AGH
University of Science and Technology in Krakow

Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering

DEPARTMENT OF AUTOMATICS AND ROBOTICS

DOCTORAL DISSERTATION

ANNA STIEF

COMBINING DATA FROM DISPARATE SOURCES FOR CONDITION MONITORING PURPOSES

DISCIPLINE:
Automatics and Robotics
SUPERVISOR:
Jerzy Baranowski Ph.D
ASSISTANT SUPERVISOR:
James R. Ottewill Ph.D

Krakow 2019

Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie

Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej

KATEDRA AUTOMATYKI I ROBOTYKI

ROZPRAWA DOKTORSKA

ANNA STIEF

ŁĄCZENIE DANYCH Z RÓŻNYCH ŹRÓDEŁ W CELU MONITOROWANIA STANU URZĄDZEŃ PRZEMYSŁOWYCH

DYSCYPLINA NAUKOWA:
Automatyka i Robotyka
PROMOTOR:
Dr. hab. inż. Jerzy Baranowski
PROMOTOR POMOCNICZY:
Dr. James R. Ottewill

Kraków 2019

Abstract

Industrial processes and machinery generate a vast amount of data from a variety of disparate sources which may potentially be valuable for monitoring purposes. The goal of this thesis is to investigate how disparate data available in an industrial setting may enable more reliable and robust condition assessment. Feature design and selection is investigated as it is one of the first steps towards accurate fault detection and diagnosis. Feature selection methods are reviewed from the perspective of their applicability for condition monitoring and data fusion problems. The ReliefF method, which has been found to be a suitable fit for condition monitoring applications, is further studied and extended to cope with feature redundancy. A ReliefF-based hybrid method is created for feature selection. The thesis also investigates new algorithms to fuse data from disparate sources recorded online, offline, and periodically for equipment condition monitoring. A generic two-stage Bayesian framework is developed, which is composed of a feature-level fusion and a decision-level fusion of the feature-level fusion results. Feature-level fusion is implemented with Naive Bayes classifiers. Thresholds-based likelihood functions, Gaussian likelihood functions, Kernel Density Estimation or a newly developed Interpolated Kernel Density Estimation technique may be used for the feature-level Bayesian fusion depending on the condition monitoring data and system. Decision-level fusion is conducted with a Naive Bayes formulation using confusion matrices. Furthermore, two methods are proposed to account for the operating condition dependency of features when using the two-stage Bayesian framework, which is a typical condition monitoring challenge. The new methods are validated through multiple applications on two case studies containing heterogeneous data obtained from induction motors and a multiphase flow facility. The results confirm that the methods improve the diagnostics performance, while creating a robust, modular and scalable monitoring framework.

Streszczenie

Procesy przemysłowe i maszyny generują ogromną ilość danych z wielu różnych źródeł, które potencjalnie mogą być wartościowe dla celów monitoringu i diagnostyki. Celem tej pracy jest zbadanie, w jaki sposób różne dane dostępne w warunkach przemysłowych mogą umożliwić bardziej wiarygodną i solidną ocenę stanu systemu. Badania obejmują dobór i selekcję cech sygnałów, ponieważ jest to jeden z pierwszych kroków w kierunku dokładnego wykrywania błędów i diagnozy. Metody selekcji cech są badane z perspektywy ich przydatności do monitorowania stanu i problemów z fuzją danych. Metoda ReliefF została przeanalizowana i rozszerzona o mechanizmy kompensacji redundancji cech. W celu wyboru cech utworzono metodę hybrydową wykorzystującą ReliefF. W pracy zbadano również nowe algorytmy łączenia danych z różnych źródeł zarejestrowanych online i offline. Opracowano ogólną dwustopniową strukturę Bayesowską, która składa się z fuzji na poziomie cech i fuzji wyników fuzji na poziomie decyzji. Fuzja na poziomie cech jest implementowana naiwnymi klasyfikatorami Bayesowskimi. W przypadku fuzji bayesowskiej na poziomie cech można użyć funkcji wiarygodności opartych na progach, funkcji wiarygodności Gaussa, estymatorów jądrowych lub nowo opracowanej techniki interpolowanych estymatorów jądrowych w zależności od danych monitorowania stanu i systemu. Fuzja na poziomie decyzyjnym jest przeprowadzana z użyciem formuły naiwnego klasyfikatora Bayesowskiego przy użyciu macierzy konfuzji. Ponadto proponuje się dwie metody, uwzględnienia zależności cech od warunków pracy przy użyciu dwustopniowej struktury Bayesowskiej, która jest ważnym zadaniem monitorowania stanu. Nowe metody zwalidowanow różnych zastosowaniach dla dwóch studiów przypadku zawierających heterogeniczne dane na temat silników indukcyjnych i wielofazowych instalacji przepływowej. Wyniki potwierdziły, że metody poprawiły wydajność diagnostyki, tworząc solidne, modułowe i skalowalne struktury monitorowania.

