Norwegian University
of Life Sciences

Master's Thesis 2019 30 ECTS
Faculty of Science and Technology

Predictive Maintenance for Fouling in a Plate Heat Exchanger

Numan Mohammad
Master of Science in Data Science

PREFACE

With the advent of the artificial intelligence era, the industry focus is to develop automatic and improved systems while predicting the future issues in the industrial processes earlier, before they occur. A real-time predictive maintenance system for the processing equipment will be valuable and helpful to avoid costly failures. In this project, the focus is to develop a system for Grundfos to predict the disturbances caused by fouling in heat exchangers earlier. The background of this study and the problem statement are presented first. This document presents a brief look into the key concepts of thermodynamics and heat exchangers (especially plate heat exchangers). The theories behind the fouling process, relationship of fouling with different variables, and effect of fouling on economics, the environment, and health are described here. The importance of predictive maintenance along with detailed insight on the fundamentals of machine learning and machine-learning algorithms is discussed. The fouling phenomenon in heat exchangers in terms of machine learning is described in the Results and Discussion section. Since this work is an initiative towards building a predictive maintenance system, suggestions are also presented at the end. This document also contains Python code attached in the appendix that is used to achieve the results.

ABSTRACT

Heat exchangers are very common in industrial processes, such as waste heat recovery and the food and chemical industry as well as in daily life like air-conditioning. One of the major hurdles in the performance, process, and operation of heat exchangers is fouling. Fouling can be of many types, but in the water industry, crystallisation fouling is one of the leading problems. Crystallisation fouling occurs on the surfaces of heat exchangers because of aqueous solutions containing a high concentration of magnesium and calcium ions. In this project, we applied machine-learning tools to make a predictive model. The purpose is to predict when crystallisation fouling is likely to accelerate and thereby reduce the performance of the heat exchanger, so early precautions can be taken to mitigate the fouling phenomenon or plan the future cleaning schedule to reduce the loss of production.

In order to collect data, a lab-scale setup of plate heat exchangers was prepared in which crystallisation fouling by Calcium carbonate ($CaCO_3$) was allowed to occur. The setup was run from fouling-free to fully fouled operation of the heat exchanger, while the readings of the variables were recorded at an every-minute interval during the operation. It was found that the operation of the heat exchanger can be classified into three stages (i.e. no fouling, some fouling, and full fouling). These stages can also be predicted with high accuracy using machine learning tools.

Exploratory analysis by PCA emphasises the presence of hidden structures and patterns in the data, resulting in three clear clusters. These stages can be differentiated with high accuracy by both clustering and classification methods. Logistic regression was chosen as the algorithm for predictive modelling because of its attributes and performance.

Since the prediction of probabilities of the stages helps provide a good assessment of risks, logistic regression performed better in this regard. The pattern of probabilities of the no-fouling stage before the actual some-fouling stage indicates that precautions are required to avoid the some-fouling stage that can otherwise reduce the performance of heat exchangers. When the number of features in the data was reduced, the performance of logistic regression improved, indicating that some variables are causing noise in the prediction. Four features, cold-water exit, hot-water exit, POS, and speed were evaluated as important by feature importance permutation.

Our results emphasise that the data collected from the sensors can be used to predict when the heat exchanger is not working normally. The potential to build high-quality predictive models for fouling is encouraging despite various limitations in this work.

ACKNOWLEDGEMENTS

First and foremost, I thank Allah Almighty for giving me the strength, purpose, and endurance to complete this work.

I offer my sincerest gratitude to my supervisors, Associate Professor Kristian Hovde Liland, Associate Professor Oliver Tomic, and senior engineer Kristian Sørby Omberg, for their valuable input, productive criticism, and encouraging guidance. Their knowledge and expertise in the field of data science was very helpful in achieving the goals of the project. Their advice, trustworthy support, and continuous encouragement was priceless. This accomplishment would not have been possible without their assistance and dedicated involvement in every step throughout research and writing.

Finally, I would like to offer special thanks to my colleagues who were a constant source of support and encouragement for me.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

$CaCO_3$	Calcium Carbonate
$CaSO_4$	Calcium Sulphate
EDA	Exploratory Data Analysis
IoT	Internet of Things
˚C	Degree Celsius
OvR	One vs Rest
PCA	Principal Component Analysis
PdM 4.0	Predictive Maintenance 4.0
Re	Reynolds number
SKLEARN	Sci-kit Learn
W	Watt

1 Introduction

1.1 Background

Heat exchangers are very common in industrial processes, such as waste heat recovery and the food and chemical industry as well as in daily life like air-conditioning. One of the major hurdles in the performance, process, and operation of heat exchangers is fouling, which is the result of the build-up of unwanted materials on the surface of the heat exchangers. The negative consequences of fouling are quite significant.

For major industrial nations, the estimated total fouling-related costs exceeds US$4.4 milliard annually (Ibrahim, 2012). The losses related to the fouling of heat exchangers are estimated to be 0.25% to 0.30% of their gross domestic product (Ibrahim, 2012). The cleaning costs are usually around US$40,000 to 50,000 per heat exchanger per cleaning (Ibrahim, 2012). Apart from financial losses, fouling results in major health and environmental hazards, such as the emission of carbon dioxide and the discharge of potentially harmful fouling inhibitors into the environment.

As far as the effects of fouling on heat exchangers are concerned, the efficiency and heat transfer rate of heat exchangers are reduced. That further results in the loss of energy and overall production in a production plant. If fouling is not mitigated properly, the equipment can be damaged, which can stop the production plant. The maintenance of heat exchangers is essential to avoid unexpected costs and problems.

Since fouling in heat exchangers causes significant environmental and economic issues, mitigation is essential. Machine learning tools are handy in this regard. The ability to find hidden patterns and structures in the collected data from sensors is helpful. These patterns can give us indications of when the fouling is likely to happen and accelerate. The predictive maintenance system created using machine learning can help to schedule the optimal time for the maintenance and cleaning of heat exchangers.

Grundfos is leading worldwide in advanced pump solutions and develops innovative, efficient, and sustainable water solutions. One of the main goals is to reduce the environmental footprint by encouraging sustainable solutions. Grundfos is looking to develop and innovate a better solution to predict and estimate when fouling is likely to initiate, accelerate, and thus reduce the performance of heat exchangers. As a first step towards achieving this goal, a lab-scale setup of heat exchangers is created to study the fouling phenomenon owing to limestone presence in hard water.

1.2 Problem Statement

The estimated cost of unplanned downtime of industrial manufacturers is $50 billion annually of which 42% is caused by equipment failure (IndustryWeek, 2019). Unplanned maintenance can lead to excessive maintenance and equipment replacement costs. Data from the US Department of Energy signify the importance of predictive maintenance by indicating it as extremely cost-effective (Jiménez, 2018). A predictive maintenance system can reduce 25% to 30% of the maintenance cost and 70% to 75% of breakdowns, while increasing 35% to 45% in uptime (Jiménez, 2018).

The main idea of predictive maintenance in this study is to predict the presence of fouling before it causes any significant loss in the performance and surface of the heat exchanger. Grundfos is currently investing to make a predictive maintenance system that can automatically plan the maintenance schedules for process equipment. The ultimate goal of Grundfos is to build a system similar to Predictive Maintenance 4.0 (PdM 4.0).

Predictive Maintenance 4.0 involves the implementation of machine-learning techniques to identify useful hidden patterns and structures in huge volumes of data and generate insight to improve decision making for better productivity of assets. Therefore, the overall goal of this project, as an early approach, is to develop a predictive system to determine the fouling phenomenon in a heat exchanger.

1.3 Goals and objectives

The objectives of this thesis are the following:

1. Literature review of key concepts and theory.
2. Model training and testing.
    a. build a predictive model based on the given data
    b. predict and prevent fouling before it occurs using machine learning tools
    c. find specific parameters that could indicate the presence of fouling in the heat exchanger
3. Suggestions for further work.

1.4 Limitations

There are some clear limitations that will be considered in this paper. First, the focus will be on plate heat exchangers. During the experiment, it was assumed that only crystallisation fouling caused by calcium carbonate ($CaCO_3$) influenced the performance and operation of heat exchangers. If any other type of fouling or any other interruption happens during the operation of the heat exchanger, we ignore it. Therefore, the collected variables are only affected by the presence of $CaCO_3$ fouling.

2 Theory and Key Concepts

This section gives a general review of the thermodynamics and importance of heat exchangers along with a detailed discussion on the fouling phenomenon and its negative effects on heat exchangers and the environment.

2.1 Thermodynamics Fundamentals

Thermodynamics is the branch of physics that deals with the transformation of different forms of energy. The relation between heat, energy, and work is studied. There are four laws of thermodynamics. The first and second laws are important here. The first law also known as the law of energy conservation, which states that energy cannot be destroyed or created but can change from one form to another. For example, when fuel in a car engine is ignited, a portion of the energy is used as work (to move the car) while the remaining energy is released to the surroundings, but the total energy of the system remains constant. The sum of the work done and the energy wasted is equal to the energy produced by burning the fuel. During the process, energy is neither created nor destroyed but changes from one form to another. The conversion of all energy into work is limited by the second law. According to the second law, heat flows spontaneously from a hotter region to a colder region, but the heat cannot flow spontaneously from a colder region to a hotter region unless work is applied. A process is irreversible if the entropy (a measure of unavailable energy) of an isolated system does not decrease.

2.2 Heat Exchangers

A key device in thermodynamics is the heat exchanger. A device used to transfer thermal energy between fluids (two or more) at different temperatures is called a heat exchanger. A few of the major applications are in the food, chemical, and process industries, air-conditioning, power sector etc. (Kakaç, Liu, & Pramuanjaroenkij, 2002). By 2020, the projected global market size of heat exchangers will cross $20.5 billion (Prescient & Strategic Intelligence, 2016).

Heat exchangers are classified based on the following attributes (Shah & Sekulic, 2003):

1. Heat transfer mechanisms and processes,
2. Number of fluids and their flow arrangements,
3. Surface compactness.

Common types of heat exchangers are shell and tube and plate heat exchangers. The convective heat transfer equation is considered a basic equation in designing heat exchangers (Melo, Bott, & Bernardo, 2012), which is given as follows:

$$Q = UA\Delta T,$$

where
Q = the heat transfer rate
U = the overall heat transfer coefficient
ΔT = the negative driving force (difference between the fluid temperature and surface temperature)
A = the heat transfer surface area

2.2.1 Plate heat exchanger

The basic building block of plate heat exchangers is metal plates. The two fluids at different temperatures are passed through the portholes of a pack of the corrugated plates, as shown in **Figure** *2.1*. The fluids are passed through alternate channels while the heat transfer occurs through the plate located between the channels. The determination of the number of plates is one of the important tasks. It can be determined by considering the following factors: the pressure drop, temperature, physical properties, and flow rate of the fluids (Alfa Laval, 2019).

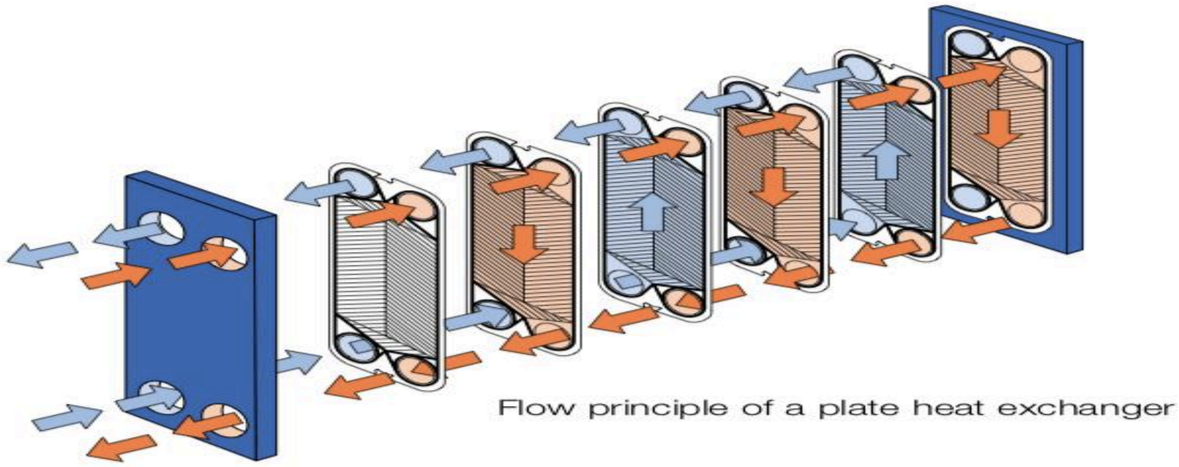


Figure 2.1: *Flow process of an M3 Plate heat exchanger (Alfa Laval, 2019).*

2.2.2 Attributes of plate heat exchangers

Plate heat exchangers are useful in many applications owing to the following attributes (Shah, Subbarao, & Mashelkar, 1988):

1. The flexibility to readily change or reorganise the heat transfer surface area for variable tasks.
2. The plates cause high turbulence which results in reduction of fouling. Plate heat exchangers are 10% to 25% less prone to fouling as compared to most common shell and tube heat exchangers.
3. These heat exchangers have high heat transfer coefficients.
4. The design of these heat exchangers prevents the leakage of one fluid into the other.

The most suitable application of plate heat exchangers is in liquid-liquid heat transfer (Shah, Subbarao, & Mashelkar, 1988). Some of the examples are their use in pharmaceutical, dairy, and beverage industries.

2.3 Fouling

Fouling is referred to as the build-up of unwanted materials on the surface of heat exchangers. The deposited material can result from insoluble salts, suspended solids, scale, or algae (Ibrahim, 2012). The accumulation and deposition can occur on the interior or exterior of heat exchangers. Some of the common encountered foulants in industrial operations can be divided into the following categories (Bott, 1995):

1. Organic materials: This category includes biological materials, such as bacteria and algae, and heavy organic deposits, such as tar.
2. Inorganic materials: This category includes magnesium and calcium salts, iron oxide, dusts and mud.

