PERFORMANCE
1PWCirculation_Period	1PWCirculation_Period	1PWCirculation_Period	not included
1PWClosed_Cycle_Period	1PWClosed_Cycle_Period	not included	1PWClosed_Cycle_Period
1PWLower_Spray_Circulation_Period	1PWLower_Spray_Circulation_Period	1PWLower_Spray_Circulation_Period	not included
1PWPeriod	1PWPeriod	1PWPeriod	1PWPeriod
1PWRPM	1PWRPM	1PWRPM	1PWRPM
1PWTemperature	1PWTemperature	1PWTemperature	1PWTemperature
1PWTop_Spray_Circulation_Period	1PWTop_Spray_Circulation_Period	1PWTop_Spray_Circulation_Period	not included
1PWUpper_Spray_Circulation_Period	not included	1PWUpper_Spray_Circulation_Period	not included
1PWWaiting	1PWWaiting	1PWWaiting	1PWWaiting
1PWWaterInlet	1PWWaterInlet	1PWWaterInlet	1PWWaterInlet
2MWCirculation_Period	not included	2MWCirculation_Period	not included
2MWClosed_Cycle_Period	2MWClosed_Cycle_Period	not included	2MWClosed_Cycle_Period
2MWLower_Spray_Circulation_Period	2MWLower_Spray_Circulation_Period	2MWLower_Spray_Circulation_Period	2MWLower_Spray_Circulation_Period
2MWPeriod	2MWPeriod	2MWPeriod	2MWPeriod
2MWRPM	2MWRPM	2MWRPM	2MWRPM
2MWTemperature	2MWTemperature	2MWTemperature	not included
2MWTop_Spray_Circulation_Period	2MWTop_Spray_Circulation_Period	2MWTop_Spray_Circulation_Period	2MWTop_Spray_Circulation_Period
2MWUpper_Spray_Circulation_Period	not included	2MWUpper_Spray_Circulation_Period	not included
2MWWaterInlet	2MWWaterInlet	2MWWaterInlet	2MWWaterInlet
3MFPeriod	3MFPeriod	not included	3MFPeriod
4CRCirculation_Period	not included	4CRCirculation_Period	not included
4CRLower_Spray_Circulation_Period	4CRLower_Spray_Circulation_Period	4CRLower_Spray_Circulation_Period	not included
4CRPeriod	4CRPeriod	4CRPeriod	4CRPeriod
4CRRPM	4CRRPM	4CRRPM	4CRRPM
4CRTop_Spray_Circulation_Period	4CRTop_Spray_Circulation_Period	4CRTop_Spray_Circulation_Period	4CRTop_Spray_Circulation_Period
4CRUpper_Spray_Circulation_Period	4CRUpper_Spray_Circulation_Period	4CRUpper_Spray_Circulation_Period	not included
4CRWaterInlet	4CRWaterInlet	4CRWaterInlet	4CRWaterInlet
7RSCirculation_Period	not included	7RSCirculation_Period	not included
7RSClosed_Cycle_Period	7RSClosed_Cycle_Period	7RSClosed_Cycle_Period	7RSClosed_Cycle_Period
7RSLower_Spray_Circulation_Period	7RSLower_Spray_Circulation_Period	7RSLower_Spray_Circulation_Period	7RSLower_Spray_Circulation_Period
7RSPeriod	7RSPeriod	7RSPeriod	7RSPeriod
7RSRPM	7RSRPM	7RSRPM	7RSRPM
7RSTemperature	7RSTemperature	7RSTemperature	7RSTemperature
7RSTop_Spray_Circulation_Period	not included	7RSTop_Spray_Circulation_Period	not included
7RSUpper_Spray_Circulation_Period	7RSUpper_Spray_Circulation_Period	7RSUpper_Spray_Circulation_Period	7RSUpper_Spray_Circulation_Period
7RSWaterInlet	not included	7RSWaterInlet	not included
8DSClosed_Cycle_Period	8DSClosed_Cycle_Period	not included	8DSClosed_Cycle_Period
8DSRPM	8DSRPM	not included	8DSRPM
10DYDoor_Openening	10DYDoor_Openening	10DYDoor_Openening	not included
10DYExtra_Heater_Offset	10DYExtra_Heater_Offset	not included	10DYExtra_Heater_Offset
10DYFan	10DYFan	10DYFan	10DYFan
10DYFan_Flap	10DYFan_Flap	10DYFan_Flap	10DYFan_Flap
10DYPeriod	10DYPeriod	10DYPeriod	10DYPeriod
10DYWaiting	10DYWaiting	10DYWaiting	10DYWaiting

3.6 Designing Dishwasher Cleaning Cycle with Targeted Outputs

In the R&D department, one of the main goals of the designer is to create a new, efficient dishwasher cycle. In this scenario, the designer is trying to achieve CPI values of 3.65 and 3.80 for an intensive program. To do this, the designer will need to manipulate the steps of the cleaning cycle, but there is no clear idea or guess about which steps to manipulate.

Determining the intensive cleaning cycle steps for a cleaning cycle with 3.65 and 3.80 CPI will require the expertise of domain experts. They may have insights into which steps are most effective at achieving these values. Once the designer achieves an increase in CPI, they will need to determine the new values of DPI and EC. Overall, creating a new dishwasher cycle can be a complex process, and it needs to optimize

Table 3.20: Comparison of the regression models' results with features selected by domain expert, features selected by genetic algorithm, and all features.

Dependent Variable	**MAE**	R^2
CPI	0.049	0.75
DPI	2.42	0.74
EC	0.033	0.88
WC	0.73	-1.010
TD	4.97	0.95

(a) Domain expert features

Dependent Variable	**MAE**	R^2
CPI	0.044	0.84
DPI	2.14	0.80
EC	0.032	0.89
WC	0.37	0.91
TD	2.99	0.99

(b) Genetic algorithm features

Dependent Variable	**MAE**	R^2
CPI	0.049	0.76
DPI	2.50	0.72
EC	0.033	0.88
WC	0.78	-0.29
TD	6.32	0.88

(c) All features

Table 3.21: Results of test data obtained by digital twin performance laboratory using domain expert features and models of CPI and DPI.

No	**CPI_e**	**CPI_p**	**Err % CPI**	**DPI_e**	**DPI_p**	**Err % DPI**
1	3.27	3.33	1.83%	83.23	83.09	0.17%
2	3.27	3.32	1.53%	65	68.48	5.35%
3	2.32	2.32	0.00%	15	15.34	2.27%
4	3.53	3.44	2.55%	84	83.44	0.67%
5	3.33	3.33	0.00%	91.67	87.23	4.84%
6	3.27	3.32	1.53%	65	68.48	5.35%
7	3.34	3.31	0.90%	85.39	84.22	1.37%
8	3.29	3.32	0.91%	85.67	87.7	2.37%
9	3.3	3.34	1.21%	84.11	88.26	4.93%

multiple and conflicting objectives.

A brute force solution is to incrementally change the values of the independent variables of the known intensive program within limits. This is a search to find the best cleaning cycles among millions of created programs. The steps of the case study can be given as follows:

- A designer decides the independent variables that can be manipulated.