Acknowledgement

I am grateful for the funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 675215 under the PRONTO project, for the financial support.

I am thankful for the ABB Corporate Research Center for hosting me during the PRONTO project. I would like to express my gratitude to my industrial supervisor Dr. James Ottewill for the continuous support, for his patience, persistence, and immense industrial knowledge. His guidance helped me in all the time of research and writing of this thesis. I am thankful for my line manager Mariusz Wojcik, who always kept me challenged, and for other ABB colleagues, especially for Marcin Firla, Piotr Lipnicki and Michal Orkisz, from whom I learned a lot during the years spent in ABB.

I am thankful for AGH University of Science and Technology for hosting me during my secondment. My sincere thanks go to my academic supervisor Dr. Jerzy Baranowski without whom this thesis would not have been possible and who always provided great theoretical insights for my research.

Very special gratitude goes out to all the participants of the PRONTO Network. My genuine thanks also go to Professor Nina Thornhill, who was the project coordinator of PRONTO and who gave me many good tips and advice research-wise and career-wise. I would like to thank my fellow doctoral students for their feedback, cooperation and of course friendship, they were a great company during the project. I am especially thankful for Ruomu Tan for conducting together the multiphase flow facility case study and being an always cheerful roommate in the ABB office.

I am thankful for Cranfield University for hosting me during my secondment and providing me with the opportunity of conducting experiments on the multiphase flow facility. I would like to thank my host Professor Yi Cao, who provided me with good advice during my secondment. I am also thankful for Liyun Lao, Stan Collins, Henry Tandoh, Prafull Sharma and Godfrey Nnabuife for providing helpful insight to the design and the implementation of the experiment.

I am also grateful for the following people who carried out the measurement campaign for the induction motor case study: Maciej Sułowicz, Konrad Weinreb, Janusz Petryna, Arkadiusz Dziechciarz from Cracow University of Technology and Wojciech Batko, Maciej Kłaczyński, Jacek Wierzbicki, Tadeusz Wszołek, Jacek Frączek from AGH University of Science and Technology.

Contents

1. Introduction

The drive for increased performance stretches operation boundaries. This leads to greater risk of component failures, therefore Condition-based maintenance (CBM) is becoming ever more important. It is a decision-making strategy which aims to optimise asset and plant availability, allowing corrective maintenance measures to be taken on the basis of the actual condition of a component, while at the same time aiming to keep maintenance costs as low as possible. CBM can also potentially improve operational safety and reliability, as the method may be used to forecast failures. However, monitoring systems have to be developed carefully in a way that false and missed alarm rates remain low. False alarms can cause unnecessary maintenance actions and reduce the trust in a monitoring system, whilst missed alarms can lead to failures and unplanned downtime.

Different sensors may be used to monitor the health state of a system with different sensor types, as one sensor might be more adept at detecting one fault or operation mode than another sensor. One feature derived from a signal recorded from a particular sensor might be capable of detecting one type of fault, while a different feature calculated from the same source might be more successful at detecting a different type of fault. Hence, feature design and selection is one of the first steps towards successful fault detection and diagnosis.

Industrial processes and machinery can now generate a vast amount of data from a variety of disparate sources, each of which may potentially be valuable for CBM. Condition monitoring approaches which fuse data from multiple sensors have the potential to diagnose faults with reduced false and missed alarm rates. Data may not only take the form of a time-domain sensor data. Alarm and event logs, maintenance logs, design data, connectivity, and topology information may also be used as the input of the monitoring system. New process and condition monitoring techniques need to be developed to tackle the new challenges of heterogeneous data and combine them in a way which leverages their strengths and suppresses their limitations.