Since the deposited material resists the transfer of heat, the efficiency of heat exchangers decreases owing to the presence of fouling. The resistance to heat flow increases as the thickness of fouling layers on the surface increases. The phenomenon of heat transfer reduction in the presence of fouling is shown in **Figure *2.2***. Heat flow resistance across a solid surface is specified as follows (Bott, 1995):

$$\frac{\text{Solid thickness}}{Thermal\ conductivity\ of\ the\ particular\ solid\ (foulant)}.$$

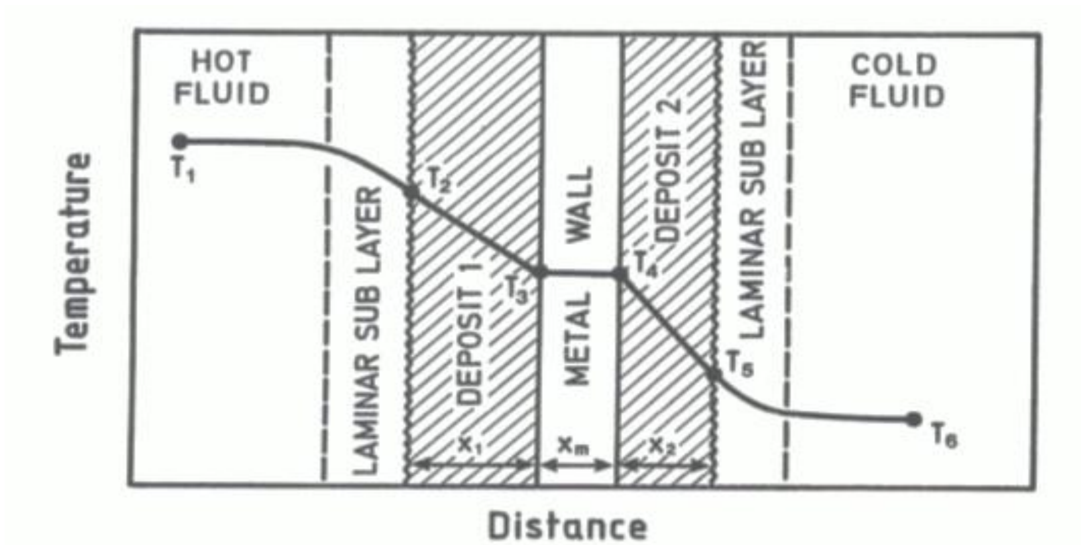


Figure 2.2: *Temperature distribution across the heat exchanger surface in the presence of fouling layers (deposit 1 and deposit 2). Laminar sub-layers are boundary layers that also reduce the heat flow. The temperatures of bulk hot and cold fluids are represented by T1 and T6, respectively (Bott, 1995).*

2.3.1 Laminar and turbulent flow

As obvious from the ***Figure 2.2,*** the boundary layer of the fluid also influences the heat transfer rate. The thickness and behaviour of the boundary layers of the fluid vary depending on the fluid behaviour. The density and viscosity of fluids influence the way fluids behave. The Reynolds number indicates whether the fluid flow is laminar or turbulent in a pipe:

$$Re = \frac{du\rho}{\eta},$$

where *Re* is the Reynolds number, d is the inside diameter of the pipe, and ρ, η, and u are the density, viscosity, and velocity of the fluid, respectively. The Reynolds number represents the ratio of momentum forces to viscous forces. The flow property of a fluid can be estimated from the Reynolds number as given below (Bott, 1995):

1. $Re < 2000$ indicates laminar flow,
2. $2000 \leq Re \leq 3000$ indicates the transition region (flow is between turbulent and laminar),
3. $Re > 3000$ indicates turbulent flow.

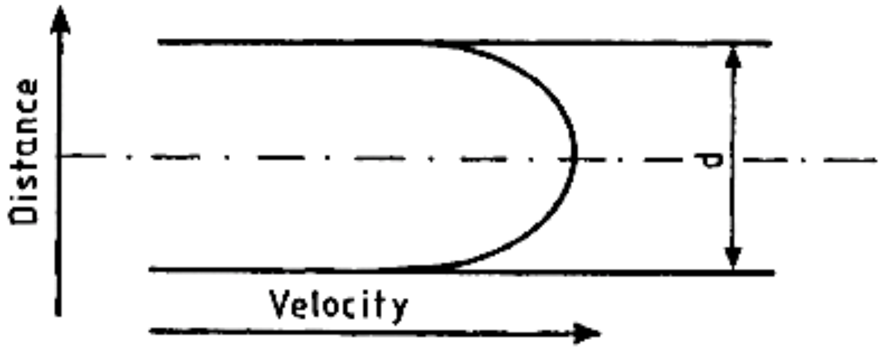


Figure 2.3: *Velocity profile of laminar flow showing that the layers near the boundary are very slow compared to the flow in the middle (Bott, 1995).*

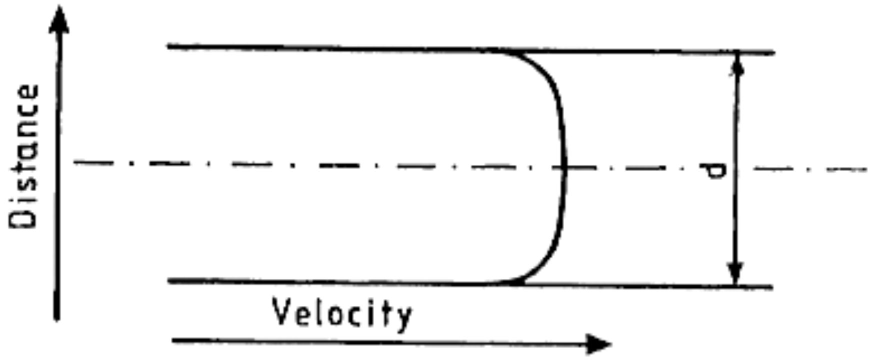


Figure 2.4: *Velocity profile of turbulent flow showing that the layers near the boundary are not as slow as in the laminar flow (Bott, 1995).*

The velocity profile of the laminar flow is parabolic, while the turbulent flow is not parabolic, as shown in **Figure** *2.3* and **Figure** *2.4*. The mean velocity of the laminar flow is half the maximum velocity, while that of the turbulent flow is 0.82 *times* the maximum velocity in a tube (Bott, 1995). The maximum velocity is at the centre of the tube, as the fluid suffers the minimum resistance at this location. As mentioned earlier, high turbulence causes a reduction in fouling. The reason is obvious from the velocity profiles. The boundary layers are more likely to remain stagnant in laminar flow compared to turbulent flow.

2.3.2 Fouling process

The general fouling process can be divided into a five-step fouling formation process

- Initiation,
- Transport,
- Adhesion,
- Removal, and
- Ageing.

2.3.2.1 Initiation

The period during which no deposition takes place when a clean heat exchanger starts operating is the initiation time (Kazi, 2012) as shown in **Figure** *2.5*. The period during which the fouling process begins until the fouling resistance again diminishes to zero is called the roughness delay time (Kazi, 2012). The roughness delay time is different for different types of fouling. For example, it is very small for particulate fouling and fairly large for crystallisation fouling (Kazi, 2012). Linear, falling, and asymptotic are three categories of typical fouling curves that may occur after the roughness delay time. The linear fouling curve is a result of very strong deposits. The falling curve is a result of deposits with lower mechanical strength, and the asymptotic curve is the result of weak deposits (Kazi, 2012). The most common type for different types of fouling is the asymptotic fouling curve (Kazi, 2012).

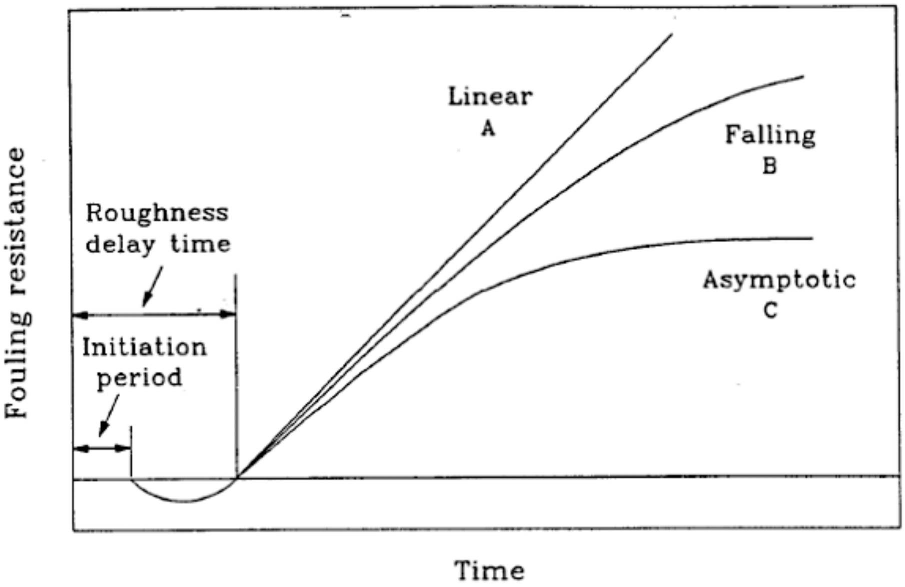


Figure 2.5: *Three typical fouling curves that may occur in a general fouling process (Kazi, 2012).*

2.3.2.2 Transport

In this step, the solid particles present in the bulk fluid are transferred to the surface. Mass transfer of foulant components within a fluid occurs when a concentration gradient of that particular material exists.

Let us consider the diffusion of molecules of a component A towards a surface through a fluid B. The random motion of molecules, which is called Brownian movement, causes the mass transfer within the fluid system in a laminar flow given by Fick's law (Bott, 1995):

$$N_A = -D_{AB}\frac{dc_A}{dx}, \qquad 2.1$$

where
N_A = the diffusion rate of the molecules of A,
$\frac{dc_A}{dx}$ = the concentration gradient of A, and
D_{AB} = the diffusivity of A through fluid B.

In turbulent flow, the random physical movement of molecules occurs owing to turbulent conditions given as follows (Bott, 1995):

$$N_A = -(D_{AB} + E_D)\frac{dc_A}{dx}, \qquad 2.2$$

where E_D is the eddy diffusion as a result of turbulent conditions.

The transport of foulant particles depends on the following (Bott, 1995):

- The concentration gradient between the fluid/surface interface and the bulk fluid,
- The physical properties of the components of the system, and
- The fluid flow conditions (laminar or turbulent).

The transport mechanism of fouling materials on the surfaces is extremely complex, making it difficult to fully understand and comprehend the precise mechanisms resulting in the fouling process (Bott, 1995).

2.3.2.3 Adhesion

The interaction occurs between the surfaces and particles of the fouling material in the initiation stage. As the fouling layer thickens, the interaction occurs between the particles attached to the surface and the fresh foulant particles reaching the solid/fluid interface. The attractive forces that bring the foulant particles to the surface and create little interactions are long-range forces, which may be magnetic attraction, van der Waals, and electrostatic forces (Bott, 1995).

2.3.2.4 Removal

The shear forces at the fluid/deposit interface are responsible for the removal process. These shear forces depend on the following (Kazi, 2012):

- Surface roughness,
- Physical properties like the viscosity of the fluid, and
- Velocity gradients at the fluid/deposit interface.

The loosely attached particles are removed from the deposits by these forces.

2.3.2.5 Ageing

Ageing of fouling begins right after deposition forms. It increases or decreases the bond strength of the deposit. Changes in the crystalline structure can change the chemical and mechanical properties of the fouling layers with ageing (Kazi, 2012).

In short, fouling is a consequence of two simultaneous processes:

- Deposition process,
- Removal process.

Therefore, the net accumulation rate is the deposition rate minus the removal rate.

The change in the deposition rate and removal rate results in three typical fouling curves, as shown in **Figure** *2.5*. In the linear fouling curve, the deposition rate is faster compared to the removal rate. The deposition rate decreases in the falling curve. The removal rate increases and becomes equal to the deposition rate in the asymptotic fouling curve (Kazi, 2012).

2.3.3 Classification of fouling

The fouling on heat exchanger surfaces is classified into six groups (Melo, Bott, & Bernardo, 2012):

1. Biological,
2. Particulate,
3. Crystallisation,
4. Chemical reaction,
5. Corrosion, and
6. Freezing.

Industrial heat exchanger fouling, except in a few special cases, cannot occur entirely owing to one mechanism (Melo, Bott, & Bernardo, 2012). Industrial problems are the consequences of the occurrence of several mechanisms, such as cooling water systems that involve combinations of biological, particulate, and corrosion mechanisms (Melo, Bott, & Bernardo, 2012).

2.3.3.1 Crystallisation fouling

Local ground water, which is actually hard water, is used in laboratory-scale experiments as described in Section 6.1. Hard water contains a high concentration of magnesium and calcium ions. The hardness salts (alkaline earth metal salts) are often found in boilers, cooling condensers, desalination salts, geothermal plants, food processing equipment, oil production equipment, and reverse osmosis plants (Melo, Bott, & Bernardo, 2012). These salts in hard water cause fouling problems in heat exchangers. Crystallisation fouling is one of the common causes of fouling on the surface of heat exchangers (Melo, Bott, & Bernardo, 2012). It usually occurs in aqueous solutions containing soluble inorganic hardness salts, such as $CaCO_3$ and calcium sulphate ($CaSO_4$).

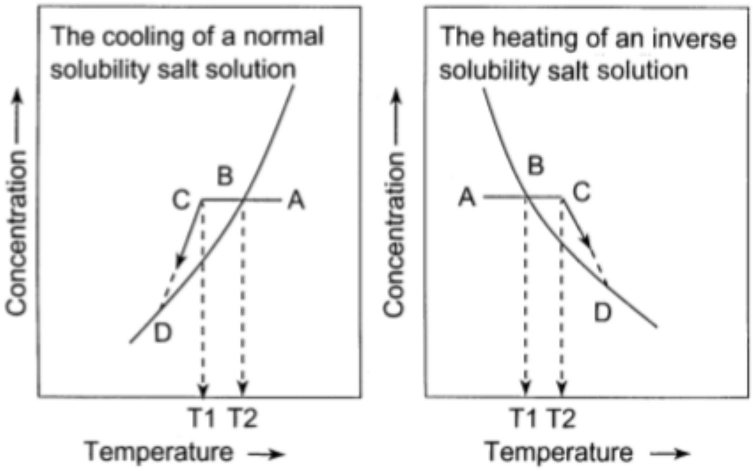

The cooling of a normal solubility salt solution
Concentration
B
C
A
D
T1 T2
Temperature
The heating of an inverse solubility salt solution
Concentration
B
A
C
D
T1 T2
Temperature

Figure 2.6: *Normal and inverse solubility phenomenon. In normal solubility (left), the solution is undersaturated at point A, and it becomes saturated when cooled to point B. The solution converts to supersaturation on further cooling, and nucleation occurs at C. In inverse solubility, the undersaturated solution at point A on heating becomes saturated at point B. The solution converts to supersaturation on further heating, and nucleation occurs at C (Khan, Budair, Sheikh, & Quddus, 1996).*

The three sequential stages (Bott, 1995) occur in the crystallisation processes given as follows:

Supersaturation: A solution is said to be saturated if a soluble solid substance is in equilibrium with the solvent. If more solid is added, it will not be dissolved in the solvent and will remain a solid. A solution can be made supersaturated by decreasing the temperature of a saturated solution in normal solubility situations. A soluble solid substance has different saturation levels at different temperatures.

In normal solubility salts, the solubility increases with increasing temperature, while solubility decreases with increasing temperature in inverse solubility salts. The phenomenon of normal and inverse solubility has been described in **Figure** ***2.6***. Some of the inorganic salts, such as $CaCO_3$ and $CaSO_4$ have inverse solubility characteristics in water (Khan, Budair, Sheikh, & Quddus, 1996). Predominately present in cooling water (Khan, Budair, Sheikh, & Quddus, 1996), $CaCO_3$ is less soluble in warm water, as shown in **Figure** ***2.6***. This lower solubility results in scale deposition on the surface of the heat exchanger. Supersaturation can occur if the temperature of a normal solubility salt solution is decreased or if inverse solubility salts are increased. There are other ways that can cause supersaturation, but these are out of the scope for this study. The rate of the deposition process in crystallisation is usually determined by the extent of supersaturation (Bott, 1995).