Table 3.22: Results of test data obtained by digital twin performance laboratory using domain expert features and models of EC, WC, and TD.

No	EC_e	EC_p	Err% EC	TD_e	TD_p	Err% TD	WC_e	WC_p	Err% WC
1	0.88	0.9	2.27%	240	241	0.42%	8.75	9.63	10.06%
2	1.06	1.13	6.60%	58	56	3.45%	10.6	10.73	1.23%
3	0.64	0.64	0.00%	30	29	3.33%	10.8	10.68	1.11%
4	1.18	1.17	0.85%	148	147	0.68%	17.6	18.11	2.90%
5	0.95	0.93	2.11%	210	217	3.33%	9.68	9.51	1.76%
6	1.06	1.13	6.60%	58	56	3.45%	10.8	10.73	0.65%
7	0.83	0.83	0.00%	181	187	3.31%	11.44	11.45	0.09%
8	0.94	0.96	2.13%	215	218	1.40%	11.59	11.45	1.21%
9	1.05	1.06	0.95%	226	225	0.44%	12.11	11.51	4.95%

Table 3.23: Results of test data obtained by digital twin performance laboratory using all features and models of CPI and DPI.

No	CPI_e	CPI_p	Err % CPI	DPI_e	DPI_p	Err % DPI
1	3.27	3.34	2.14%	83.23	83.1	0.16%
2	3.27	3.33	1.83%	65	68.54	5.45%
3	2.32	2.32	0.00%	15	15.43	2.87%
4	3.53	3.45	2.27%	84	83.08	1.10%
5	3.33	3.33	0.00%	91.67	87.22	4.85%
6	3.27	3.33	1.83%	65	68.54	5.45%
7	3.34	3.31	0.90%	85.39	83.96	1.67%
8	3.29	3.32	0.91%	85.67	87.8	2.49%
9	3.3	3.34	1.21%	84.11	88.18	4.84%

Table 3.24: Results of test data obtained by digital twin performance laboratory using all features and models of EC, WC, and TD.

No	EC_e	EC_p	Err% EC	TD_e	TD_p	Err% TD	WC_e	WC_p	Err% WC
1	0.88	0.9	2.27%	240	241	0.42%	8.75	9.72	11.09%
2	1.06	1.13	6.60%	58	55	5.17%	10.6	10.77	1.60%
3	0.64	0.64	0.00%	30	31	3.33%	10.8	10.67	1.20%
4	1.18	1.17	0.85%	148	143	3.38%	17.6	17.21	2.22%
5	0.95	0.94	1.05%	210	217	3.33%	9.68	9.64	0.41%
6	1.06	1.13	6.60%	58	55	5.17%	10.8	10.77	0.28%
7	0.83	0.83	0.00%	181	187	3.31%	11.44	11.55	0.96%
8	0.94	0.96	2.13%	215	218	1.40%	11.59	11.53	0.52%
9	1.05	1.06	0.95%	226	220	2.65%	12.11	10.97	9.41%

- A prediction model is run by using these independent variables.
- If the model can be accepted according to the values MAE and R^2, the designer creates the artificial program cycles by manipulating the independent variables in a significant derivation.
- Prediction model predicts the desired output values for all cycles.
- The designer selects the new cleaning cycle according to the targeted outputs from the predicted output values.

In our case study, we are looking for the CPI value in an intensive program. The designer's choice for the independent variables is: "1PWWaterInlet, 2MWWaterInlet, 2MWTemperature, 4CRWaterInlet, 7RSWaterInlet, 7RSTemperature, 1PWRPM, 2MWRPM, 4CRRPM, 7RSRPM". The XGBoost regression model prediction performance for these variables is MAE 0.078 and R^2 50. The model is not best however can be used. Then the designer creates 1048575 new programs by incrementally changing the independent variables within limits. The critical point of the prediction model is that we can not perform classical experiments to 1 million programs. However by using the digital twin laboratory discussed in Section 3.5 we can make predictions.

Table 3.25: CPI and DPI predictions of the 1 million cleaning cycles.

(a) CPI prediction

Item	Value
count	1048575
mean	3.51
std	0.0705
min	3.32
25%	3.46
50%	3.51
75%	3.56
max	3.69

(b) DPI prediction

Item	Value
count	1048575
mean	85
std	1.73
min	81
25%	84
50%	85
75%	87
max	90

From the results in Table 3.25, we can decide that it is impossible to get a CPI value of 3.80 by these manipulations. However, the 3.65 CPI value is still ok, and after filtering the related cycles that can be accepted as CPI value 3.65, we can now pre-

dict other output values. The XGBoost regression model prediction performance for DPI is MAE 4.34 and R^2 60; see Table 3.25b. Results of other outputs are given in Table 3.26.

Table 3.26: WC, EC, and TD predictions of the 1 million cleaning cycles.

Item	**WC Value**	**EC Value**	**TD Value**
count	1048575	1048575	1048575
mean	16.2	1.11	111
std	0.5	0.05	9.5
min	15.1	1.01	96
25%	15.8	1.07	105
50%	16.2	1.11	108
75%	16.5	1.15	120
max	17.5	1.24	133

The designer can choose the cleaning cycles from the predicted values of CPI, DPI, EC, TD, and WC according to preferences and use the independent variables as a new cleaning cycle. In Table 3.27, three cleaning cycles producing CPI > 3.6 and DPI > 86 are given.

Table 3.27: Selected features for program with CPI > 3.6 and DPI > 86.

Number of Cycles	1PWWaterInlet	2MWWaterInlet	2MWTemperature	4CRWaterInlet	7RSWaterInlet	7RSTemperature	1PWRPM	2MWRPM	4CRRPM	7RSRPM
743144	4.6	5	68	4.2	4.2	60	3000	2800	2800	3400
558834	4.6	4.6	68	4.2	4.6	60	3000	2800	2800	3400
927474	4.6	5.4	68	4.2	4.6	60	3000	2800	2800	3400

The disadvantage of the methodology is the number of possible cleaning cycles can be huge. In the digital twin case study, if there are 162 independent variables, and if we have only 3 alternative values for all independent variables, we need 3^{162} cleaning cycles means impossible to create and predict. This methodology can be applicable only with a limited number of independent variables, like in our case. To solve this issue, we will construct a multi-objective optimization problem in the next chapter.

CHAPTER 4

MULTI-OBJECTIVE OPTIMIZATION OF A DISHWASHER CLEANING CYCLE

In this chapter, we design a dishwasher cleaning cycle by solving a multi-objective optimization problem. First, we define a multi-objective optimization problem using the regression models discussed in Chapter 3 as objective functions and the corresponding features as decision variables. After, we use evolutionary algorithms such as non-dominated sorting genetic algorithm III (NSGA-III), constrained two-archive evolutionary algorithm (C-TAEA), and reference vector guided evolutionary algorithm (RVEA) to solve the underlying optimization problem. Lastly, to decide optimal solution, the user preference is in charge by weighting the objective functions.