The goal of the thesis is to develop novel methods to combine data from disparate sources recorded online, offline, and periodically in an automated way for equipment condition monitoring. Incorporating data from a greater number of diverse sources can enable a more reliable and robust condition assessment. Condition monitoring approaches which fuse data from multiple sensors and sources have the potential to diagnose faults more accurately than using only a single source of information. Disparate data can contain complementary information regarding the health state of the monitored system, therefore their fusion can improve the results of fault diagnostics. The work objectives based of this thesis are as follows:

- Determination of algorithm requirements with reference to data typically available in an industrial setting.

- Development of methods for performing feature selection from high dimensional datasets.

- Development of data fusion methods for fault classification using multivariate statistics and Bayesian reasoning.

- Development of methods for fusing disparate data types (for example binary data, such as alarms fused with sensor measurements).

The newly proposed methods are validated on two case studies, one for monitoring a commonly used component of industrial processes and one for monitoring an entire process plant. The two case studies were selected to show that the proposed methods are both applicable for component level and plant level monitoring. A component of a system may be monitored by several sensors, while a complex process plant may be monitored using various monitoring systems. Both case studies contain heterogeneous data, which are as follows:

- Component level monitoring - An induction motor case study. Different types of sensor data (vibration, acoustic and electric data) are available from both healthy and faulty induction motors operating under different environmental and loading conditions.

- Plant level monitoring - A multiphase flow facility case study. Disparate data (alarms, process measurements, high-frequency pressure, and ultrasonic data) are available under different operating conditions recorded from an industrial scale multiphase flow facility both with and without seeded faults.

The thesis is structured as follows. Chapter 2 gives an overview of the condition monitoring literature, with special focus on disparate data found in process plants. This chapter also highlights the challenges and opportunities of current state-of-the-art condition monitoring methods and translates these into requirements for industrial condition monitoring systems. Chapter 3 reviews the fundamental terms of data fusion, abstraction levels, methods and challenges of data fusion focussing on its applicability to the field of condition monitoring. This chapter also summarizes the advantages of data fusion, outlining directions for research for the thesis. Chapter 4 discusses the importance of feature selection with relevance to data fusion and condition monitoring applications. Two newly developed extensions of the ReliefF feature ranking method are proposed to account for removing correlated features and provide a feature selection framework. Chapter 5 reviews Bayesian methods for data fusion and for condition monitoring applications. A generic two-stage Bayesian framework is proposed, which fuses data from disparate sources on the feature- and on the decision-level for more accurate, transparent and scalable condition monitoring. Chapter 6 describes the component- and plant- level monitoring case studies. Chapter 7 shows six different applications of the proposed methods validated both on the component- and plant- level monitoring case studies. Chapter 8 summarizes the contributions and findings of the thesis, discusses the advantages and limitations of the newly proposed methods and opens up future research directions before a final conclusion is given. The Appendix contains the list of publications from the thesis and the description of the PRONTO project.

2. Condition-based maintenance

In this chapter condition-based maintenance is introduced. Important condition monitoring terms and concepts are defined which are widely used both in academia and in industry. Motivations, strategies, and the tasks of condition monitoring are also introduced. The chapter also gives an overview of its applications and discusses the economic feasibility of maintenance. The data sources are reviewed that may contain relevant information for decision support systems and may be used in condition-based maintenance. An overview of the design and operation of condition monitoring systems is provided. The chapter concludes with the challenges and opportunities of condition monitoring with a focus on the requirements of modern condition monitoring systems.

2.1. From "run to failure" to condition-based maintenance

2.1.1. Motivation to perform maintenance

Efficient operation of large scale industrial processes is key to achieving high yield, reduced maintenance actions and improved performance. A process plant is in *normal operation* when it is able to fulfil the desired function satisfactorily and effectively (Tidriri et al., 2016). A *fault*, which is defined as an unpermitted deviation of at least one characteristic property or parameter of the system from the normal operation (Isermann, 2006), might lead to *failure*, which is defined as the permanent inability of the system to perform a required function under specified operating conditions (Isermann, 2006).