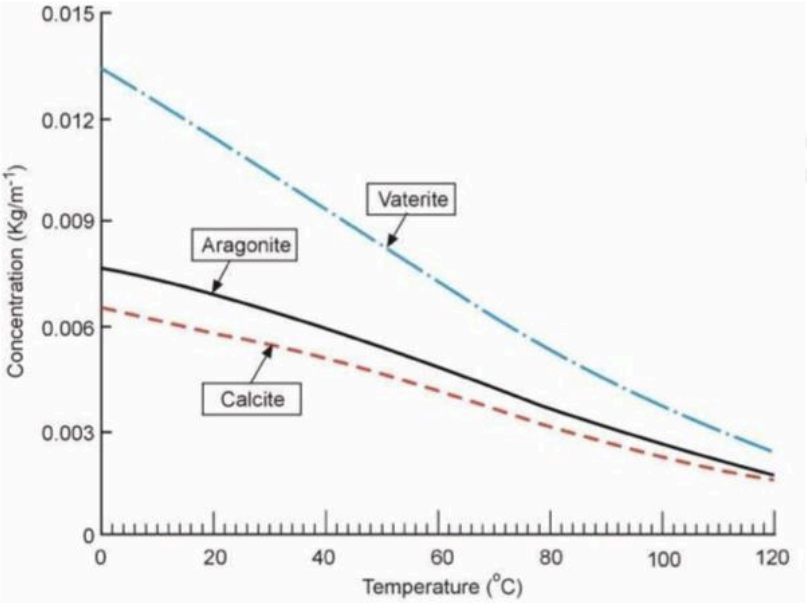


Figure 2.7: *Calcite, aragonite, and vaterite are three polymorphs of calcium carbonate. The curves show the decrease in solubility of calcium carbonate coming from different polymorphs with increasing temperature (Jamialahmadi & Müller-Steinhagen, 2012).*

Nucleation: A cluster that forms in the solution when molecules, atoms, or ions come close together is called a nucleus. These nuclei grow and become stable if conditions like mass transport to the nucleus are sufficient. The nuclei die or grow depending on the conditions. Heat exchanger surfaces provide nucleation sites (locations where nuclei can form and grow). Similar to the initiation step discussed in the fouling formation process, during the induction period in crystallisation, nucleation occurs but causes no (or a negligible) change in the overall heat transfer coefficient in the heat exchanger (Geddert, Augustin, & Scholl, 2011). The length of the induction period is influenced by a number of factors but can be determined by the following parameters (Geddert, Augustin, & Scholl, 2011): operation-related parameters, such as the flow regime, process-related parameters, such as the temperature, and equipment-related parameters, such as surface properties.

Crystal growth: Once nucleation is successful and the nuclei achieve critical size, the growth period starts.

2.3.4 Influence of fouling

The mitigation of fouling is a costly process. The main areas related to fouling costs are discussed below.

2.3.4.1 Capital investment

Since the deposition of fouling particles reduces the surface area of heat exchangers, the normal size of heat exchangers is increased by 70% to 80%, of which 30% to 50% is for fouling, to extend the operation time (Kananeh & Peschel, 2012). The annual capital cost increases because of over-sizing the heat exchanger to accommodate for fouling. Investment is also needed to use expensive anti-fouling agents. In the presence of fouling, more power or energy is required to transfer heat. Since the deposits result in a pressure drop because of the lower area for fluids to flow, more pumping power is required, resulting in more cost. Failure to assess the possible fouling hazard in time may result in heat exchanger replacement.

2.3.4.2 Loss of production and energy

In the presence of fouling, more energy is required to transfer heat. As deposit results in pressure drop because of lower area for fluid to flow, more pumping power is required. The overall production of the plant will be compromised if the heat exchanger performance decreases.

2.3.4.3 Maintenance cost

Planned maintenance and cleanliness of heat exchangers is essential to avoid the failure of equipment. Since a complex, multistage process of cleaning heat exchangers involves a variety of surfactants, acids, and alkaline solutions (Müller-Steinhagen, Malayeri, & Watkinson, 2009), the mitigation of fouling is costly. The cleaning costs are usually around US$40,000 to 50,000 per heat exchanger per cleaning (Ibrahim, 2012).

2.3.5 Fouling relation with variables

Fouling is an unsteady and dynamic process (Awad, 2011). Various design and operational variables are found to influence the fouling phenomenon. A few of the most important parameters (Awad, 2011) are given below:

1. **Fluid Flow velocity:** The thermal performance of the heat exchanger increases when the flow velocity is increased, which further reduces the fouling rate.
2. **Surface material:** The surface material can significantly affect the fouling process. A good and reliable material should be chosen based on the requirements and cost. For example, glass and graphite tubes have low thermal conductivity but often resist fouling, while carbon steel is least expensive but corrosive.
3. **Surface temperature and roughness:** Since surface temperature may increase or decrease the fouling rates, the optimum surface temperature is required to reduce fouling rates. The surface roughness can significantly influence the fouling process. Roughness

on the surface encourages initial deposition, while surfaces with a better finish delay the fouling process.

4. **Fluid properties:** Shear forces play an important role in the removal step of the fouling process. These forces depend on properties such as the viscosity and density of the fluid. Since viscosity is an important factor in deciding the sublayer thickness on the surface, the transport step is influenced by this sublayer.
5. **Others:** Impurities and suspended materials, the heat transfer process, and design considerations are also important parameters that influence the fouling phenomenon in heat exchangers

2.3.6 Environmental and health hazards

The environmental effect of fouling in heat exchangers is significant. Fouling contributes in terms of environmental and health hazards in the following ways (Müller-Steinhagen, Malayeri, & Watkinson, 2009):

- Since more energy is required from the electric power generators to transfer heat in the presence of fouling, carbon dioxide emissions are increased, thus contributing to global warming.
- Potentially harmful fouling inhibitors, such as anti-fouling and anti-foaming agents, are discharged into land or water resources.
- Since a complex, multistage process of cleaning heat exchangers involves the use of a variety of surfactants, acids, and alkaline solutions, their discharge to the environment is potentially harmful.
- The discharge of removed deposits that may include bacteria, carcinogenic, and radioactive matter to the environment.

2.3.7 Fouling monitoring techniques

Monitoring of fouling is fundamental and crucial in fouling research in heat exchangers. The most difficult part is to understand the mechanism of fouling, as it depends on several factors, such as heat and mass transfer, fluid flow, surface materials, chemical reactions, and so on. In addition, a prerequisite to encourage fouling studies is to have much more experimental data (Zhenhua, Yongchang, & Chongfang, 2008).

The different measurement approaches used to monitor fouling are given below (Melo, Bott, & Bernardo, 2012):

1. The hot fluid temperature, test fluid velocity, and inlet temperature are kept constant. If any change in the outlet temperature of the test fluid occurs, it indicates the presence of fouling in the heat exchanger.
2. The inlet temperature, outlet temperature, and flow rate of the test fluid are kept constant. If any increase in the heating medium temperature occurs, it indicates the presence of fouling in the heat exchanger.

3 Predictive Maintenance

The main idea of predictive maintenance is to predict failures before they cause any significant loss in the performance of industrial equipment. The benefits of predictive maintenance include the reduction of unscheduled downtime, maintenance costs, and production costs.

The process of predictive maintenance can be summarised in three steps (CSS Electronics, 2019) as follows:

1. Collection of real-time data,
2. Prediction of future failures and issues by processed data using machine-learning tools, and
3. Taking necessary actions by informing the maintenance team.

Predictive maintenance consists of four levels (CSS Electronics, 2019):

- Level 1: Visual inspections,
- Level 2: Instrument inspections,
- Level 3: Real-time condition monitoring, and
- Level 4: Decision making using big data analytics.

Level 4 can also be called Predictive Maintenance 4.0 (PdM 4.0), which involves the implementation of machine-learning techniques to identify useful hidden patterns and structures in huge volumes of data and generate insight to improve decision making for better productivity of assets. Some of the commercially available technologies are discussed below.

4 Commercial Predictive Maintenance Applications

4.1 Petasense

Petasense (2019) is an industrial Internet of things (IoT) startup that helps industrial customers monitor, assess, and predict machine health in real time. Wireless sensors are installed on machines, and these sensors send data securely to the cloud, where machine learning is implemented to assess asset health and optimisation issues. Then, the performance and health of the assets can be visualised in real time by the maintenance team. Petasense systems seems to follow Predictive Maintenance 4.0 quite well.

4.2 Hexxcell

Hexxcell Ltd. (2019) developed a state-of-the-art software called Hexxcell Studio $^{\text{TM}}$ to help industries assess, mitigate, and predict fouling in refinery heat exchangers. This software has successfully predicted the future behaviour of fouling, the possible ways to mitigate it, and its economic effects. Advanced monitoring of Hexxcell gives insight into the thermo-hydraulic performance of heat exchangers by visualising important parameters, such as fouling resistance, stress, velocity, etc. In predictive maintenance, the performance and operation of the previous month for the heat exchanger is investigated in detail. The performance is then predicted over the next few months based on the current operation plan. Possible future scenarios are analysed, and an operation plan is recommend based on future benefits for the next months.

5 Machine Learning Fundamentals

Numerous data are generated daily by individuals, by industries, and by internet searches. Collecting such data would be meaningless if we cannot obtain information or knowledge from it. Machine learning can help us extract information from such collected data. Machine learning, a subfield of artificial intelligence, transforms data into valuable knowledge to make predictions using algorithms. It automatically extracts relevant information from data and applies it to analyse new and unseen data. It provides a way to make predictive models based on hidden patterns in a historical dataset. The prerequisite to build a machine-learning model is to have some data that can be used to train algorithms. Furthermore, the quality and reliability of the model is dependent on the quality and truthfulness of the data. Better the quality data produce a better model. For example, if dataset contains information wrongly typed or inserted, the quality of model will be compromised. In simple terms, machine learning can help build a system that learns from data; identifies patterns, structures, and correlations in data; and makes reliable predictions.

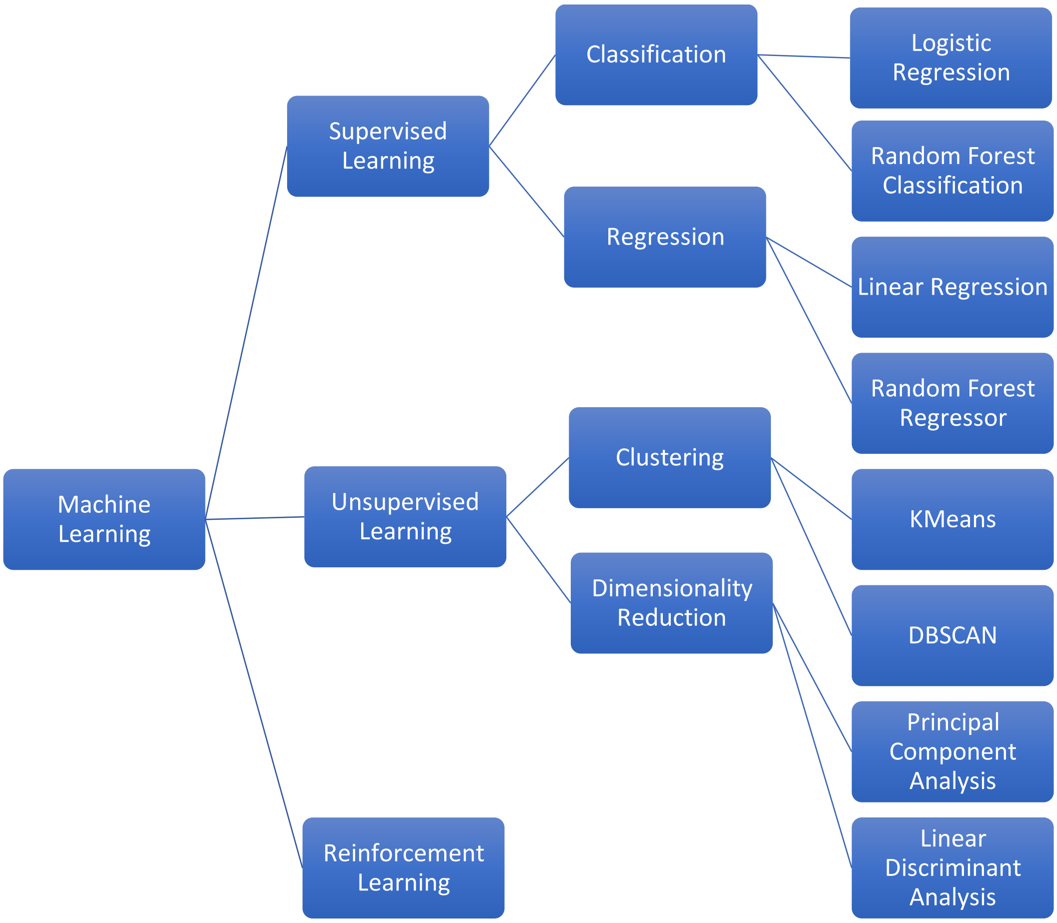
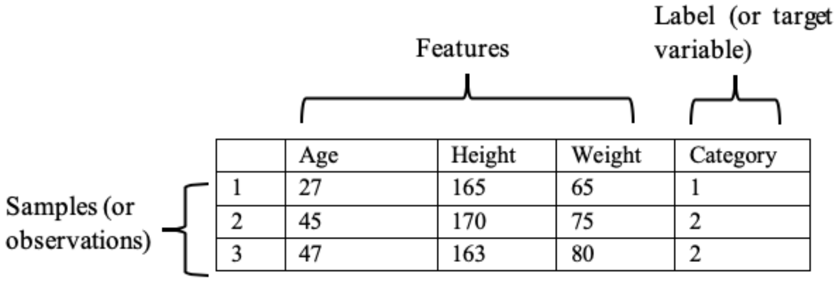


Figure 5.1: *An overview of the machine learning models. This figure displays only a few machine learning models.*

Machine learning can be of three types, depending on the data as follows:

1. Supervised learning,

2. Unsupervised learning, and
3. Reinforcement learning.



	Age	Height	Weight	Category
1	27	165	65	1
2	45	170	75	2
3	47	163	80	2

Figure 5.2: *A schematic showing the basic terms of machine learning (i.e. samples, features, label) in an example dataset. In this example, there are 3 samples, 3 features and 2 categories.*

5.1 Supervised Learning

Supervised learning is applied on labelled data. Every observation or sample provided in the data must have output label assigned to it as shown in **Figure** *5.2*. We can say the labels help as an immediate feedback during learning process of the supervised learning. Examples of supervised learning include classification and regression. Only classification is discussed below.

5.1.1 Classification

The idea is to identify or predict the category of an object or observation. The dataset consists of a number of features and samples. Each sample contains a value for each feature and a label. The model, trained on past observations, predicts the categorical class labels of new and unseen samples. The machine-learning algorithm learns a set of rules or decision boundaries to distinguish between classes. Classification can be a binary task or multiclass task. If the target variable in a given dataset consists of only two classes, then it is a binary classification problem. In the multiclass classification problem, the target variable in a given dataset consists of more than two classes. There are many classification algorithms but only two of them are described below.

5.1.1.1 Logistic regression

In linear regression, the linear relation and the strength of the linear relation between two variables can be described if their relationship satisfies the following form (Menard, 2002):

$$Y = \alpha + \beta \mathrm{X}, \qquad 5.1$$

where *Y* is the dependent variable being predicted, *X* is an independent, predictor variable that is being used to predict *Y*, α is an intercept on *Y* when $X = 0$, and β represents the regression coefficient, that is slope of the line with the best linear estimation of *Y* from *X*.