4.1 Multi-Objective Optimization Problem (MOOP)

The dishwasher cleaning cycle design problem is:

$$\begin{aligned} &\min_{x} f(x) = [-CPI(x), -DPI(x), EC(x), TD(x), WC(x)]^T, \\ &\text{s.t.} \quad x_i^{\ell} \leq x_i \leq x_i^{u}, \quad i = 1, 2, ..., n, \end{aligned}$$

in where $x \in R^n$ is formed by n decision (independent) variables. The constraint set is called variable bounds, restricting each decision variable x_i to take a value within a lower x_i^{ℓ} and an upper x_i^{u} bound. These bounds constitute a decision variable space.

$CPI(x)$, $DPI(x)$, $EC(x)$, $WC(x)$, and $TD(x)$ are objective functions obtained from the statistical learning problem that has been discussed in Chapter 3. The $CPI(x)$ and $DPI(x)$ are maximized, whereas the $EC(x), WC(x)$, and $TD(x)$ are

minimized due to the nature of the problem. The objective functions correspond to the regression models that are selected according to the best prediction performance obtained in Chapter 3; see Table 4.1. Since the total number of features is 30, the solution vector x belongs to $\mathbb{R}^{30}$. In addition, Table 4.2 shows the list of independent variables.

Table 4.1: Selected regression models and number of features for each objective function.

Dependent Variable	**# of Features**	**Model**	R^2	**MAE**
CPI	10	GBR w GA	0.85	0.05
DPI	7	GBR w GA	0.81	2.18
EC	10	3-NN w GA	0.90	0.04
WC	17	LR w GA	0.89	0.41
TD	9	LR w GA	1.00	2.90
U	30			

Table 4.2: Decision variables of objective functions.

Variable No	**Union of Features**	**CPI with GBR**	**DPI with GBR**	**EC with 3-NN**	**WC with LR**	**TD with LR**
x_0	1PWRPM			1PWRPM	1PWRPM	
x_1	1PWTemperature	1PWTemperature	1PWTemperature		1PWTemperature	
x_3	1PWWaiting	1PWWaiting				1PWWaiting
x_5	1PWClosed_Cycle_Period				1PWClosed_Cycle_Period	
x_6	1PWPeriod			1PWPeriod		1PWPeriod
x_8	1PWWaterInlet				1PWWaterInlet	
x_9	2MWRPM	2MWRPM			2MWRPM	
x_{11}	2MWLower_Spray_Circulation_Period			2MWLower_Spray_Circulation_Period		
x_{12}	2MWClosed_Cycle_Period				2MWClosed_Cycle_Period	2MWClosed_Cycle_Period
x_{13}	2MWPeriod	2MWPeriod	2MWPeriod	2MWPeriod		2MWPeriod
x_{14}	2MWTop_Spray_Circulation_Period		2MWTop_Spray_Circulation_Period	2MWTop_Spray_Circulation_Period		
x_{15}	2MWWaterInlet				2MWWaterInlet	
x_{16}	3MFPeriod	3MFPeriod				
x_{17}	4CRRPM	4CRRPM				
x_{19}	4CRPeriod	4CRPeriod			4CRPeriod	
x_{20}	4CRTop_Spray_Circulation_Period	4CRTop_Spray_Circulation_Period			4CRTop_Spray_Circulation_Period	
x_{22}	4CRWaterInlet				4CRWaterInlet	4CRWaterInlet
x_{23}	7RSRPM	7RSRPM			7RSRPM	7RSRPM
x_{24}	7RSTemperature			7RSTemperature		
x_{25}	7RSLower_Spray_Circulation_Period				7RSLower_Spray_Circulation_Period	
x_{26}	7RSClosed_Cycle_Period				7RSClosed_Cycle_Period	
x_{27}	7RSPeriod				7RSPeriod	7RSPeriod
x_{28}	7RSUpper_Spray_Circulation_Period			7RSUpper_Spray_Circulation_Period		
x_{29}	8DSRPM	8DSRPM				
x_{30}	8DSClosed_Cycle_Period		8DSClosed_Cycle_Period	8DSClosed_Cycle_Period	8DSClosed_Cycle_Period	8DSClosed_Cycle_Period
x_{31}	10DYWaiting			10DYWaiting		
x_{32}	10DYFan		10DYFan			
x_{33}	10DYFan_Flap		10DYFan_Flap			
x_{34}	10DYExtra_Heater_Offset		10DYExtra_Heater_Offset	10DYExtra_Heater_Offset	10DYExtra_Heater_Offset	
x_{36}	10DYPeriod				10DYPeriod	10DYPeriod

Next, we continue with solving the MOOP of best regression performances objective functions.

4.2 Designing Dishwasher Cleaning Cycles by Solving MOOP

In this case study, we try to solve the MOOP to obtain the independent variables that compose the dishwasher cleaning cycle by using the evolutionary algorithms outlined

in Section 2.3.2.

Dishwasher's R&D experts traditionally design dishwasher cleaning cycles through trial and error experiments based on only their knowledge. In this case study, we will design a more energy-efficient ECO program by solving a MOOP. The problem can be defined as designing a new eco program with a lower EC value. Improving the EC value by just 5% leads us to save 0.25 billion kWh of electricity annually worldwide.

Here we use the most used eco program, which is a 16-place setting, built-in, 60cm dishwasher eco program with code DWECO051. The program EC value is 0.75 and aims to be between 0.6 and 0.7 kWh per cycle, while CPI and DPI should be a minimum of 3.2 and 80, respectively; see Table 4.3. There are no strict limitations on TD and WC in the ECO program; however, not to go extreme points, the targets of the new cleaning cycle TD and WC will be the same as the base cleaning cycle. In addition, the decision variable set, which is displayed in Table 4.4, has been determined by gathering the objective functions' independent variables.

Table 4.3: Target values of new design program with respect to original outputs of base program DWECO051.

Objective	**Experimental Value**	**Target Value**
CPI	3.28	minimum 3.2
DPI	86	minimum 80
EC	0.75	between 0.6 & 0.7
WC	9.40	maximum 10
TD	229	maximum 300

Initially, the decision variables have typical values from the previously designed DWECO051 base program. To determine the feasible set, we set the lower limits of the decision variables x_i^{ℓ} at 0.9 of the base program standard values and the upper limits x_i^{u} at 1.1; see at Table 4.4.

In general, to find the values of the decision variables to achieve EC between 0.6-0.7, CPI as 3.2, and DPI as 80, we first use evolutionary algorithms, which are producing multiple non-dominated points close to the Pareto-optimal front as possible, with a wide trade-off among objectives. Then, the algorithm choosing one of the obtained points using higher-level information, that is, the weights of the importance of the

objective functions, is applied.

To solve the underlying MOOP, we apply different algorithms: NSGA-III, which is an improvement of NSGA-II developed for multi-objective optimization problems with more than two objectives, C-TAEA that is an algorithm with a more sophisticated constraint-handling for many (more than two) objective optimization algorithm, and RVEA that is a reference direction based algorithm used an angle-penalized metric.

Table 4.4: List of decision variables.