Plants can be composed of several subsystems, such as electrical, process and mechanical subsystems, each composed of several components. Component reliability is a key aspect of fault-free operation, however all components will degrade during their lifetime. *Degradation*, which is defined as a "detrimental change in physical condition, with time, use, or external cause" (BS EN 13306:2010), is a normal physical phenomenon which can be due to operation under certain load, mechanical fatigue or environmental conditions such as temperature and humidity, often involving a high degree of randomness (Jardine et al., 2006). Once degradation has reached a certain level or there is a failure, maintenance actions are required to find the faulty component, renew, repair or replace it and finally return the process to normal operation.

Due to the growing need for safer, more reliable and more predictable industrial processes operating at their maximum performance, the role of maintenance is becoming more important. Plants are often instrumented with a large number of sensors allowing continuous supervisory control of the process. A well-chosen maintenance strategy which takes into account the current health state and operating conditions of the process can increase the safety of the plant, minimize downtime, maximize efficiency and reliability while reducing operation and maintenance costs.

2.1.2. Maintenance strategies

There are three maintenance strategies described in the literature: *corrective maintenance*, *predictive maintenance* and *condition-based maintenance* (Randall, 2011).

- **Corrective maintenance**: The earliest maintenance strategy was corrective or breakdown maintenance, sometimes referenced as "run to failure". This approach operates without planned maintenance. As no maintenance is done during the operation, there is no counter-measure against low system efficiency. Once a failure happens, the system is shut down and maintenance has to be scheduled to repair or to exchange the faulty element resulting in unplanned downtime. If the faulty element is not available or the fault is too serious, the time to repair might significantly increase and the production could stop for a longer period resulting in a large production loss to the operator (Jardine et al., 2006). The root cause of failure might originate from a component which is easy to replace and which could have been repaired before a costly failure and shut down happened. In general, corrective maintenance is not applied anymore in modern safety-critical industrial facilities, however, it can be a good strategy for components which are cheaper to monitor and regularly maintain (Randall, 2011).

  Figure 2.1 shows a typical corrective maintenance scenario (Jaramillo et al., 2014). After the wear-in, the physical condition of the asset starts to degrade. When a fault happens, it remains unnoticed. Later the fault degrades until a failure, which leads to unscheduled maintenance and downtime. Once corrective maintenance actions have taken place, the component is replaced and the system is made operational. In this case, there is a cost only when the failure occurs.

- **Preventive maintenance**: A more advanced maintenance strategy is preventive maintenance, which has been applied in industrial practice since the 1950s. The idea behind this strategy is to schedule periodic maintenance on the system to check if there is any sign of a fault, and if there is, to repair it immediately. The advantage of the method is that there are a the fewer number of serious failures compared to the corrective maintenance strategy and it is relatively cheap when the maintenance interval is well-set. The disadvantage of the approach is the possibility of unnecessary downtime, as sometimes, even if there is no fault, the system is shut down for maintenance (Jardine et al., 2006; Randall, 2011). There was some effort made by Jardine and Tsang (2005) to determine the optimal interval of maintenance checks based on reliability and cost data. However, often there is not enough prior knowledge available for such analysis.

  Figure 2.2 shows a typical preventive maintenance scenario. In this case, the system is stopped periodically according to a predefined plan. Costs occur at every preventive action.

- **Condition-based maintenance**: The development of computer science, sensor technologies and new standards of reliability and maintenance has led to the growth of condition-based maintenance (CBM) strategies. It is a decision-making strategy which aims to optimize asset and plant availability by incorporating information and insights provided by the condition monitoring (CM) system into the decision making process.

  CBM seeks to optimize availability by incorporating information and insights provided by the CM system into the decision making process (making more informed decisions).