In multiple regression, the variable to be predicted can be estimated from several independent predictor variables as follows (Menard, 2002):

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k, \qquad 5.2$$

where k indicates the number of independent predictor variables, and $\beta_1 + \beta_2 + \cdots + \beta_k$ are partial slope coefficients.

Logistic regression is an extension of the linear regression model. The target variable in the linear model is continuous, while that in the logistic regression is categorical. Logistic regression is simple, and predicts well for linearly separable classes. It is one of the most widely used classification algorithms in the industry. Logistic regression, a linear model for binary classification, is extended to multiclass classification using the one vs rest (OvR) technique (Raschka, 2015).

The log of the odds ratio is called the logit function, given as follows:

$$logit(p) = \log \frac{\mathrm{p}}{(1-\mathrm{p})}, \qquad 5.3$$

where $\frac{\mathrm{p}}{(1-\mathrm{p})}$ is the odds ratio, and p is the probability of the event to be predicted. To obtain the probability of a particular sample given its features belonging to a particular class, the logistic function is used. The logistic function (also known as the sigmoid function) is calculated as the inverse of the logit function (Raschka, 2015):

$$\Phi(z) = \frac{1}{1+e^{-z}}, \qquad 5.4$$

where

$$z = \alpha + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k.$$

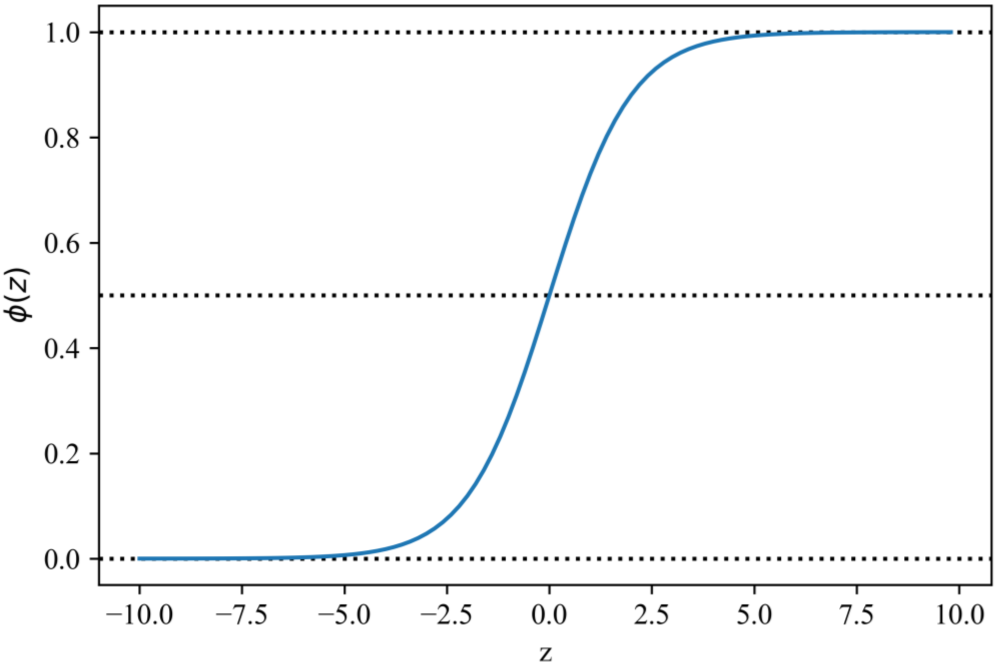


Figure 5.3: *Logistic function*

For any real number input, the output value of this function has the range [0, 1]. In addition, $\Phi(z)$ approaches 0 when z approaches negative infinity ($z \rightarrow -\infty$) because of the large denominator. Moreover, $\Phi(z)$ approaches 1 when z approaches positive infinity ($z \rightarrow \infty$) because e^{-z} becomes very small for large z values.

The binary classification is achieved by converting the predicted probability of the classes with an intercept at $\Phi(z)$= 0.5:

$$\hat{y} = \begin{cases} 1 \ if \Phi(z) \geq 0.5 \\ 0 \ otherwise \end{cases}. \qquad 5.5$$

5.1.1.2 Random forest classifier

A random forest algorithm has become popular owing to its scalability and classification performance (Raschka, 2015). Random forest is an ensemble of decision trees. In ensemble methods, multiple learners are combined into a meta-classifier to achieve good performance and generalisation (Raschka, 2015). We can think of it as combining the opinions from 20 experts to make a prediction, compared to using a prediction made by only one expert. An important step in ensemble learning is to combine the results obtained by the learners using some suitable combination scheme, such as majority voting for classification.

A single base learning algorithm is used mostly in ensemble methods, but sometimes multiple learning algorithms are suitable (Raschka, 2015). In a decision tree algorithm, the base learning algorithm in random forest classification, the breakdown of data is achieved based on decisions made by asking a series of questions as shown in **Figure *5.4***. The goal is to achieve the largest information gain at each split.

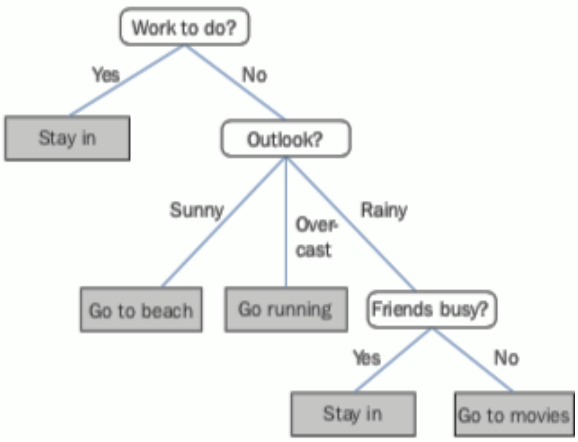


Figure 5.4: *A simple decision tree built to predict which activity is better to do depending on the questions asked on a particular day (Raschka, 2015).*

The steps to implement a random forest algorithm are given as follows (Raschka, 2015):

1. Take *n* samples from the original data by bootstrapping. Bootstrapping is a resampling method used to do random sampling of data with replacement.
2. Build a decision tree for samples (obtained by bootstrapping), where *d* features are chosen randomly without replacement at each node. The node is split based on a feature that results in maximising the information gain.
3. Redo *k* times for Steps 1 to 2.
4. Assign labels using a majority voting scheme. In majority voting, the prediction achieved by each tree is aggregated.

5.2 Unsupervised Learning

If we do not have access to labelled data, the learning is not possible by guidance. The solution to this problem is to identify hidden information in the unlabelled data based on some criteria that might result in depicting different conceptually important results. The main idea behind unsupervised learning is to find underlying patterns and structures in the data. Examples are data clustering, anomaly detection, dimensionality reduction for data compression, etc. Only clustering is discussed in detail below.

5.2.1 Clustering

The main concept of clustering is to group similar data points together based on some criteria. Each group contains objects that are similar to each other compared to objects in other groups. The purpose is to identify and spot the similarities between various data points, which can allow us to classify data based on these similarities. K-means, one of the clustering algorithms, is described in detail below.

5.2.1.1 K-means

The steps to implement k-means are as follows:

1. Specify the number of clusters *k* (assume clusters or one may have an idea of how many clusters are possible).
2. Specify the centre positions of the clusters *k* in the data space. We can randomly specify centres or choose some data points as centres.
3. Each data point is assigned to the nearest cluster centre based on a chosen distance measure, such as Euclidean distance.
4. After assigning all the data points to the relevant centres, the mean of the data points of each cluster is calculated separately. The centres are now moved to the means of the respective clusters.
5. Steps 3 and 4 are repeated until the threshold or required criteria is achieved by algorithm.

A few of the shortcomings of k-means clustering are as follows (Marsland, 2011):

- The number of clusters *k* must be specified.
- The initial position of clusters in the data space can result in very different solutions. The algorithm can stick in local minima.
- As the repositioning of the centres of the clusters depends on the mean of the data points of the respective clusters, k-means is unable to handle outliers effectively.

5.3 Reinforcement Learning

The main idea is to find the optimal policy to accomplish a goal by achieving the maximum accumulated reward. In this learning, the objective is to maximise an accumulated reward. A system (an agent) learns and improves the performance by interacting with the environment via states, actions, and rewards. The reward can be positive or negative for each action depending on the achieved state. This learning does not have instantaneous feedback. The feedback is delayed, and reinforcement learning is also sequential and not independent and identically distributed.

5.4 Building a Machine-Learning System

Machine learning provides a way to make predictive models based on hidden patterns in a historical dataset. As stated previously, the quality and truthfulness of the data are the main prerequisites for machine-learning models. Better quality data result in a better model. Pre-processing of the dataset is sometimes the most time-consuming step, where we convert raw, unstructured data into a meaningful representation that could be fed into machine-learning algorithms for training and testing purposes.

In this project, the focus is to train machine learning models to predict the probability of fouling in heat exchangers. The steps to implement a machine-learning model are shown in **Figure 5.5**.

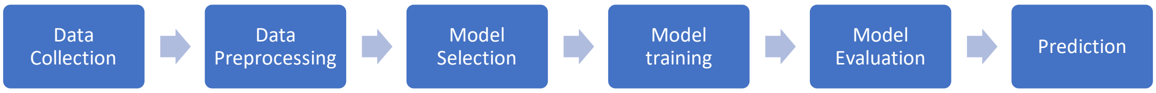


Figure 5.5: *A roadmap for building a machine-learning system.*

5.4.1 Data collection

Before the following steps to implement machine-learning models, a concise description of a problem to be addressed must be defined clearly. Once a problem is defined, the next step is to make decisions regarding data. For example, what kinds of data are required? Where can these data be collected? What will be the best procedure to collect the data? Therefore, the plans are made to collect data. One important consideration should be to ensure that the collection and recording of data must be done using a correct and appropriate method. The data should be reliable and trustworthy. In this project, a small lab-scale system was set up to collect required data as described in Section 6.1.

5.4.2 Data pre-processing

Raw data are rarely available in the form and shape required to feed to machine-learning algorithms. Here are a few important things to remember in pre-processing data:

- Datasets often contain a number of missing values. Sometimes, it is possible to remove samples containing missing values. However, removing those samples can result in the loss of important information. Imputation techniques are used to deal with these values in the proper way to avoid loss of information. A few of the simple imputation techniques are forward filling, backward filling, filling with the mean or median, etc. If we impute values, one must keep in mind that the imputed value may not be the true (missing) value.
- Algorithms can only work on numerical data. If a dataset contains categorical features, those features are converted into dummy matrices.
- More data should be collected if possible and necessary.
- A dataset is split data into training and test datasets to check for generalisation error. The test dataset can usually be 10% to 25% of the total dataset, depending on the number of samples and type of problem.

5.4.3 Model selection

The right choice from the available machine-learning models is an important step. First, we need to determine which type of machine-learning problem (supervised learning or unsupervised learning problem) is related to our dataset. Second, if the problem is a supervised learning problem, we need to determine the type of classification, such as binary or multiclass. Since different models behave differently on different type of problems, the choice matters regarding obtaining sensible results. The preparation and implementation steps are different for numerical data problems, sequential problems (e.g. text), and the image analysis. To cope with different kind of problems, different strategies are needed.

5.4.4 Model training

Iterative learning is achieved in this step. The selected category of models is implemented on training data. The algorithm is allowed to learn on pre-processed data in this step. Each algorithm has certain parameters that are adjusted or tuned to improve the performance of the model. Poor performance could be attributed to overfitting or underfitting of the model or the lack of information contained in the data. In overfitting, the model learns details and random fluctuations too well from the training data. Since this model learned very specific things from the training data, performance of the model is poor on unseen and new data. In underfitting, the model learns poorly from training data. The model does not capture and/or learn enough information from training data and therefore, resulting in poor performance on both training and test data.

5.4.4.1 Model generalisation

Generalisation of a model must be a key point to consider. A model is generalised well if it performs slightly worse on the test data than training data. Poor generalisation results in good performance of the model on training data but poor performance on test data. If a machine-learning model is not generalised well, the reliability and performance of the model on unseen data is compromised. Training data are subdivided into training and validation datasets to determine the performance of the models during training. These datasets are used to avoid poor generalisation of the model.

Overfitting is a common problem. To avoid poor performance and achieve better generalisation, the data are split into three portions: training, validation and test. A model is trained on the training data (usually a major portion of the original data). During training, the performance of the model is tested on the validation dataset. When the model achieves almost equal and good performance on both the training and validation data, the model should have good generalisation. The generalisation of the model is further approved if the model also gives good performance on test data.

To achieve unbiased training, cross validation is used, in which we split the training data into a number of folds (e.g. k-folds). For k-fold cross validation, one fold is used as the validation data while the remaining *k*-1 folds are for training the model. This procedure is repeated *k* times, which results in *k* models and performance estimates. The procedure to use cross validation is shown in **Figure *5.6***.

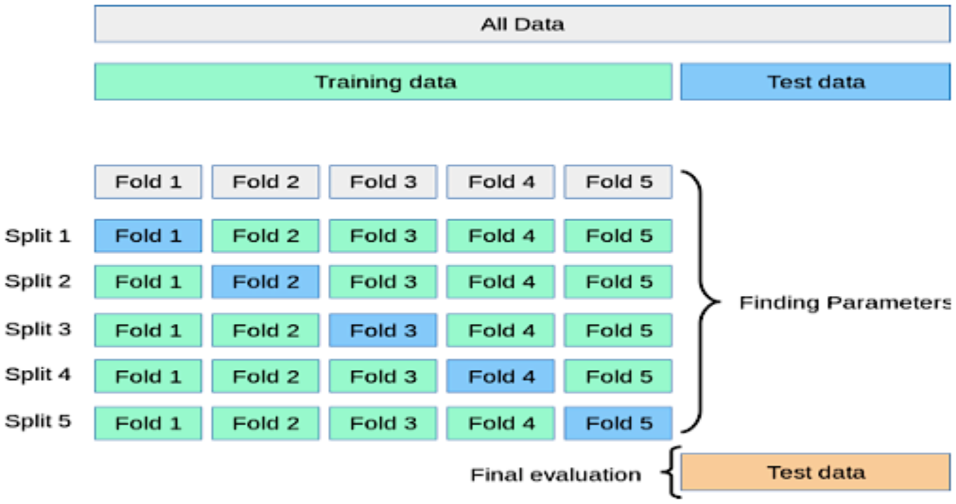


Figure 5.6: *Implementation of cross validation on training data to fine-tune parameters and obtain better generalisation. From training data, five folds are created. In each split, a fold is chosen as the validation data and the other folds are used to train the model. The group of parameters resulting in best average performance on these splits is chosen to evaluate the test data (scikit-learn, 2019).*

When a dataset is very small, nested cross validation (shown in **Figure** *5.7*) is used to avoid generalisation error. In this approach, each data point appears once in the test data. In this way, every kind of variation appears in the test data that might appear in future unseen data. If the performance on nested cross validation is satisfactory, the algorithm will most likely appropriately handle unseen and future data.