Variable Name	**Variable No**	**Union**	x_i	x_i^ℓ	x_i^u
1PWRPM	x_0	1	2800	2520	3080
1PWTemperature	x_1	2	54	48.6	59.4
1PWWaiting	x_3	3	0	0	0
1PWClosed_Cycle_Period	x_5	4	6	5.4	6.6
1PWPeriod	x_6	5	43	38.7	47.3
1PWWaterInlet	x_8	6	4	3.6	4.4
2MWRPM	x_9	7	3000	2700	3300
2MWLower_Spray_Circulation_Period	x_{11}	8	22.79	20.511	25.069
2MWClosed_Cycle_Period	x_{12}	9	0	0	0
2MWPeriod	x_{13}	10	68	61.2	74.8
2MWTop_Spray_Circulation_Period	x_{14}	11	20.29	18.261	22.319
2MWWaterInlet	x_{15}	12	0	0	0
3MFPeriod	x_{16}	13	0	0	0
4CRRPM	x_{17}	14	2400	2160	2640
4CRPeriod	x_{19}	15	14	12.6	15.4
4CRTop_Spray_Circulation_Period	x_{20}	16	4	3.6	4.4
4CRWaterInlet	x_{22}	17	2.6	2.34	2.86
7RSRPM	x_{23}	18	2600	2340	2860
7RSTemperature	x_{24}	19	54	48.6	59.4
7RSLower_Spray_Circulation_Period	x_{25}	20	6	5.4	6.6
7RSClosed_Cycle_Period	x_{26}	21	4	3.6	4.4
7RSPeriod	x_{27}	22	28	25.2	30.8
7RSUpper_Spray_Circulation_Period	x_{28}	23	12	10.8	13.2
8DSRPM	x_{29}	24	2400	2160	2640
8DSClosed_Cycle_Period	x_{30}	25	0	0	0
10DYWaiting	x_{31}	26	80	72	88
10DYFan	x_{32}	27	0	0	0
10DYFan_Flap	x_{33}	28	0	0	0
10DYExtra_Heater_Offset	x_{34}	29	0	0	0
10DYPeriod	x_{36}	30	82	73.8	90.2

4.2.1 Solution by NSGA-III

Computed solutions by NSGA-III, a well-known multi-objective optimization algorithm based on non-dominated sorting and crowding, are displayed in Figure 4.1. Here, f_1, f_2, f_3, f_4, and f_5 represent Pareto optimal points (or solutions) of CPI, DPI, EC, TD, and WC, respectively.

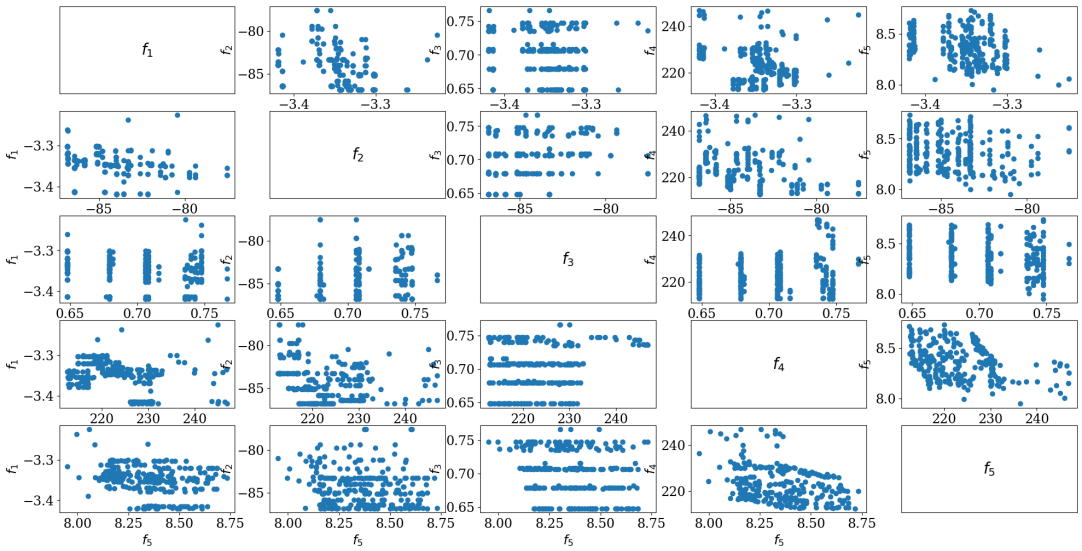


Figure 4.1: Computed solutions obtained by NSGA-III.

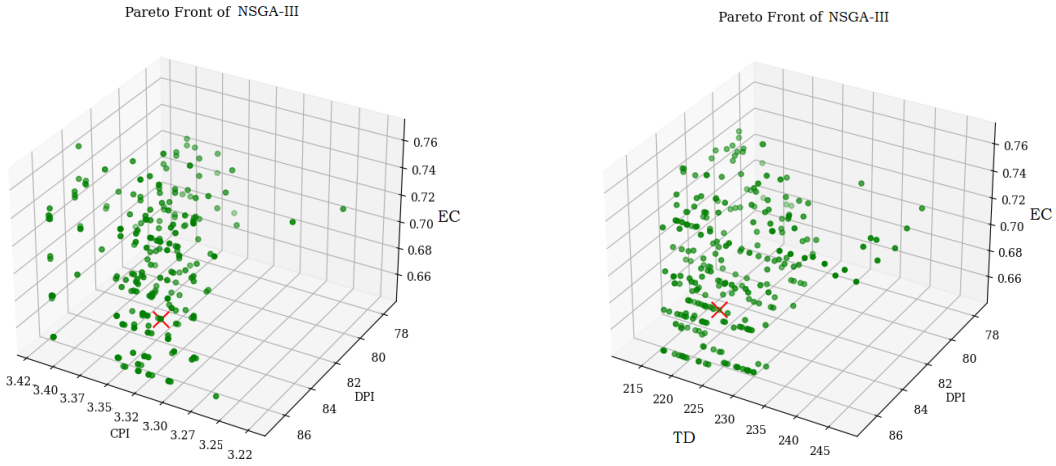
We have 2000 points in the solution since the population size is taken as 2000, and the result is achieved at the 50th generation. The total run time of NSGA-III is 539 seconds, an acceptable value. Corresponding Pareto front limit values are provided in Table 4.5.

Table 4.5: Range of Pareto optimal solutions obtained by NSGA-III algorithm and selected point values for minimum energy case.

Objective Function	Minimum Value	Maximum Value	Selected Point
CPI	3.22	3.42	3.35
DPI	78	87	83
EC	0.65	0.77	0.65
WC	7.95	8.73	8.21
TD	212	246	218

As we aim to design a lower EC cleaning cycle, we select the solution point by giving a weight value of 0.96 to EC and 0.01 to each CPI, DPI, WC, and TD, with a total value of 1. The selected design point's CPI is 3.35, DPI is 83, EC is 0.65, TD is

218, and WC is 8.21. The design point is acceptable with the user preferences, and the improvement in EC is 0.75 to 0.65, which is 13.3%. The position of the selected point on the Pareto front is shown by "×" in Figure 4.2.



(a) Position of selected point on CPI, DPI, EC axes.

(b) Position of selected point on TD, DPI, EC axes.

Figure 4.2: Position of selected point on Pareto front in NSGA-III.

The solution with EMO enables a selection of any point in Figure 4.2 concerning DM's preferences by changing the weights of the importance of objective functions. Since the value of EC is crucial in this study, we assign the highest weight value among other objective functions. The design of the cycles ends up with the new values of the decision variables.