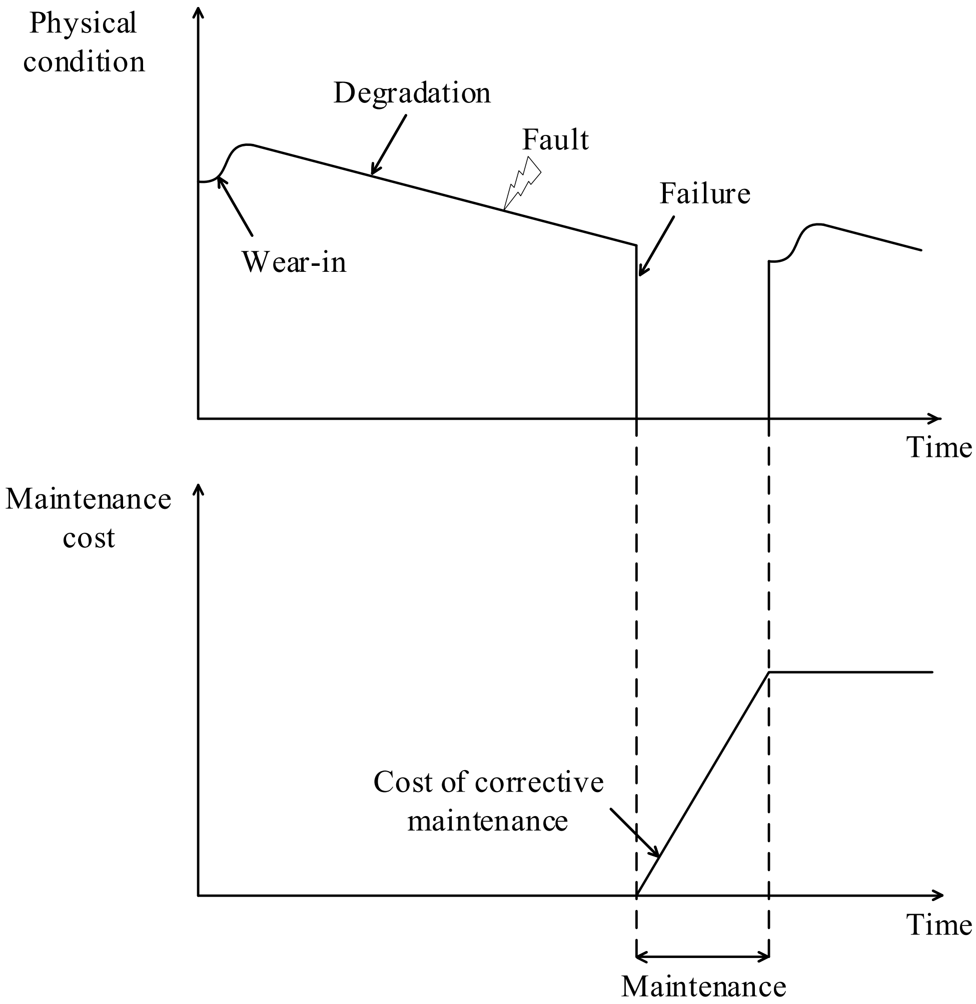


Figure 2.1: System performance and maintenance costs for a corrective maintenance strategy (Jaramillo et al., 2014)

> At the same time, CBM aims to keep maintenance costs as low as possible (Peng et al., 2010; Kan et al., 2015). By reducing unnecessary maintenance actions the downtime of the system can be minimized. CBM can potentially improve operational safety and reliability, as the method may be used to forecast failures. If the CM system shows an alarm, maintenance can be scheduled to avoid a sudden failure, which could otherwise lead to unplanned downtime and increased maintenance costs. Although this is an effective strategy to reach safe operation and minimal downtime, the installation and accurate set-up of the CM system can be costly and requires initial engineering effort and maintenance during operation.

Figure 2.3 shows how condition-based maintenance may be related to the physical condition of the asset and to maintenance costs (Jaramillo et al., 2014). The CBM system has an initial engineering effort with a cost. Once the condition monitoring system detects a fault, a short shut down period may be planned along with a focused maintenance action. If a component shows low reliability or efficiency in the long run, there might be another planned shut down to upgrade the components.