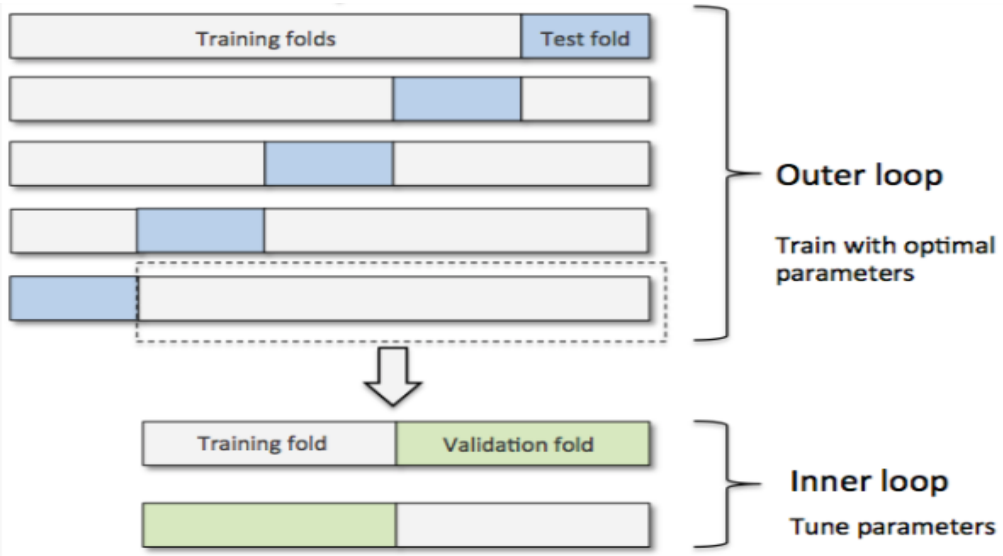


Figure 5.7: *Implementation of nested cross validation on data. In this step, data are split into a number of folds and then assigned as training data, validation data, and test data, as shown above. The hyperparameters are tuned on the training data and validation data. The generalisation is then checked on the test data (Raschka, 2015).*

5.4.4.2 Feature importance permutation

Feature importance permutation is used to differentiate between important and useless noisy features. The important variables significantly affect the decision making of the classification algorithm, whereas noisy features can create disturbances in the resulting predictions that might result in a poor model in later stages. The feature importance permutation method helps to select a few features that are important for a predictive model to perform well.

The steps to implement feature importance permutation are as follows:

1. Train the model using optimal hyperparameters.
2. Calculate the ground truth performance of the model with test data.
3. Randomly permute a feature in the test data and compute the performance of the same trained model on this test data containing a permuted feature.
4. The difference in performance of the model between the baseline and predictions based on the permuted feature gives the feature importance.
5. An increase in performance shows that the permuted feature is important, while a decrease in performance suggests that the permuted feature may be less important.
6. Repeat Steps 3 and 4 until every feature has been permuted.
7. All steps can be repeated to check the performance on different training-test splits to obtain the unbiased feature importance.

5.4.5 Model evaluation

The performance of models is evaluated using some metrics and then is compared. The evaluation of the model depends on the type of problem, such as classification or regression. Our problem is a multiclass classification problem. The metrics used for evaluation in this project are accuracy and the confusion matrix. Sometimes, it is important to know which features are more significant in a given dataset. Permutation importance is helpful in this regard and is discussed below in detail.

5.4.5.1 Accuracy

Accuracy is one of the most common metrics for model evaluation. This is the number of instances predicted correctly divided by the total number of instances to be predicted. Accuracy of a binary classification problem can be calculated by following formula:

$$\frac{\text{number of instances correctly predicted as class 1+ number of instances correctly predicted as class 2}}{\text{total number of instances to be predicted}}.$$

Accuracy can be misleading if the dataset does not contain equal proportions of classes. Accuracy on imbalanced classes does not account well for minor classes in the dataset. It only accounts for the corrected number of predictions, disregarding the fact of majority and minority class proportions.

5.4.5.2 Confusion matrix

A confusion matrix is a square matrix that reports the predictions in a more expressive way. The square matrix describes the counts of the true and false predictions, as shown in **Table *5.1***. The confusion matrix is helpful to view the number of correct and incorrect predictions of the majority and minority classes.

The accuracy can also be calculated from confusion matrix by the following formula:

$$\frac{\mathrm{TP+TN}}{\mathrm{TP+FP+FN+TN}}.$$

Table 5.1: *Confusion matrix*

	Predicted: Positive class	**Predicted: Negative class**
Actual: Positive class	True Positives (TP)	False Negatives (FN)
Actual: Negative class	False Positives (FP)	True Negatives (TN)

5.4.6 Prediction

Once a model is trained, the model is applied on test data. Test data have never previously been seen by the model. If the model achieves good or satisfactory results during training, the next step is to determine its performance on the test data. The good performance on the test data indicates that our machine-learning model is appropriate to use on real-world data.

5.5 Exploratory Analysis

Exploratory data analysis (EDA) is an approach to perform an initial investigation on data using summary statistics and graphical representations. It allows us to understand and grasp the data first before planning to solve a particular problem using machine learning models.

5.5.1 Importance of EDA

Exploratory analysis is performed for the following reasons:

- Detecting faults and inaccuracies (e.g. missing values or anomalies) in a dataset,
- Looking through data to make and test assumptions,
- Determining which models will be appropriate,
- Determining the correlations and relationships among the variables, and
- Assessing a rough estimate of relationships and interactions between explanatory and outcome variables.

5.5.2 Principal component analysis

Datasets usually consist of high dimensions, but visualising high dimensional data is a challenging task. Principal component analysis (PCA) is used here to visualise the behaviour of the 9D dataset in 2D and 3D, respectively. Principal component analysis (PCA), a statistical technique, transforms variables into linearly uncorrelated variables termed as principal components using orthogonal transformation. The PCA is a dimensionality reduction method. The PCA algorithm finds an axis along the direction in which the variance is largest, then locates another axis that is orthogonal to the first axis with next largest variation and continues until all possible axes are found. The resulting new latent variables are uncorrelated.

The steps to implement the PCA algorithm are as follows (Marsland, 2011):

1. Create a matrix $\boldsymbol{X}$ (size = $N*M$) from the vectors $x^i = (x^{1i}, x^{2i}, \ldots, x^{Mi})$ of N data points.
2. Data are centred by subtracting the mean of each column, resulting in a new matrix $\boldsymbol{B}$.
3. The covariance matrix is computed as follows:

$$C = \frac{1}{\mathrm{N}} B^T B. \tag{5.6}$$

4. Eigenvalues and eigenvectors of matrix ***C*** are then computed as follows:

   $$V^{-1} C V = D, \tag{5.7}$$

   where *D* is a diagonal eigenvalue matrix of size *M x M*, and ***V*** contains the eigenvectors of ***C*** and ***D***.
5. The columns of ***D*** are sorted in decreasing order of eigenvalues, and the same order is implemented for the columns of ***V***.
6. After removing dimensions with small eigenvalues, we have ***L*** dimensions left in the data.

6 Methodology

In this section, a lab setup for the experiment is first described briefly, then the software used to achieve the specific goals is discussed, along with their attributes. After that, the processing of data is detailed. In the end, the exploratory analysis of the data and the relationship among the variables in the data are reviewed in detail.

6.1 Experimental Setup

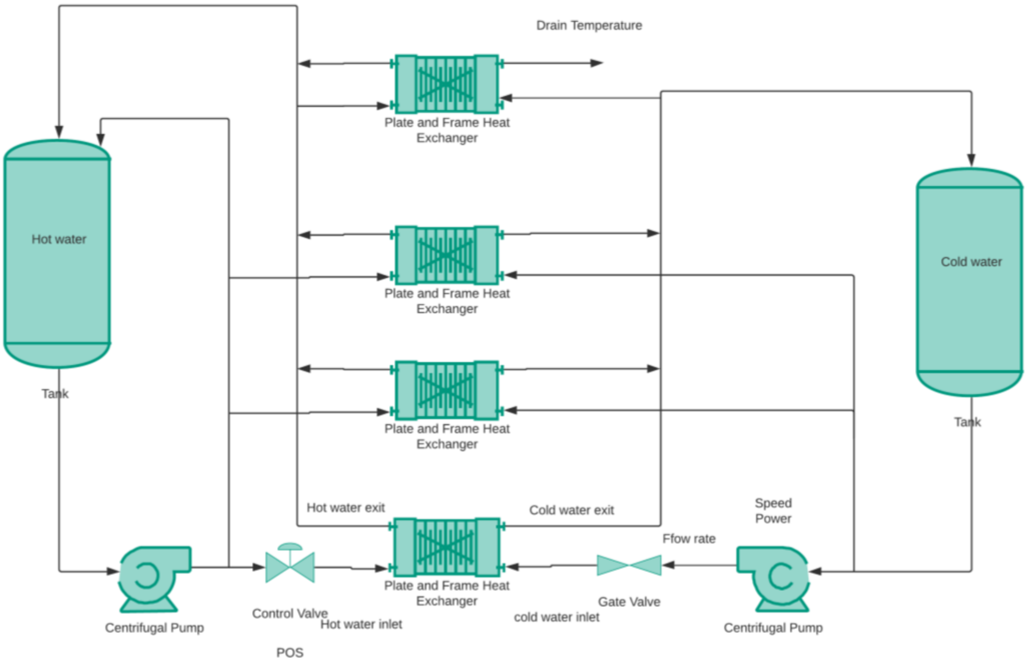


Figure 6.1: *Simple schematic of the lab-scale setup. The data from only one heat exchanger are used for analysis. The effects of the other heat exchangers are ignored during this study.*

A laboratory-scale arrangement was set up as shown in **Figure *6.1***. In the original setup, four heat exchangers are used for exchanging heat. Alfa Laval M3 gasketed plate-and-frame heat exchangers are used to exchange heat between two fluids. However, for the sake of simplification, and as a first step, the dataset obtained from only one heat exchanger (at the bottom in **Figure *6.1***) is used for analysis.

The recorded measurements used as variables/features in this project are listed in Table ***6.1***. For test purposes, the local ground water was used. As the equipment was set up in Denmark, the local ground water is actually hard water. The presence of lime stone or chalk in hard water causes fouling in the heat exchanger. The laboratory-scale system is set up to study the effect of this type of fouling on the heat exchanger performance. A chemical fluid is also mixed in the cold-water stream to further increase the fouling rate. Thus, the process of chalk deposition increased.

Table 6.1: *Key Parameters recorded during the experiment*

Parameter names used in this thesis	Parameters description
Hot-water inlet	Temperature (°C) of hot water entering the heat exchanger
Cold-water inlet	Temperature (°C) of cool water entering the heat exchanger
Drain temperature	Drain Temperature (°C)
Hot-water exit	Temperature (°C) of hot water leaving the heat exchanger after exchanging heat
Flow rate	Distilled water flowrate into heat exchanger (litre/min)
Cold-water exit	Temperature (°C) of distilled water leaving the heat exchanger after exchanging heat
POS	Percentage of valve open for hot-water flow into heat exchanger
Speed	Speed (%) of the pump pushing cold water to the heat exchanger
Power	Power (W) of the pump pushing cold water to the heat exchanger

6.2 Software

Python 3.6 (Oliphant, 2007) is the main programming language used. Pandas, an open source library, was used to handle the dataset. Scikit-learn (Pedregosa, et al., 2011) is a Python library built on NumPy (Oliphant, 2006), matplotlib, and ScipY that provides efficient data mining and analysis tools. Matplotlib (Hunter, 2007) is used here to produce high-quality figures. The packages used along with their versions are listed in Appendix 11.1.

6.3 Formulation of Data

In this project, the data were obtained from heat exchangers operating on a lab-scale setup as described earlier. The dataset consists of four days' measurements. The heat exchanger operates from clean to fully fouled over 4 days. The readings of variables were recorded every minute.

The collected data are processed to feed into the algorithms as follows:

- The first and last few rows were removed to avoid any influence of the initial startup and final shutdown of the heat exchanger on the analysis.
- The system was allowed to run for 15-minute intervals. That means the system operates for 15 minutes, then stops for the next 15 minutes, then again operates for 15 minutes and so on.
- In the dataset, we only included data when the system is in operation, while the remaining data are not used for analysis.

- The dataset was labelled based on the exploratory analysis results and the domain knowledge. The labels are as follows:
    - No fouling (when the heat exchanger operates normally),
    - Some fouling (when the heat exchanger operates with some fouling), and
    - Full fouling (when the heat exchanger is fully fouled).
- We used 15-minute windows for each observation of the sample of data. The maximum of each variable measured during this window is calculated.
- The total data points after pre-processing consist of 130 observations and nine features. The number of observations for each class are as follows:
    - No fouling: 98,
    - Some fouling: 13, and
    - Full fouling: 19.
- Since the transition between no-fouling and some-fouling is important compared to the transition between some-fouling and full-fouling in this project, the three hours of data that represent the transition between the some-fouling and full-fouling stages were removed. In the real world, the transition between the no-fouling and some-fouling stages is important to know. The full-fouling stage actually means the heat exchanger is of no use. As our main purpose in this project is to make a predictive maintenance model, as an early approach, the some-fouling stage is more important to predict than when the heat exchanger is about to become fully fouled or heavily fouled. It is also economically and practically unfeasible to implement maintenance once the heat exchanger is heavily fouled. Since heavy fouling can result in equipment failure, it will eventually stop production. This dataset is used for analysis. The time range for three stages used in this dataset is presented in **Table *6.2***.

Table 6.2: *Time range for stages of operation of heat exchanger*

Stages of operation of heat exchanger	Start time	End Time
No fouling	11.00 on 10th of January	11.45 on 12th of January
Some fouling	12.00 on 12th of January	18.00 on 12th of January
Full fouling	21.00 on 12th of January	06.00 on 13th of January

Table 6.3: *Time range for stages of operation of heat exchanger for 'full data'*

Stages of operation of heat exchanger	Start time	End Time
No fouling	11.00 on 10th of January	11.45 on 12th of January
Some fouling	12.00 on 12th of January	20.45 on 12th of January
Full fouling	21.00 on 12th of January	06.00 on 13th of January

- The behaviour of the data was also observed by including the three removed hours in the some-fouling stage. A few results of this data are also mentioned in the Results and

Discussion section, and important differences are described there. Wherever this dataset is used in this thesis, ‘full data’ is mentioned. The number of observations for each class are:

- No fouling: 98,
- Some fouling: 18, and
- Full fouling: 19.

7 Results and Discussion

The exploratory analysis of data is discussed in Section 7.1. The results obtained from clustering and classification algorithms are then discussed. In the last section, the features that plays role in making predictions are described.

7.1 Evaluation of Dataset

The variation of features with time is shown in **Figure** ***7.1***. Hot-water inlet and cold-water inlet temperature did show very slight change during the 4-day run of the heat exchanger. The pattern for other variables almost remained the same until 12.00 on the 12th of January. The part where the changes in variables are not enormous is assumed to be part of the no-fouling stage.