4.2.2 Solution by RVEA

As a second approach, we use reference vector guided evolutionary algorithm (RVEA); see Figure 4.3 for the computed Pareto optimal points.

The population size is 2000 like in the NSGA-III, and the algorithm is terminated at the 50th generation. The total run time for the RVEA run is 562 seconds, an acceptable period. Range of Pareto front points and selected values are provided at Table 4.6. By using the same weight distributions as done in NSGA-III, the selected design point's CPI is 3.34, DPI is 87, EC is 0.65, WC is 8.86, and TD is 220. The design point is acceptable with the user preferences, and the improvement in EC is 0.75 to 0.65, which is 13.3%. The position of the selected point on the Pareto front

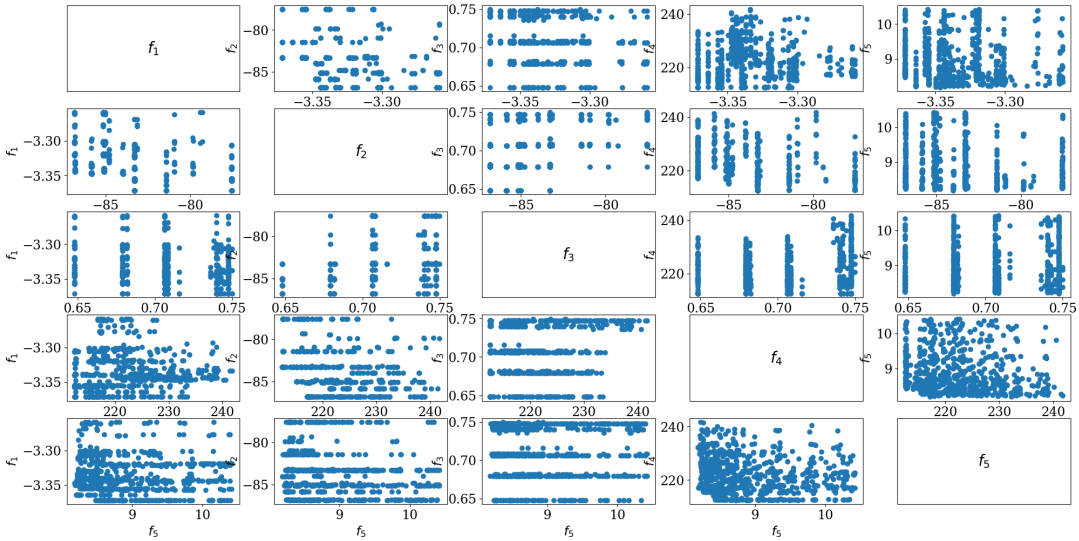
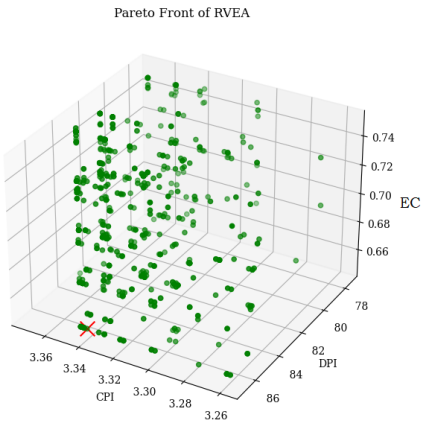


Figure 4.3: Computed solutions obtained by RVEA.

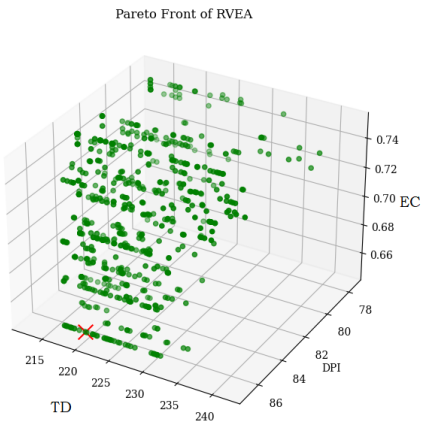
can be seen in Figure 4.4.

Table 4.6: Range of Pareto optimal solutions obtained by RVEA and selected point values for minimum energy case.

Objective Function	**Minimum Value**	**Maximum Value**	**Selected Point**
CPI	3.26	3.37	3.34
DPI	78	87	87
EC	0.65	0.75	0.65
WC	8.18	10.42	8.86
TD	212	241	220e



(a) Position of selected point on CPI, DPI, EC axes.



(b) Position of selected point on TD, DPI, EC axes.

Figure 4.4: Position of selected point on Pareto front in RVEA.

4.2.3 Solution by C-TAEA

C-TAEA, a more sophisticated constraint-handling for many-objective optimization algorithms, is an another approach to solve the optimization problem. Computed solutions obtained by C-TAEA are exhibited in Figure 4.5.

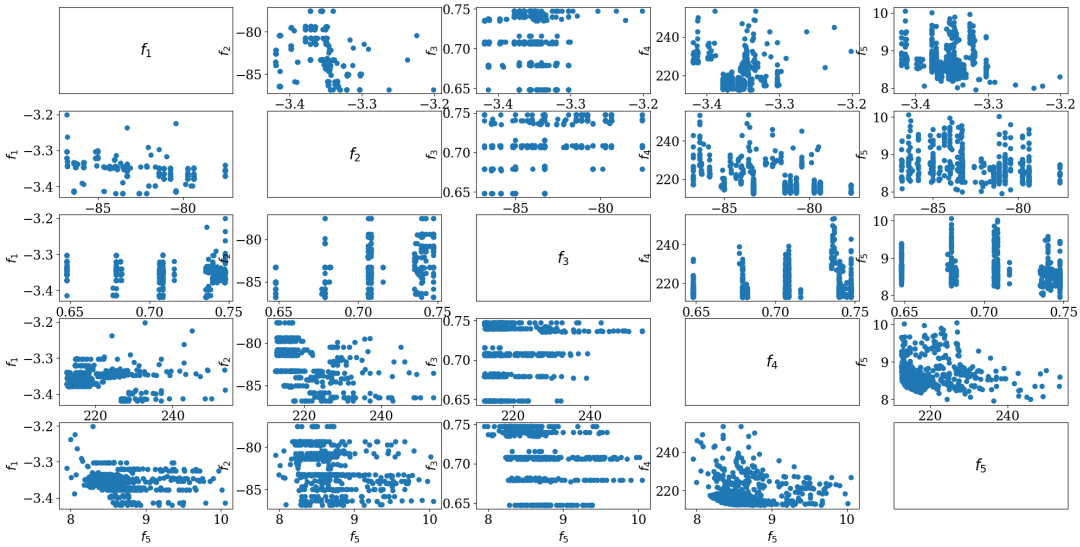


Figure 4.5: Computed solutions obtained by C-TAEA.