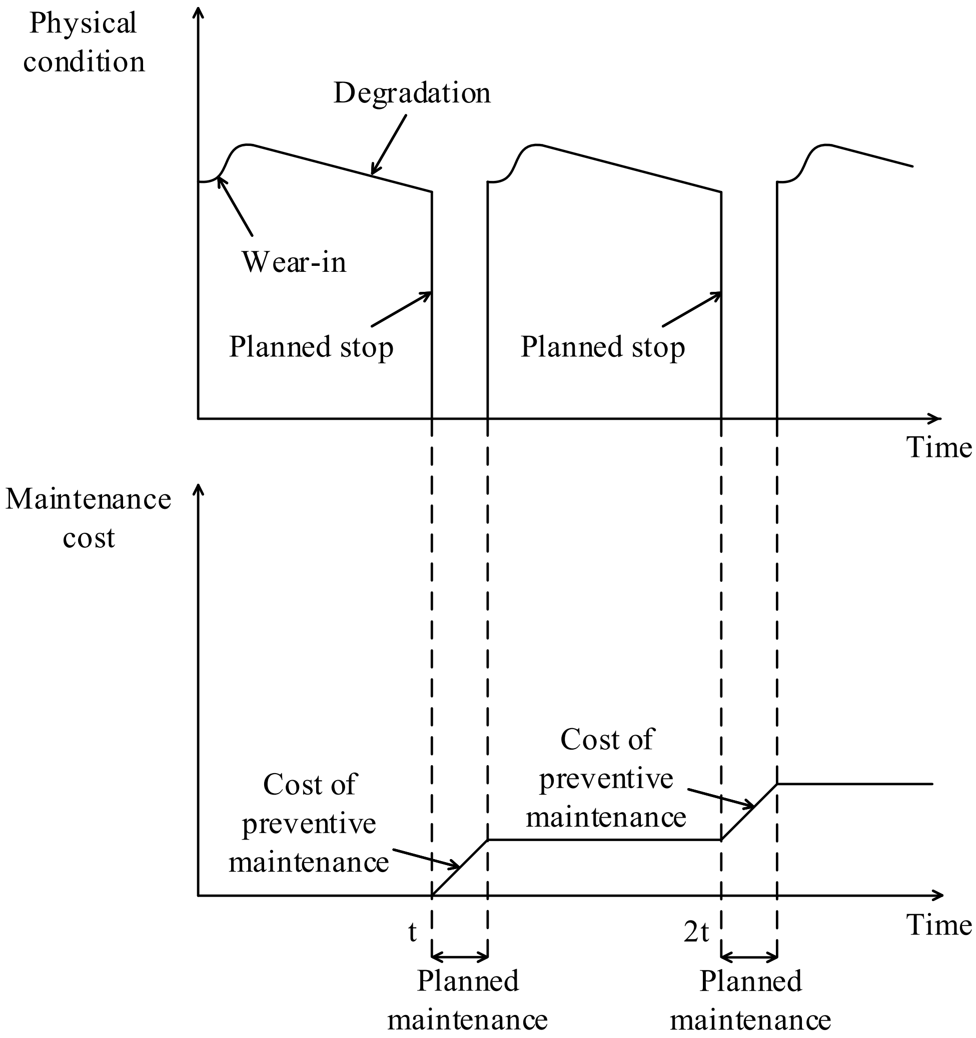


Figure 2.2: System performance and maintenance costs for a preventive maintenance strategy (Jaramillo et al., 2014)

All of the three above mentioned strategies have their own application areas where they are successfully implemented. Sometimes they are used in combination within one plant when some components are easy to replace and can run till failure. Other components are maintained regularly, while operation critical equipment is continuously monitored and even in case of the first signs of degradation they are maintained. Throughout the rest of the thesis concepts and methods connected to condition monitoring for CBM will be explored, developed and tested.

2.1.3. Condition monitoring as the key to condition-based maintenance

Condition monitoring is an essential element of a condition-based maintenance program. The possible industrial applications of condition monitoring are very wide, from component-level to plant-level monitoring applications. Table 2.1 provides an overview of the application fields where condition monitoring has been actively applied. Table 2.1 also contains a collection of review papers, which provide