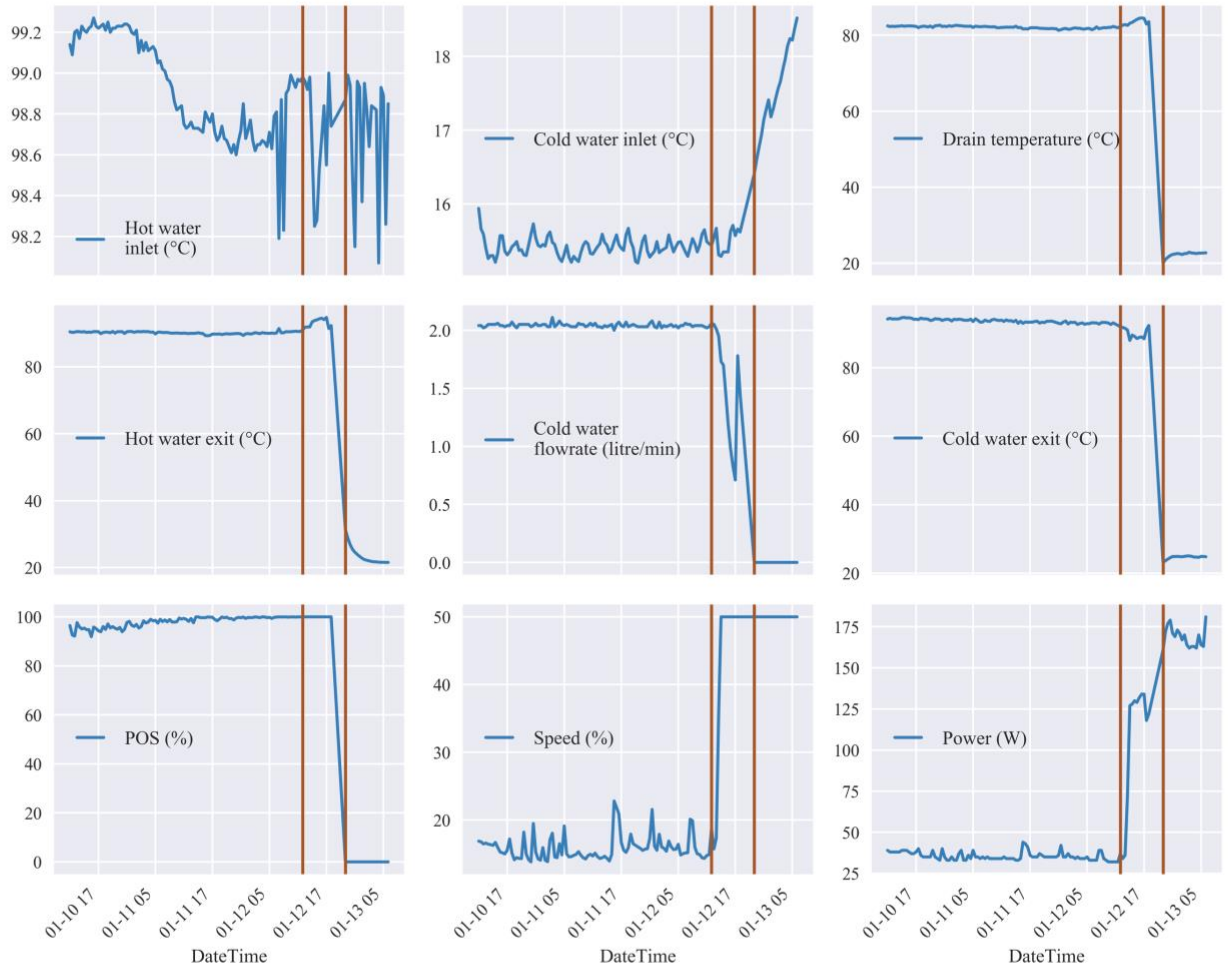


Figure 7.1: *Visualisation of the features recorded in the dataset. The three stages of operation of a heat exchanger are separated by brown vertical lines. The left side represents the no-fouling region, the middle is the some-fouling region, and the right is the full-fouling region.*

In the no-fouling stage, the heat exchanger is working normally. The chances of initiation of fouling at a latter part of the no-fouling stage has most likely occurred, but capturing the initiation of fouling is quite difficult through simple visualisation of variables. After this part, the changes in the values of variables are quite abrupt. The later parts of the dataset are considered the some-fouling and full-fouling stages respectively. The three stages of the heat exchanger performance are shown in **Figure** *7.1*.

7.1.1 Visualisation of principal components

Moreover, PCA is applied to visualise the collective behaviour of the high dimensional data into a three-dimensional PCA scores plot. Since PCA is an unsupervised method, the labels of the dataset were not exposed to the PCA algorithm during the calculation of the principal components. The three clusters are clearly visible in the **Figure** *7.2*. The three clusters of samples indicate the presence of hidden patterns and structures in the dataset. These hidden patterns might be the result of the presence of variations in the three categories. Since we explicitly know the labels of the datasets, the data points are coloured based on these categories. Very few samples of the some-fouling stage appear to be in the no-fouling cluster. Otherwise, the three clusters are well separated.

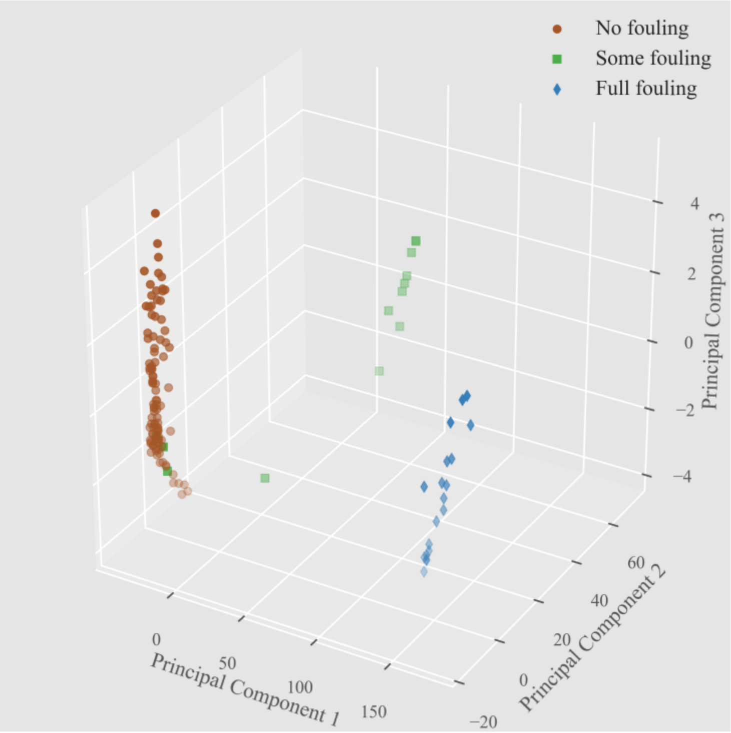


Figure 7.2: *Visualising first three principal components obtained by PCA*

7.1.2 Correlation Matrix

A correlation matrix is a square matrix consisting of the correlation of each variable with all variables in the dataset. The colour bar shows the correlation span between 1 and -1. The value is 1 and -1 when two variables are perfectly correlated and perfectly anti-correlated respectively. Variables are perfectly correlated to themselves as shown in diagonal in **Figure** *7.3*. The value is 0 when there is no relationship between the two variables.

Correlation matrices are different for the three stages of operation of the heat exchanger as shown in **Figure** *7.3*. In full fouling, the light grey colour shows that the variables contributing this phase remain constant (does not show standard deviation) during the full-fouling stage.

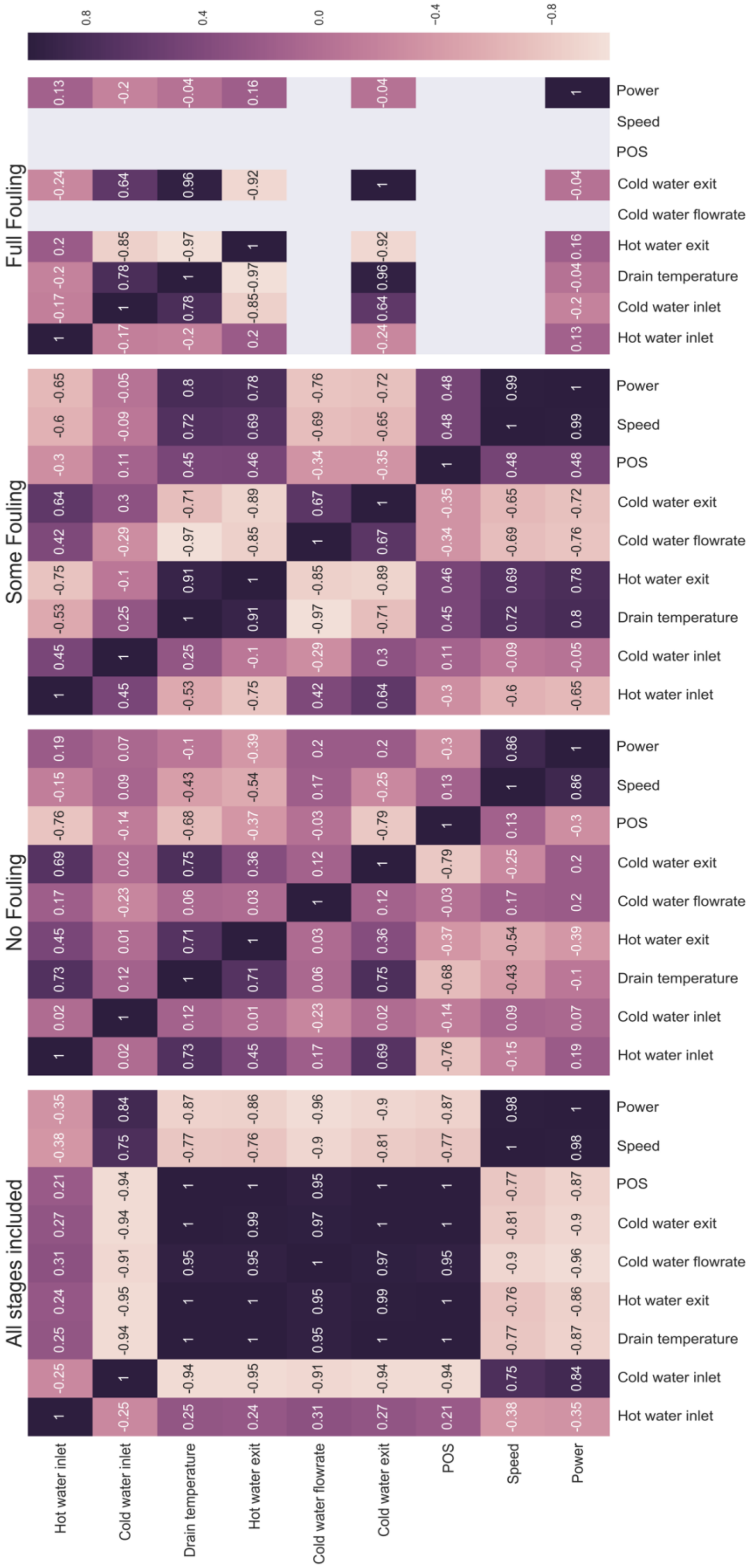


Figure 7.3: *Heatmaps for data (all stages included), no-fouling, some-fouling, and full-fouling stage.*

It is quite evident that different variables have different correlations and that the three stages can be differentiated based on these correlation matrices. Heatmap (all stages included) of the data shows very high (positive or negative) correlations between features. The no-fouling stage heatmap shows only a few high correlations but mostly features lie in the range of low to middle correlations. The some-fouling stage displays strong correlations indicating a different stage of operation of the heat exchanger. The transition of the low correlated features to highly correlated features indicates that the normal operation of the heat exchanger is being hindered. There is a clear difference in the three stages of operation of heat exchanger as evident in **Figure** *7.3*.

7.2 Clustering Algorithm

In clustering, we used k-means clustering, which is an unsupervised learning method. This algorithm is not exposed to our labels while learning, but we explicitly know the labels of each observation. To measure the performance of our k-means model, accuracy was explicitly calculated. In this method, a stratified k cross-validation technique is used in which we considered validation dataset as test dataset and the reported confusion matrix based on samples once they appear in test data. As evident from the confusion matrix in **Table** *7.1*, the prediction of the no-fouling and full-fouling stages is 100% accurate, but four some-fouling observations have been misclassified as the no-fouling class. The interesting result is to find time step at which these misclassified observations occur. It was found that the samples which are predicted wrongly, are right after the real 'no fouling' class ends.

Table 7.1: *Confusion matrix of k-means model*

		Predicted classes		
		No fouling	Some fouling	Full fouling
Actual Classes	No fouling	98	0	0
	Some fouling	4	9	0
	Full fouling	0	0	19

7.3 Classification Algorithms

As classification methods, the logistic regression and random forest classifier are used to build a predictive model. Since the dataset was very small, we do not have test data explicitly. Nested cross-validation is used in classification algorithms to make each data point appear once in test data to reduce bias results as much as possible as described earlier. In this way our model is tested on each sample, and the result was reported in the confusion matrix. Confusion matrix reports the prediction of sample only when it is in test data. Logistic regression predicted three observations that belong to the some-fouling class as the no-fouling class. Meanwhile the random forest classifier predicted only one observation that belong to the some-fouling class as the no-fouling class. The prediction of the no-fouling and full-fouling stages is 100% accurate in both classifiers. Only very few observations belonging to the some-fouling stage are predicted wrongly. One of the interesting observations was that those samples that are predicted wrongly are right after the real no-fouling class ends. The three samples at time steps 2017-01-12 12:00:00, 2017-01-12 12:30:00, and 2017-01-12 13:00:00 are predicted wrongly by logistic regression, while the wrong prediction of the sample at time steps 2017-01-12 12:00:00 was made by the random forest classifier.

Table 7.2: *Confusion matrix of the logistic regression model obtained using nested cross validation.*

		Predicted classes		
		No fouling	Some fouling	Full fouling
Actual Classes	No fouling	97	0	0
	Some fouling	3	10	0
	Full fouling	0	0	19

Table 7.3: *Confusion matrix of the random forest classifier model obtained using nested cross validation.*

		Predicted classes		
		No fouling	Some fouling	Full fouling
Actual Classes	No fouling	98	0	0
	Some fouling	1	12	0
	Full fouling	0	0	19

The algorithms perform quite well but fail to predict the earlier some-fouling stages. The initiation of fouling is the first stage in fouling formation. It is quite hard to distinguish between no-fouling and the initiation stage. As described earlier, during the initiation, the nucleation sites are created that can stay or die depending on the site stability and mass transfer. The change in variables will be very slight in the presence of nucleation sites, but prediction of the initiation step is important, as after initiation, the process of fouling can be pretty fast as described by the linear, falling, and asymptotic curves in **Figure** ***2.5*** and can result in big loss if found later.

The probability of a membership of an observation can sometimes be useful to interpret the strength of belonging to a class. Therefore, the predicted probabilities of classes are plotted to determine whether we can view and capture this early fouling phenomenon. In the logistic regression plot as shown in **Figure** *7.4*, at early timesteps, the probability of the no-fouling stage is 1.0, but as time passes, the probability of that the heat exchanger is working fouling free slightly decreases. These slight fluctuations might indicate the initial fouling layer build-up stages. The fluctuations might be the result of the nucleation sites creation and destruction on the surface of the heat exchanger, providing an early warning for the personnel to look for possible ways to avoid the negative effect of fouling. As it was observed during the experiment, the fouling might have started far before (our assigned) some-fouling stage. According to **Figure** *7.4*, the initiation step of fouling seemed to disturb the operation of the heat exchanger at around 14.00 on 11th of January. The interpretation of this figure could be further emphasised if we had measured during the experiment the presence of fouling using a sensor.