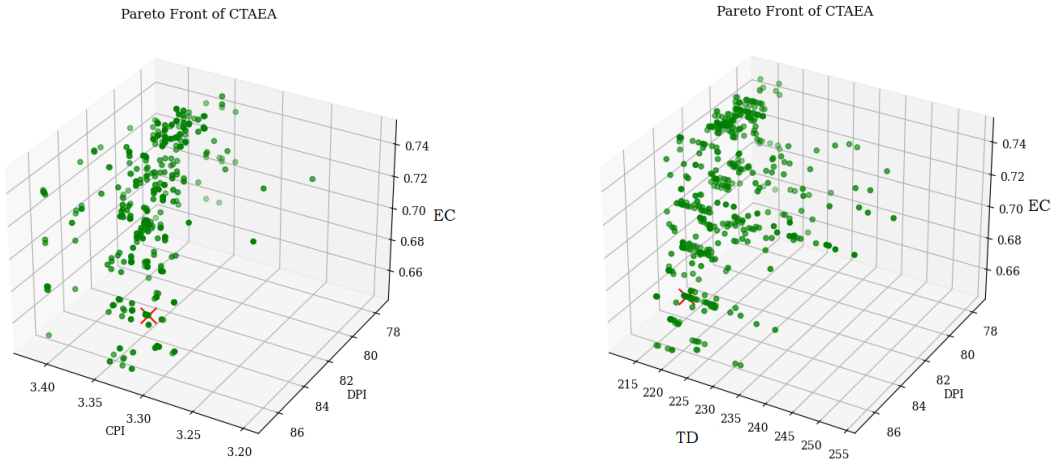
We have 1820 points in the solution since the population size is 1820, and the result is achieved at the 50th generation. The population size is determined by the combination $C(n+p-1,p)$, in where $n=5$ is the number of objectives and $p=12$ is the number of points on the unit simplex that is number of partitions. The total run time for the RVEA run is 785 seconds. The Pareto front limit values are also given in Table 4.7.

Table 4.7: Range of Pareto optimal solutions obtained by C-TAEA and selected point values for minimum energy case.

Objective Function	**Minimum Value**	**Maximum Value**	**Selected Point**
CPI	3.20	3.42	3.35
DPI	78	87	83
EC	0.65	0.75	0.65
WC	7.95	10.05	8.56
TD	212	253	213

The selected design point's CPI is 3.35, DPI is 83, EC is 0.65, WC is 8.56, and TD is 213 according to the weight distribution discussed in the previous cases. The design point is acceptable with the user preferences, and the improvement in EC is 0.75 to

0.65, which is 13.3%. The position of the selected point on the Pareto front can be seen in Figure 4.6.



(a) Position of selected point on CPI, DPI, EC axes.

(b) Position of selected point on TD, DPI, EC axes.

Figure 4.6: Position of selected point on Pareto front in C-TAEA.

4.2.4 Discussion

By using three different algorithms of evolutionary methods, we design three different cleaning cycles with the aim of improvement in the EC value. The outputs of the new design cleaning cycles achieved by all methods are very close to each other and we have improved EC by 0.10 kWh per cycle that is fascinating improvement by only optimizing the cleaning cycle independent variables. The comparison of the algorithms can be done by using hypervolume metric (HV) and the operating time. Since in our case study we do not know the real Pareto front, we need to choose a reference point, taken as $(-3.0, -80, 1.0, 300, 12)$ that should be dominated by all Pareto optimal solutions. Comparison of the evolutionary algorithms is provided in Table 4.8 in terms of operating time and hypervolume metric.

Table 4.8: Comparison of the evolutionary algorithms.

Criteria	**NSGA-III**	**RVEA**	**C-TAEA**
Operating Time	539	562	785
HV	823	712	805

From the results in Table 4.8, we observe that NSGA-III can be considered as the

most suitable evolutionary algorithm to our case study. The values of the decision variables at the selected point are given in Table 4.9 for all evolutionary algorithms. The difference between the new design values and the original domain expert design creates the improvement in EC value of the cleaning cycle. Using our novel method, we can also design infinitely many new cycles with improvements in CPI, DPI, WC, and TD. Additionally, we have ability to design MOOP with different datasets and different regression models.

Table 4.9: Values of the decision variables concerning evolutionary algorithms.

Variable Name	**Variable No**	**Standard Value**	**NSGA-III**	**RVEA**	**C-TAEA**
1PWRPM	x_0	2800	2531	2534	2525
1PWTemperature	x_1	54	49.5	49.5	48.6
1PWWaiting	x_3	0	0	0	0
1PWClosed_Cycle_Period	x_5	6	5.4	5.4	5.7
1PWPeriod	x_6	43	38.7	38.7	38.7
1PWWaterInlet	x_8	4	3.6	3.6	3.6
2MWRPM	x_9	3000	3239	2703	2735
2MWLower_Spray_Circulation_Period	x_{11}	22.79	21.63	21.72	21.41
2MWClosed_Cycle_Period	x_{12}	0	0	0	0
2MWPeriod	x_{13}	68	67.6	62.7	61.5
2MWTop_Spray_Circulation_Period	x_{14}	20.29	18.46	18.94	18.32
2MWWaterInlet	x_{15}	0	0	0	0
3MFPeriod	x_{16}	0	0	0	0
4CRRPM	x_{17}	2400	2331	2634	2627
4CRPeriod	x_{19}	14	13.6	12.7	12.9
4CRTop_Spray_Circulation_Period	x_{20}	4	3.6	4.3	3.9
4CRWaterInlet	x_{22}	2.6	2.4	2.3	2.3
7RSRPM	x_{23}	2600	2859	2858	2846
7RSTemperature	x_{24}	54	48.6	48	47.7
7RSLower_Spray_Circulation_Period	x_{25}	6	5.5	5.4	5.4
7RSClosed_Cycle_Period	x_{26}	4	3.9	3.6	3.7
7RSPeriod	x_{27}	28	25.2	30.5	25.3
7RSUpper_Spray_Circulation_Period	x_{28}	12	11.5	11.6	12.9
8DSRPM	x_{29}	2400	2298	2193	2346
8DSClosed_Cycle_Period	x_{30}	0	0	0	0
10DYWaiting	x_{31}	80	72	72	72
10DYFan	x_{32}	0	0	0	0
10DYFan_Flap	x_{33}	0	0	0	0
10DYExtra_Heater_Offset	x_{34}	0	0	0	0
10DYPeriod	x_{36}	82	74	74	74

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this thesis, we have developed a novel framework based on statistical learning and multi-objective optimization to design a dishwasher cleaning cycle that achieves the target outputs. In developing the framework, we have first studied statistical models that predict the results of a dishwasher cleaning cycle program. A supervised learning model with the current cleaning cycle steps and their experimental results is used as the training data. Our prediction problem is high-dimensional, and we solve the high-dimensional problem by using feature engineering. Numerical simulations show that the genetic algorithm feature selection method gives the best prediction performance in terms of MAE and R-squared with the minimum number of features for both predictions of linear and nonlinear functions. Additionally, we see that regression models with gradient boosting algorithms provide accurate and efficient results for nonlinear functions in our setting. Then, obtained prediction models are used to develop a digital twin performance laboratory to forecast the outputs of the dishwasher's unseen, new cleaning cycles. The digital twin performance laboratory provides time and cost advantages in the new designs.