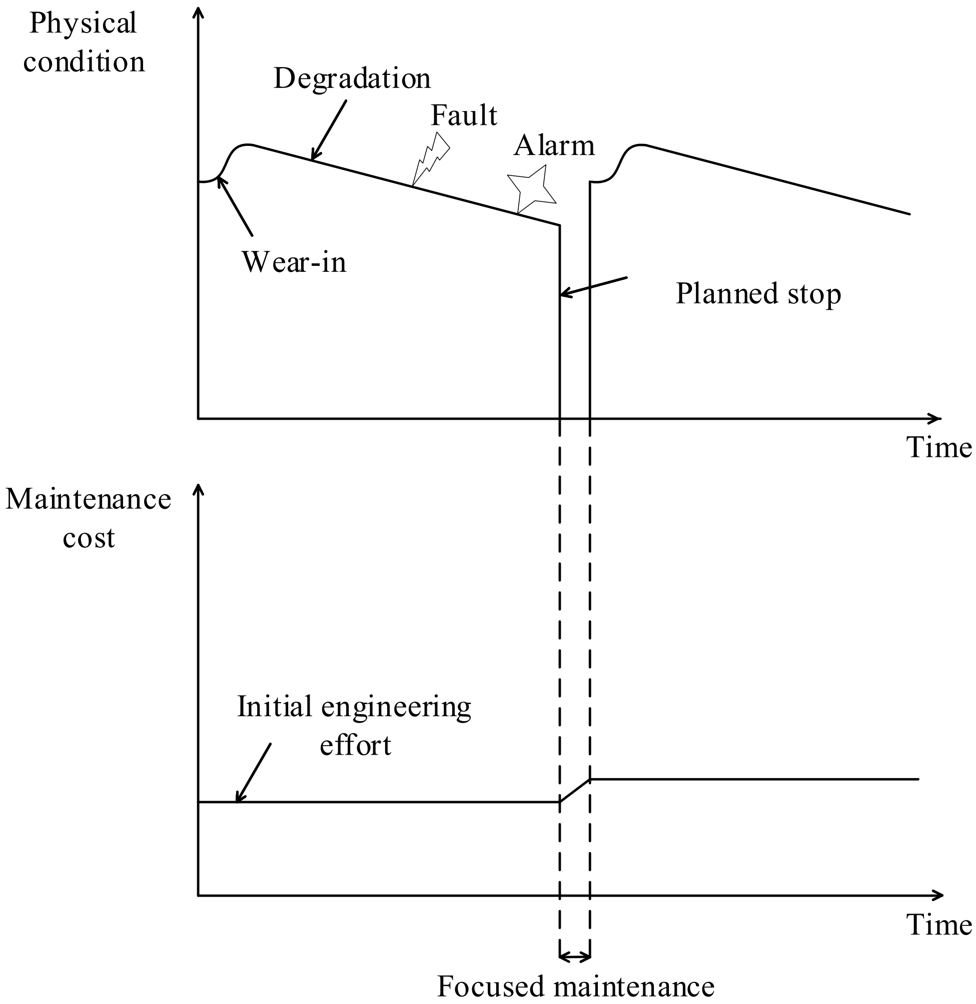


Figure 2.3: System performance and maintenance costs for a condition-based maintenance strategy (Jaramillo et al., 2014)

summaries of how CM has been applied in each application fields.

Condition monitoring has two main tasks based on the collected data from the system, *diagnostics* and *prognostics*. Diagnostics determines the current health state, while prognostics deals with the possible future health states and degradation of the system and its components. Diagnostics has a three-fold task: the indication if there is a fault present in the system is the task of *fault detection*; the location of the faulty component is the task of *fault isolation*, and the determination of the nature of the fault is the task of *fault identification*. Often the three tasks of diagnostics are referred to in the literature as *fault detection and diagnostics* (FDD). Once the current health state has been determined by diagnostics, prognostics can estimate the *remaining useful lifetime* (RUL) , the probabilities of possible failure modes and the confidence intervals of the predicted probabilities. Figure 2.4 gives an overview of the above-described terms.

Table 2.1: Application fields where condition monitoring is actively applied with reviews from each field

Application field	Reviews from the field
Electrical motors	Mehrjou et al. (2011) Nandi et al. (2005)
Compressors	Li and Nilkitsaranont (2009) Schultheis et al. (2007)
Gearboxes	Liang et al. (2018) Goyal et al. (2017)
Wind turbines	Liu et al. (2015) Márquez et al. (2012)
Transformers	de Faria Jr et al. (2015) Saha (2003)
Oil and gas	Pedersen et al. (2015) Natarajan and Srinivasan (2010)
Automotive	Mujahid and Dickert (2012) Bodensohn et al. (2005)
Locomotive	Yan et al. (2015) Newman et al. (1988)
Aviation	Caliskan and Hajiyev (2013) Bonfe et al. (2006)
Nuclear power plants	Ma and Jiang (2011) Gillen et al. (1999)
Robotics	Emran and Najjaran (2018) Dixon et al. (2000)
Manufacturing	Goyal and Pabla (2015) Kalogirou (2003)