In the random forest classifier plot as shown in **Figure** *7.6*, the probability of the no-fouling stage remains around 1.0 earlier, but the probability of some fouling appeared to be around 20% at around 07.00 on 12th of January. This might give us a strong indication that fouling might have started appearing on the heat exchanger at this point.

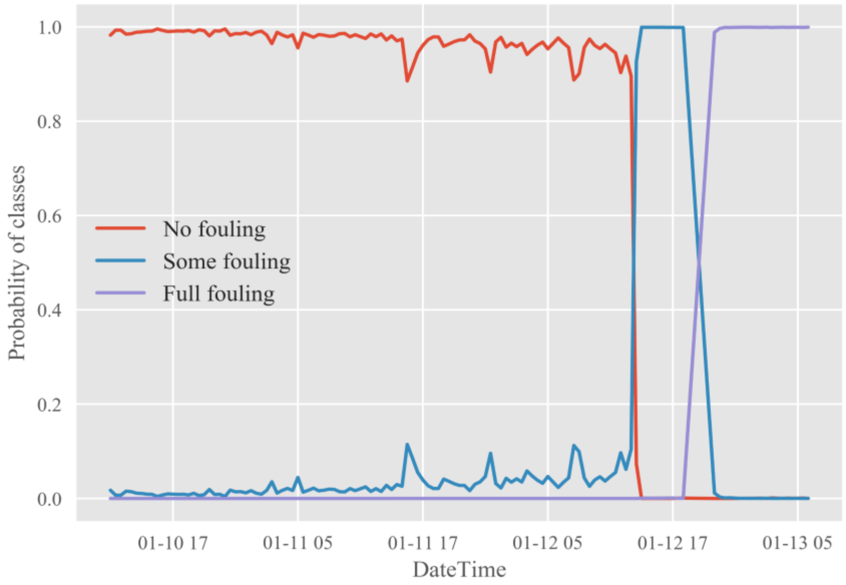


Figure 7.4: *Probabilities of classes using logistic regression.*

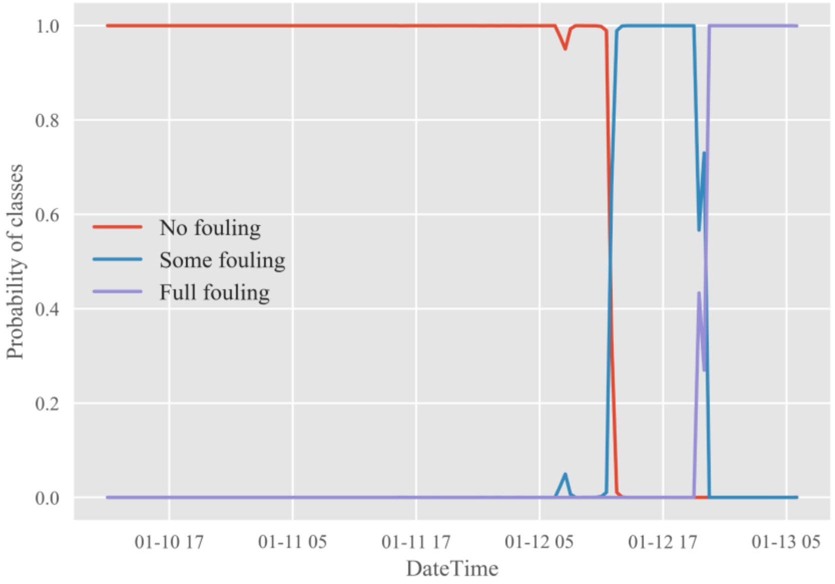


Figure 7.5: *Probabilities of classes using logistic regression with the 'full data' dataset. One interesting thing to note here is that the probability of no fouling does not show any disturbance except once displaying that the early indication of the some-fouling stage is a missing link here. The reason could be the presence of three hours of heavily fouled data in the some-fouling stage. Therefore, the full data are not used for further analysis.*

Logistic regression gives well-calibrated class probabilities (Metzen, 2015), but the random forest classifier can produce a lower misclassification error rate, while the estimation of the class probabilities can be poor (Olson & Wyner, 2018). In SKLEARN (scikit-learn, 2019), the fraction of observations of a class in a leaf gives the class probability of a single tree; therefore, the probability of the predicted class of an observation is calculated as the mean of the class probabilities predicted of the trees in the random forest. Since the reliable and accurate prediction of probabilities helps to give good assessment of risks, the logistic regression model is chosen to analyse the data further.

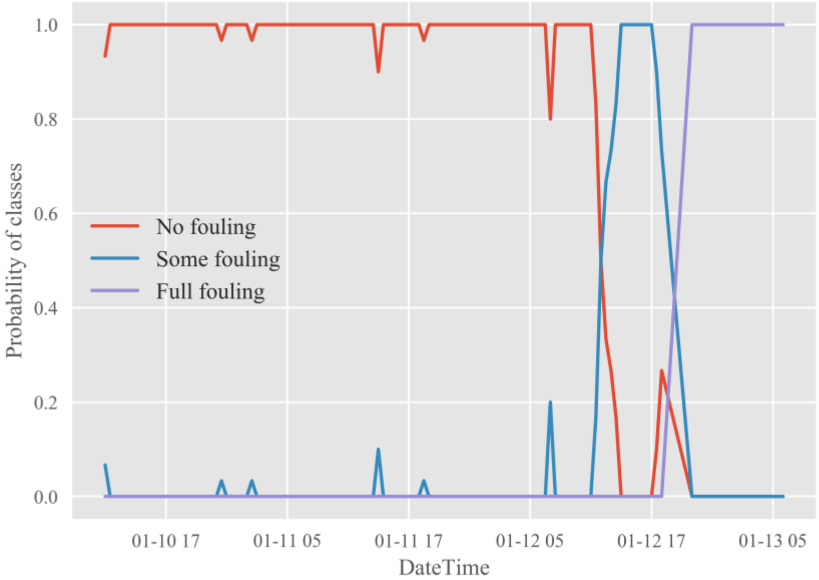


Figure 7.6: *Probabilities of classes using the random forest classifier.*

The prediction performance of model was also tested by reducing number of variables in the dataset. Sometimes, some variables can cause noise in the data. All possible combinations of the three variables are checked to determine the performance of the data. Overall, almost all the combinations give good accuracy results. The highest accuracy of 99.3% was achieved when we used these two combinations of features:

1. Hot-water exit, cold-water exit, and hot-water inlet;
2. Hot-water exit, cold-water exit, and POS.

The hot-water and cold-water exits appeared in both combinations. This also makes sense, as these two variables should be affected when the heat transfer rate in the heat exchanger changes because of fouling. The confusion matrix in **Table *7.4*** shows that the performance of model improved significantly. The confusion matrix is the same for models built on these two combinations of features.

Table 7.4: *Confusion matrix of logistic regression model obtained using nested cross-validation for the three features.*

		Predicted classes		
		No fouling	Some fouling	Full fouling
Actual Classes	No fouling	98	0	0
	Some fouling	1	12	0
	Full fouling	0	0	19

Now, we examine the probabilities of the membership of the classes. The results obtained by only using three features are slightly different and better (in our view) from those obtained earlier using all features.

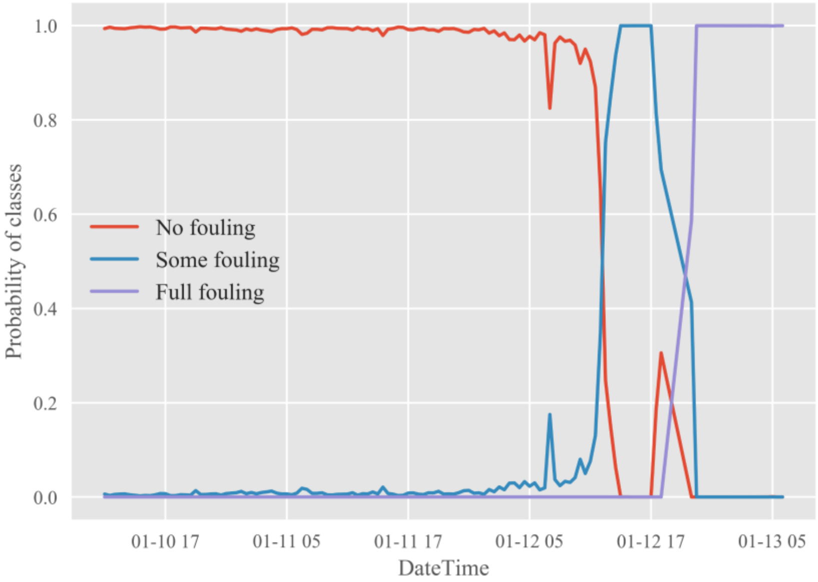


Figure 7.7: *Logistic regression based on three features (hot-water exit, cold-water exit, and hot-water inlet).*

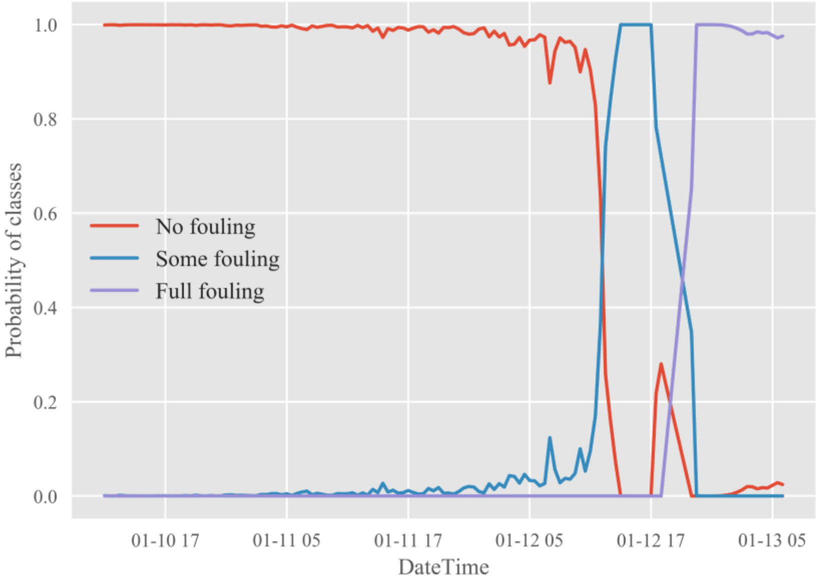


Figure 7.8: *Logistic regression based on three features (hot-water exit, cold-water exit, and POS).*

The real (our assigned) some-fouling stage starts at 12.00 on 12th of January. In our probability plots, the probability that the heat exchanger is working fouling free decreases earlier than the real some-fouling stage. Hence, the predicted probabilities of the strength of membership indicate earlier that the heat exchanger is prone to fouling soon. The probability graphs that we obtained by applying the logistic regression on the three features are smoother and better. These results would have to be confirmed when deploying the model in real life.

7.4 Important Features via permutation importance

As mentioned earlier, permutation importance is used to differentiate between important and useless noisy features. Both noisy and important features impact the predictions, but in different manner. Noisy features give poorer generalisation and poorer prediction performance while important variables do the opposite. The feature importance calculated by feature importance permutation is shown in **Figure** *7.9*. The cold-water exit has the highest importance. The second most important feature is speed. The hot-water exit and POS are also important candidates, but the other variables do not show any significant effect on making predictions. According to the permutation importance, the temperature achieved by the hot or cold water after passing through the heat exchanger gives us a significant indication of the presence of fouling.

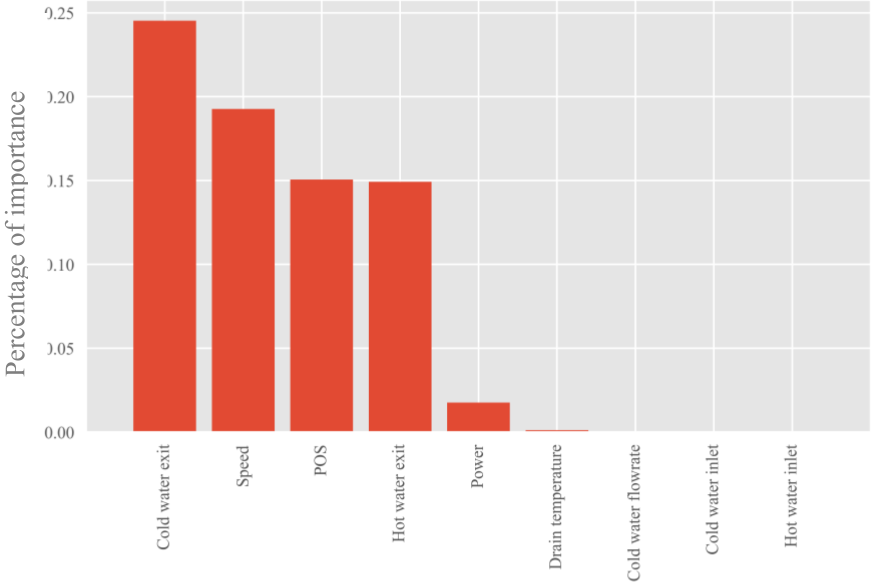


Figure 7.9: *Importance of all features via permutation importance by logistic regression.*

The hot and cold-water exit temperatures in our results appear to be the most important deciding factors in predicting fouling presence. The correctness of our results is further emphasised by the fact that these two parameters are also used in monitoring fouling, as discussed in Section 2.3.7.

8 Conclusion

The phenomenon of crystallisation fouling on plate heat exchangers was studied using the machine learning point of view. The operation of heat exchangers was classified into three stages based on domain knowledge and exploratory analysis, as follows:

1. No fouling (when the heat exchanger operates normally),
2. Some fouling (when the heat exchanger operates with some fouling),
3. Full fouling (when the heat exchanger is fully fouled).

Exploratory analysis by PCA emphasises the presence of hidden information in the data, resulting in three clear clusters. These stages can be differentiated with high accuracy both by clustering and classification methods. Our results highlight that the data collected from sensors can be used to predict when the heat exchanger is not working normally. Logistic regression was chosen as the algorithm for predictive modelling because of its attributes and performance.

The most important part in this study is to predict the point at which the fouling is about to appear and clog the heat exchanger. To achieve this goal, the probabilities of the predicted fouling stages were investigated. As evident from the probability of classes (no fouling and some fouling) predicted by logistic regression, the probabilities show some disturbances before the start of the real some-fouling stage, which indicates the initiation stage. When we reduce the number of feature in the data, the performance of the logistic regression improved, indicating that some variables are causing noise in the prediction. Four features, cold-water exit, hot-water exit, POS, and speed, were evaluated as important by feature importance permutation. Overall, the most important parameters were hot-water exit and cold-water exit in our findings. The results make it clear that the potential to build a high-quality predictive model for fouling is encouraging, despite various limitations in this work.