After setting the obtained predictive models as objective functions for each outputs (or dependent variable), we have defined a multi-objective optimization problem as a new approach to designing a new dishwasher cleaning cycle. To solve the underlying MOOP, we have used evolutionary algorithms such as non-dominated sorting genetic algorithm III (NSGA-III), constrained two-archive evolutionary algorithm (C-TAEA), and reference vector guided evolutionary algorithm (RVEA). Numerical results shows that all methodologies provide promoting results. But, in our setting

(which is to obtain minimum energy), NSGA-III have yielded the best performance in terms of operating time and hypervolume metric. It has improved the energy consumption of a cleaning cycle by 13.3%.

As a future work, expanding the dataset used to train and test the framework will help improve the generalization capabilities of the framework, making it applicable to a broader range of dishwashers and user contexts; it will also enhance the accuracy and robustness of the predictions. Additionally, as a further step the regression performance in a high dimensional problem can be studied to improve by robust linear regression for high dimensional data methods [30] since the new design cleaning cycle performances are highly correlated with the regression performances of the statistical models.

Constrained multi-objective optimization is important in the design of cleaning cycles to obtain effective cleaning cycles, and as a future study the MOOP can be solved with constraints [48]. In terms of practice, since especially energy and water consumption optimization will be a standard approach due to its environmental impact in the future, integrating the framework with smart dishwasher appliances or home automation systems will allow for dynamic adjustments and personalized recommendations based on current conditions, such as water hardness, load size, or dirtiness level.

REFERENCES

[1] H. A. Alaka, L. O. Oyedele, H. A. Owolabi, V. Kumar, S. O. Ajayi, O. O. Akinade, and M. Bilal, Systematic review of bankruptcy prediction models: Towards a framework for tool selection, Expert Systems with Applications, 94, pp. 164–184, 2018.

[2] M. Amer, M. Goldstein, and S. Abdennadher, Enhancing one-class support vector machines for unsupervised anomaly detection, in *Special Interest Group on Knowledge Discovery and Data Mining*, pp. 8–15, 2013.

[3] F. Amini and G. Hu, A two-layer feature selection method using genetic algorithm and elastic net, Expert Systems with Applications, 166, p. 114072, 2021.

[4] Arçelik, Arçelik home page, Available at `https://www.arcelik.com.tr/`, accessed: (01.06.2023).

[5] E. Atasoy, B. Cetin, and O. Bayer, Experiment-based optimization of an energy-efficient heat pump integrated water heater for household appliances, Energy, 245, p. 123308, 2022.

[6] atom ml, Automated tool for optimized modelling, Available at `https://pypi.org/project/atom-ml/`, accessed: (01.01.2023).

[7] S. E. Attoui and M. Meddeb, A generic framework for forecasting short-term traffic conditions on urban highways, in *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–10, 2021.

[8] T. Bartz-Beielstein, J. Branke, J. Mehnen, and O. Mersmann, Evolutionary algorithms, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 4(3), p. 178–195, 2014.

[9] Beko, Beko dishwashers, Available at `https://www.beko.com/de-de/produkte/freistehende-geschirrsp%C3%BCler/60-cm`, accessed: (01.08.2022).

[10] I. Ben-Gal, *Data Mining and Knowledge Discovery Handbook*, chapter Outlier Detection, pp. 117–130, Springer US, 2010.

[11] P. Berkholz, R. Stamminger, G. Wnuk, J. Owens, and S. Bernarde, Manual dishwashing habits: an empirical analysis of uk consumers, International Journal of Consumer Studies, 34(2), pp. 235–242, 2010.

[12] J. Blank and K. Deb, Pymoo: Multi-objective optimization in python, IEEE Access, 8, pp. 89497–89509, 2020.

[13] E. C. Blessie and S. E. Karthikeyan, A feature selection algorithm using correlation based method, Journal of Algorithms and Computational Technology, pp. 385 – 394, 2012.

[14] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, *Feature Selection for High-Dimensional Data*, Springer International Publishing, 2015.

[15] L. Breiman, Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author), Statistical Science, 16(3), pp. 199 – 231, 2001.

[16] J. Brownlee, *XGBoost With Python: Gradient Boosted Trees with XGBoost and scikit-learn*, Machine Learning Mastery, 2016.

[17] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, Cambridge University Press, 2019.

[18] A. Charnes and W. W. Cooper, *Management Models and Industrial Applications of Linear Programming*, John Wiley, 1961.

[19] X. Chen and J. C. Jeong, Enhanced recursive feature elimination, in *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*, pp. 429–435, 2007.

[20] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation, 20(5), pp. 773–791, 2016.

[21] I. Das and J. E. Dennis, Normal-boundary intersection: a new method for generating pareto optimal points in multicriteria optimization problems, SIAM Journal on Optimization, 8(3), pp. 631–657, 1998.

[22] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley and Sons, 2001.

[23] A. V. Dorogush, V. Ershov, and A. Gulin, Catboost: Gradient boosting with categorical features support, Technical Report arXiv:1810.11363, 2018.

[24] D. Edwards and M. Hamson, *Guide to Mathematical Modelling*, chapter What Is Modelling?, pp. 1–4, Macmillan Education UK, 1989.

[25] E. Emary, H. M. Zawbaa, and A. E. Hassanien, Binary grey wolf optimization approaches for feature selection, Neurocomputing, 172, pp. 371–381, 2016.

[26] EN, Electric dishwashers for household use - methods for measuring the performance, Available at `https://standards.iteh.ai/catalog/standards/clc/6640e690-a607-473a-89f3-baf5df351e28/en-50242-2016`, accessed: (01.02.2023).

[27] T. Ergen, A. Mirza, and S. Kozat, Unsupervised and semi-supervised anomaly detection with LSTM neural networks, IEEE Transactions on Neural Networks and Learning Systems, pp. 10–20, 2017.

[28] J. Fan and Y. Fan, High-dimensional classification using features annealed independence rules, The Annals of Statistics, 36(6), pp. 2605 – 2637, 2008.

[29] F. Ferri, P. Pudil, M. Hatef, and J. Kittler, Comparative study of techniques for large-scale feature selection, in *Pattern Recognition in Practice IV*, volume 16 of *Machine Intelligence and Pern Recognition*, pp. 403–413, North-Holland, 1994.

[30] P. Filzmoser and K. Nordhausen, Robust linear regression for high-dimensional data: An overview, WIREs Computational Statistics, 13(4), pp. 1524–1534, 2021.

[31] C. Fonseca and P. Fleming, Multiobjective genetic algorithms, in *IEE Colloquium on Genetic Algorithms for Control Systems Engineering*, pp. 1–5, 1993.

[32] D. Geetha and R. Tyagi, Consumer behavior and fascinating challenges on household laundry and dishwashing, Tenside Surfactants Detergents, 53, pp. 568–575, 2016.

[33] A. Geoffrion, J. Dyer, and A. Feinberg, An interactive approach for multicriterion optimization with an application to the operation of an academic department, Management Science, 19, pp. 357–368, 1972.

[34] M. Gregorich, S. Strohmaier, D. Dunkler, and G. Heinze, Regression with highly correlated predictors: Variable omission is not the solution, International Journal of Environmental Research and Public Health, 18, p. 4259, 2021.

[35] N. Gunantara, A review of multi-objective optimization: Methods and its applications, Cogent Engineering, 5, pp. 1–16, 2018.