9 Further Work

Since we do not know the exact time when the roughness delay time actually starts, it will be better to find this period using some measuring equipment during the experiment. It will be interesting to predict the initialisation or induction stage of fouling. After initialisation, fouling sometimes can be rapid and can damage heat exchangers within almost no time. If it becomes important from the industrial point of view to also gain information about when the heat exchanger is operating under heavy fouling conditions, it will be better to classify the operation of the heat exchanger into four stages instead of three stages.

The model may improve with more training data. The given dataset consists of only one experiment of crystallisation fouling in this thesis. For future work, it will be better to collect the data by running a number of experiments.

This project was an initiative towards building a Predictive Maintenance 4.0. The number of variables observed are limited, but definitely not void of useful information. The future objective should be to predict the fouling resistance in a heat exchanger at any given time. To achieve this objective, important influencing variables, such as chemical composition, fouling parameters, and other features as suggested in the research paper (Valiambas, Andritsos, & Karabelas, 1993) would be helpful to gain more specific results.

10 References

Alfa Laval. (2019, April 10). Retrieved from http://www.thermaltransfersystems.com/pdf/alfa-laval-gasketed-heat-exchangers.pdf

Awad, M. M. (2011). Fouling of heat transfer surfaces. In *Heat transfer-theoretical analysis, experimental investigations and industrial systems* (pp. 517-519). IntechOpen.

Bott, T. R. (1995). *Fouling of heat exchangers.* Elsevier.

CSS Electronics. (2019). *Predictive Maintenance 4.0 - Practical IOT Intro for Vehicles & Machinery*. Retrieved April 2019, from https://www.csselectronics.com/screen/page/predictive-maintenance-can-bus-iot/language/en

Geddert, T., Augustin, W., & Scholl, S. (2011). Induction time in crystallization fouling on heat transfer surfaces. *Chemical Engineering & Technology, 34*(8), 1303-1310.

Haarman, M., Mulders, M., & Vassiliadis, C. (2017). Predictive Maintenance 4.0-Predict the unpredictable. *PwC and Mainnovation*.

Hexxcell. (2019). *Hexxcell*. Retrieved April 2019, from https://www.hexxcell.com

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment . *Computing In Science & Engineering*, 90-95 .

Ibrahim, H. A.-H. (2012). Fouling in Heat Exchangers. In V. Katsikis, *MATLAB-A Fundamental Tool for Scientific Computing and Engineering Applications* (Vol. 3). IntechOpen.

IndustryWeek. (2019, March 19). *How Manufacturers Achieve Top Quartile PerformanceUnlocking Performance*. Retrieved March 2019, from Unlocking performance: https://partners.wsj.com/emerson/unlocking-performance/how-manufacturers-can-achieve-top-quartile-performance/

Jamialahmadi, M., & Müller-Steinhagen, H. (2012). Crystallization fouling. *Encyclopedia of Desalination and Water Resources, www.desware.net*.

Jiménez, J. (2018, May 07). *5 Facts You Should Know about Predictive Maintenance*. (May 7, 2018) Retrieved March 2019, from Central IoT: https://www.iotcentral.io/contact

Kakaç, S., Liu, H., & Pramuanjaroenkij, A. (2002). *Heat exchangers: selection, rating, and thermal design.* CRC press.

Kananeh, A. B., & Peschel, J. (2012). Fouling in plate heat exchangers: Some practical experience. In *Heat Exchangers-Basics Design Applications.*

Kazi, S. N. (2012). Fouling and fouling mitigation on heat exchanger surfaces. In *Heat exchangers: Basics Design Applications* (pp. 507-526). IntechOpen.

Khan, M. S., Budair, M. O., Sheikh, A. K., & Quddus, A. (1996). Fouling resistance model for prediction of CaCO 3 scaling in AISI 316 tubes. , 32(1-2). *Heat and mass transfer, 32*(1-2), 73-79.

Marsland, S. (2011). *Machine learning: an algorithmic perspective.* Chapman and Hall/CRC.

Mayer, R., Aziz, T. A., & Rauber, A. (2007). Visualising class distribution on self-organising maps. *International Conference on Artificial Neural Networks* (pp. 359-368). Berlin: Springer.

Müller-Steinhagen, H., Malayeri, M. R., & Watkinson, A. P. (2009). Heat Exchanger Fouling: Environmental Impacts. *Heat Transfer Engineering, 30*, 773-776.

Melo, L., Bott, T. R., & Bernardo, C. A. (Eds.). (2012). *Fouling science and technology (Vol. 145).* Springer Science & Business Media.

Menard, S. (2002). *Applied logistic regression analysis* (Vol. 106). Sage.

Metzen, J. H. (2015, April 14). *Jan Hendrik Metzen*. Retrieved April 2019, from https://jmetzen.github.io/2015-04-14/calibration.html

Oliphant, T. E. (2006). *A guide to NumPy.* USA: Trelgol Publishing.

Oliphant, T. E. (2007). Python for Scientific Computing. *Computing in Science & Engineering*, 10-20.

Olson, M. A., & Wyner, A. J. (2018). Making Sense of Random Forest Probabilities: A Kernel Perspective. *arXiv preprint arXiv:1812.05792.*

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research, 12*, 2825-2830.

Petasense. (2019). *Petasense*. Retrieved April 2019, from https://petasense.com

Prescient & Strategic Intelligence. (2016, January). *Heat Exchangers Market*. Retrieved from https://www.psmarketresearch.com/market-analysis/heat-exchangers-market

Raschka, S. (2015). *Python machine learning.* Packt Publishing Ltd.

scikit-learn, d. (2019). *scikit-learn.* Retrieved from https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation

scikit-learn, d. (2019). *github.* Retrieved from https://github.com/scikit-learn/scikit-learn/blob/7b136e9/sklearn/ensemble/forest.py#L753

Shah, R. K., & Sekulic, D. P. (2003). *Fundamentals of heat exchanger design.* John Wiley & Sons.

Shah, R. K., Subbarao, E. C., & Mashelkar, R. A. (Eds.). (1988). *Heat transfer equipment design.* CRC Press.

Valiambas, S., Andritsos, N., & Karabelas, A. J. (1993). An assessment of data and predictive tools for cooling water fouling of heat exchangers. *Energy Efficiency in Process Technology*, 726-737.

Zhenhua, Q., Yongchang, C. H., & Chongfang, M. A. (2008). Experimental study of fouling on heat transfer surface during forced convective heat transfer. *Chinese Journal of Chemical Engineering, 16*(4), 535-540.

11 Appendices

11.1 Software Versions

Packages	Versions
Python	3.6.8
Matplotlib	2.2.2
Scikit-learn	0.19.2
Pandas	0.23.4
Seaborn	0.7.1
Numpy	1.1.4.1
Mlxtend	0.12.0

11.2 Python Code

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Mon Feb 4, 2019

@author: numanmohammad
"""
####################### import packages #######################

# Basic imports
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# sklearn
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import GridSearchCV, cross_val_score, \
    StratifiedKFold, cross_val_predict
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.ensemble import RandomForestClassifier
from sklearn.cluster import KMeans
from sklearn.metrics import accuracy_score

# other imports
from itertools import permutations
from mlxtend.evaluate import feature_importance_permutation
import warnings

# ignoring warnings
warnings.filterwarnings('ignore')

####################### import data #######################

column_names_ = ['DateTime', 'Hot water inlet',
                 'Cold water inlet', 'Drain temperature',
                 'Hot water exit', 'Cold water flowrate',
                 'Cold water exit', 'POS', 'Speed', 'Power']

excel_file = pd.ExcelFile('fouling_data.xlsx')

# The values are measured at one minute interval
df = excel_file.parse('Sheet1', names=column_names_)

df.set_index('DateTime', inplace=True)
df.drop(df.index[[0, 1, 2]], inplace=True)  # removing inital noise
```

```
####################### Preprocessing data #######################

df_15_minute = df.resample('15Min').max()

# obtaining values when heat exchanger operates
df_exchanger_working = df_15_minute.iloc[::2, :]

# create labels
df_exchanger_working['labels'] = '0'
df_exchanger_working['2017-01-10 11:00:00': '2017-01-12 11:45:00'].loc[:,
                        ['labels']] = 'No fouling'
df_exchanger_working['2017-01-12 12:00:00': '2017-01-12 18:00:00'].loc[:,
                        ['labels']] = 'Some fouling'
df_exchanger_working['2017-01-12 21:00:00':'2017-01-13 06:00:00'].loc[:,
                        ['labels']] = 'Full fouling'

# dropping rows that are not labelled
df_exchanger_working.drop(
    df_exchanger_working[df_exchanger_working.labels == '0'].index,
    inplace=True)

# separating target values from data
data = df_exchanger_working.drop(['labels'], axis=1)
target = df_exchanger_working.loc[:, 'labels']
print('Shape of data: ', data.shape)

target_labels = list(df_exchanger_working['labels'].unique())
targets_col = df_exchanger_working.loc[:, 'labels']

####################### Principal Component Analysis
#######################
pca = PCA(n_components=3)
X_reduce = pca.fit_transform(data)
print('Variance of each principal component: ',
            list(pca.explained_variance_ratio_))

####################### K-means clustering algorithm
#######################
kmeans = KMeans(n_clusters=3, n_init=1000, init='k-means++',
random_state=123)

def y_true_(y_target, set_true_val=[0, 1, 2], shape_of_target=10):
    """
    Converting categories of target variable into numerics.
    """
    s = ['No fouling', 'Some fouling', 'Full fouling']
    y_true = np.zeros((shape_of_target,))
    for j in range(shape_of_target):
        if y_target[j] == 'No fouling':
            y_true[j] = set_true_val[0]
        elif y_target[j] == 'Some fouling':
            y_true[j] = set_true_val[1]
        elif y_target[j] == 'Full fouling':
            y_true[j] = set_true_val[2]
    return y_true

strat_k_fold = StratifiedKFold(n_splits=10, shuffle=False,
random_state=123)

kmeans_ = []
pred_ = []
true_ = []
i = 0

# iterating over 10 folds to make each data point appear once in test
data
for train_index, test_index in strat_k_fold.split(data, target):
    kmeans = KMeans(n_clusters=3, n_init=1000, init='k-means++',
                    random_state=123)
    X_train, X_test = data.iloc[train_index, :], data.iloc[test_index, :]
    y_train, y_test = target[train_index], target[test_index]
    y_pred = kmeans.fit(X_train)
    y_test_pred = kmeans.predict(X_test)

    pred_.extend(y_test_pred)
    y_true = y_true_(y_test, set_true_val=[1, 2, 0],
                     shape_of_target=len(y_test_pred))
    true_.extend(y_true)

print('accuracy score: ', accuracy_score(true_, pred_))
print('confusion matrix: ', confusion_matrix(true_, pred_, labels=[1, 2,
0]))


########### Nested cross-validation for classification algorithms
############

# First find the optimal parameters using grid search
# Then use best parameters and implement cross validation
def nested_cv(train_data, target_data, estimator, params_dict,
              cv_grid=5, rnd_state_list=[123], strat_splits=5):
    """
    Implementing nested cross-validation
    :param train_data: Data to be used for cross-validation
    :param target_data: Target values for the provided data
    :param estimator: Classification algorithm to be used for cross-
validation
    :param params_dict: Parameters to be tuned of the given algorithm
    :param cv_grid: number of splits for grid search
    :param rnd_state_list: list of random state values for reproducible
results
```

```
    :param strat_splits: number of splits required for nested cross-
validation
    :return: pred: predictions of classes for each sample once in test
data
             pred_prob: prediction of probability of classes for each
sample
                        once in test set
    """
    gs = GridSearchCV(estimator=estimator, param_grid=params_dict,
cv=cv_grid)
    gs.fit(train_data, target_data)

    # cross validating by stratifying data first and
    # then implementing cross-val-score
    for n in rnd_state_list:
        strat_k_fold = StratifiedKFold(n_splits=strat_splits,
shuffle=True,
                                       random_state=n)
        # obtaining prediction of probability of classes for each
observation
        # once in test set
        pred_prob = cross_val_predict(gs.best_estimator_, train_data,
                                      target_data, cv=strat_k_fold,
                                      method='predict_proba')

        # obtaining prediction of classes for each observation once in
test set
        pred = cross_val_predict(gs.best_estimator_, train_data,
target_data,
                                 cv=strat_k_fold)

        print('\nClassification Report:\n',
                classification_report(target_data, pred))
        print('\nConfusion matrix:\n', confusion_matrix(target_data,
pred,
                                        labels=['No fouling', 'Some
fouling',
                                                                'Full
fouling']))

    return pred, pred_prob


####################### Logistic Regression ######################

param = {"C": np.logspace(-3, 3, 7),
         "penalty": ["l1", "l2"]}  # l1 lasso l2 ridge
predict_log_r, predict_prob_log_r = nested_cv(data, target,

LogisticRegression(random_state=1234),
                                    param, rnd_state_list=[123],
                                              strat_splits=10)

####################### Random Forest Classifier #######################

n_estimators = [30, 50, 100]
max_features = ['auto', 'sqrt']
max_depth = [int(x) for x in np.linspace(10, 110, num=11)]

param = {'n_estimators': n_estimators,
         'max_features': max_features,
         'max_depth': max_depth}

predict_rndf, predict_prob_rndf = nested_cv(data, target,

RandomForestClassifier(random_state=433),
                                  param, rnd_state_list=[123],
                                            strat_splits=10)

############## Logistic Regression on selected features
####################
predict_log_r, predict_prob_log_r = nested_cv(data.iloc[:, [0, 3, 5]],
target,
                                    LogisticRegression(random_state=343),
                                    param, rnd_state_list=[123],
                                              strat_splits=10)

predict_log_r, predict_prob_log_r = nested_cv(data.iloc[:, [3, 5, 6]],
target,
                                    LogisticRegression(random_state=343),
                                    param, rnd_state_list=[123],
                                              strat_splits=10)

# important features obtained by feature importance permutation
predict_log_r, predict_prob_log_r = nested_cv(
    data.loc[:, ['Cold water exit', 'Hot water exit', 'POS', 'Speed']],
            target, LogisticRegression(random_state=343),
            param, rnd_state_list=[123], strat_splits=10)

####################### Feature Importance Permutation
#######################

strat_k_fold = StratifiedKFold(n_splits=10, shuffle=False,
random_state=123)
splits = strat_k_fold.split(data, target)

feature_perm = []
i = 0
for train_index, test_index in strat_k_fold.split(data, target):
    # print('i: ', i)
    forest = LogisticRegression(C=0.1, penalty='l2', random_state=343)
    X_train, X_test = data.iloc[train_index, :], data.iloc[test_index, :]
    y_train, y_test = target[train_index], target[test_index]
    forest.fit(X_train, y_train)

    # imp_vals the average value of the importance computed from the
    # different runs (num_rounds>1).
    # imp_all contains all individual values from these runs
    # If num_rounds > 1, the permutation is repeated
    # multiple times (with different random seeds)
    imp_vals, imp_all = feature_importance_permutation(
        predict_method=forest.predict,
        X=X_test.values,
        y=y_test,
        metric='accuracy',
        num_rounds=1000,
        seed=1)
    feature_perm.append(imp_vals)
    i += 1

feat_imp = [sum(x) for x in zip(*feature_perm)]
feat_imp = [x / 9 for x in feat_imp]
```