[36] A. I. Hammouri, M. Mafarja, M. A. Al-Betar, M. A. Awadallah, and I. Abu-Doush, An improved dragonfly algorithm for feature selection, Knowledge-Based Systems, 203, p. 106131, 2020.

[37] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer New York, 2009.

[38] J. Heaton, An empirical analysis of feature engineering for predictive modeling, in *SoutheastCon 2016*, pp. 1–6, 2016.

[39] N. Hochstrate, B. Naujoks, and M. Emmerich, SMS-EMOA: Multi-objective selection based on dominated hypervolume, European Journal of Operational Research, 181, pp. 1653–1669, 2007.

[40] F. A. Hussein, N. Kharma, and R. Ward, Genetic algorithms for feature selection and weighting, a review and study, in *IEEE Document Analysis and Recognition*, pp. 1240 – 1244, 2001.

[41] IEC, International electrotechnical commission, Available at `https://www.iec.ch/homepage`, accessed: (01.02.2023).

[42] J. Ignizio, *Goal programming and extensions*, DC Heath Lexington, 1976.

[43] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*, Springer, 2013.

[44] A. Jović, K. Brkić, and N. Boović, A review of feature selection methods with applications, in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1200–1205, 2015.

[45] M. Kaisa, *Nonlinear Multiobjective Optimization*, volume 12 of *International Series in Operations Research and Management Science*, Kluwer Academic Publishers, Boston, USA, 1999.

[46] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, Lightgbm: A highly efficient gradient boosting decision tree, in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, p. 3149–3157, Curran Associates Inc., 2017.

[47] S. M. H. Khorasani, Hydraulic simulation model for dishwasher, Available at `http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-266142`, accessed: (01.02.2023).

[48] K. Li, R. Chen, G. Fu, and X. Yao, Two-archive evolutionary algorithm for constrained multiobjective optimization, IEEE Transactions on Evolutionary Computation, 23(2), pp. 303–315, 2019.

[49] A. López Jaimes, S. Zapotecas-Martínez, and C. Coello, *An Introduction to Multiobjective Optimization Techniques*, pp. 29–57, 2011.

[50] B. Mahesh, Machine learning algorithms - a review, International Journal of Science and Research (IJSR), 9, pp. 381–386, 2019.

[51] T. Niu, J. Wang, H. Lu, W. Yang, and P. Du, Developing a deep learning framework with two-stage feature selection for multivariate financial time series forecasting, Expert Systems with Applications, 148, p. 113237, 2020.

[52] D. A. Otchere, T. O. A. Ganat, J. O. Ojero, B. N. Tackie-Otoo, and M. Y. Taki, Application of gradient boosting regression model for the evaluation of feature selection techniques in improving reservoir characterisation prediction, Journal of Petroleum Science and Engineering, 208, p. 109244, 2022.

[53] R. Perez Mohedano, N. Letzelter, and S. Bakalis, Integrated model for the prediction of cleaning profiles inside an automatic dishwasher, Journal of Food Engineering, 196, pp. 101–112, 2017.

[54] B. Pes, Feature selection for high-dimensional data: The issue of stability, in *2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pp. 170–175, 2017.

[55] S. Raschka, Model evaluation, model selection, and algorithm selection in machine learning, Technical Report arXiv:1811.12808, 2020.

[56] N. Razali and Y. Wah, Power comparisons of Shapiro–Wilk, Kolmogorov–Smirnov, Lilliefors and Anderson–Darling tests, Journal of Statistical Modeling and Analytics, 2, pp. 21 – 33, 2011.

[57] scikit learn, Recursive feature elimination, Available at `https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html`, accessed: (01.09.2022).

[58] scikit learn, Sequential feature selection, Available at `https://scikit-learn.org/stable/modules/feature_selection.html#sequential-feature-selection`, accessed: (01.09.2022).

[59] scikit learn, Univariate feature selection, Available at `https://scikit-learn.org/stable/auto_examples/feature_selection/plot_feature_selection.html`, accessed: (01.09.2022).

[60] scikit-learn 1.3.1, Ensembles: Gradient boosting, random forests, bagging, voting, stacking, Available at `https://scikit-learn.org/stable/modules/ensemble.html`, accessed: (01.09.2022).

[61] scikit-learn 1.3.1, Feature selection, Available at `https://scikit-learn.org/stable/modules/feature_selection.html`, accessed: (01.09.2022).

[62] scikit-learn 1.3.1, Linear models and regularization, Available at `https://scikit-learn.org/stable/modules/linear_model.html#`, accessed: (01.09.2022).

[63] R. Stamminger, Modelling dishwashers' resource consumption in domestic usage in european households and its relationship to a reference dishwasher, Tenside Surfactants Detergents, 57(6), pp. 479–488, 2020.

[64] R. Stamminger, A. Bues, F. Alfieri, and M. Cordella, Durability of washing machines under real life conditions: Definition and application of a testing procedure, Journal of Cleaner Production, 261, p. 121222, 2020.

[65] M. A. Stephens, Edf statistics for goodness of fit and some comparisons, Journal of the American Statistical Association, 69(347), pp. 730–737, 1974.

[66] R. E. Steuer and E. Choo, An interactive weighted Tchebycheff procedure for multiple objective programming, Mathematical Programming, 26, pp. 326–344, 1983.

[67] S. Sun, M. Hua, S. Wang, and C. Zhang, How to capture tourists' search behavior in tourism forecasts? A two-stage feature selection approach, Expert Systems with Applications, 213, p. 118895, 2023.

[68] T. Thaher, A. A. Heidari, M. Mafarja, J. S. Dong, and S. Mirjalili, *Evolutionary Machine Learning Techniques: Algorithms and Applications*, chapter Binary Harris Hawks Optimizer for High-Dimensional, Low Sample Size Feature Selection, pp. 251–272, Springer Singapore, 2020.

[69] B. Tran, B. Xue, and M. Zhang, *Simulated Evolution and Learning*, chapter Overview of Particle Swarm Optimisation for Feature Selection in Classification, pp. 605–617, Springer International Publishing, 2014.

[70] H. Wickham, Tidy data, Journal of Statistical Software, 59(10), p. 1–23, 2014.

[71] A. Wierzbicki, *The use of reference objectives in multiobjective optimisation*, chapter X, pp. 468–486, Springer Verlag, 1980.

[72] D. K. Wind, Concepts in predictive machine learning, Available at `http://www.davidwind.dk/wp-content/uploads/2014/07/main.pdf` (2014), accessed: (01.05.2023).

[73] WTO, Ecodesign requirements for off mode, standby, and networked standbyecodesign requirements for off mode, standby, and networked standby energy consumption of electrical and electronic household and office equipment, Available at `https://members.wto.org/crnattachments/2022/TBT/EU/22_0395_01_e.pdf`, accessed: (01.02.2023).

[74] F. Zhang and L. J. O'Donnell, *Machine Learning*, chapter Support vector regression, pp. 123–140, Academic Press, 2020.

[75] L. Zhang and J. Wen, A systematic feature selection procedure for short-term data-driven building energy forecasting model development, Energy and Buildings, 183, pp. 428–442, 2019.