As modern industrial plants are complex, expensive and often well-instrumented, there is increasingly more interest in testing, developing and using CBM systems. CBM has a number of advantages compared to corrective and preventive maintenance. It can offer:

- Reduced downtime with maximized operating hours targeting continuous production
- Increased plant efficiency and performance
- Safe operation with reduced risk of emergency shutdowns and catastrophic failures
- Detection and diagnosis of the root cause of a fault. The maintenance can be scheduled in a timely manner, with pre-ordered parts and pre-arranged maintenance personnel, who can quickly target the faulty component.
- Indication of components with lower reliability. Replacing them can ensure the increased overall reliability of the plant.

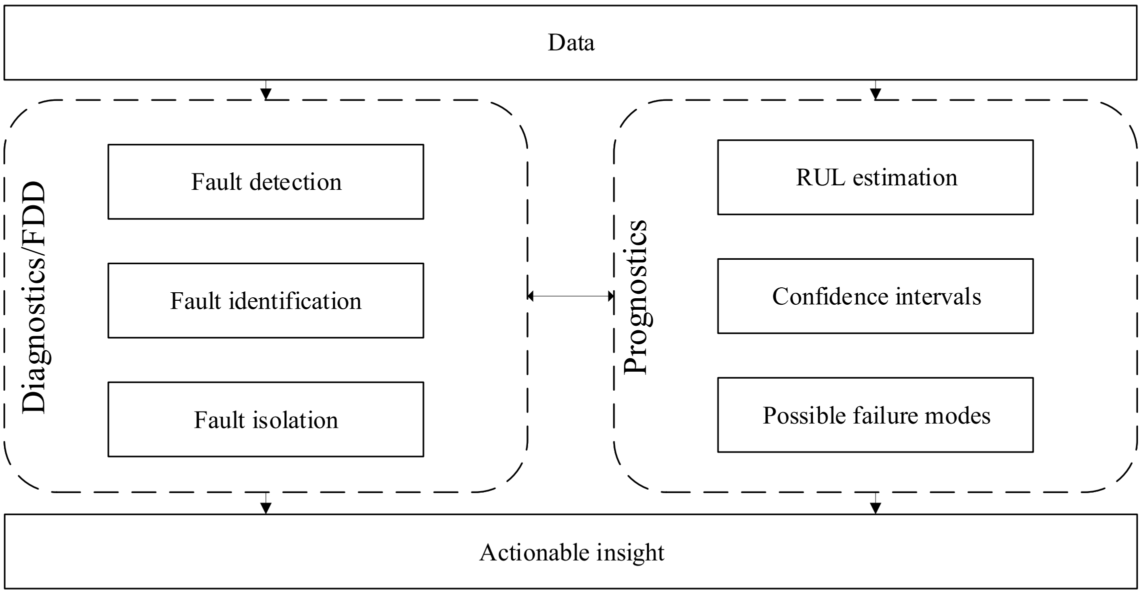


Figure 2.4: Typical steps involved in condition monitoring

- More production, less downtime, less labor costs to do maintenance, all of which result in cost savings.

A condition-based maintenance program relies on the quality or accuracy of condition monitoring. Consider a simple example of a component, which can have only normal and faulty condition, and a condition monitoring system, which can either indicate an alarm or not. In this case, four scenarios are possible:

- **True Negatives** (TN): The system is in normal condition and there was no alarm indicated by the CM system.
- **True Positives** (TP): The system is in a faulty condition and there was an alarm indicated by the CM system.
- **False Positives** (FP): The system is in normal condition and there was an alarm indicated by the CM system.
- **False Negatives** (FN): The system is in a faulty condition and there was no alarm indicated by the CM system.

The accuracy of the system is defined by the ratio of true positives and negatives compared to all the cases (Glantz, 1976).

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{TP} + \text{FP} + \text{FN}} \tag{2.1}$$

Another measure of accuracy is the F1-score (Van Rijsbergen, 2004), which is the harmonic mean of the precision ($\frac{\text{TP}}{\text{TP+FP}}$) and sensitivity ($\frac{\text{TN}}{\text{TN+FP}}$).

$$\text{F1} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{2.2}$$

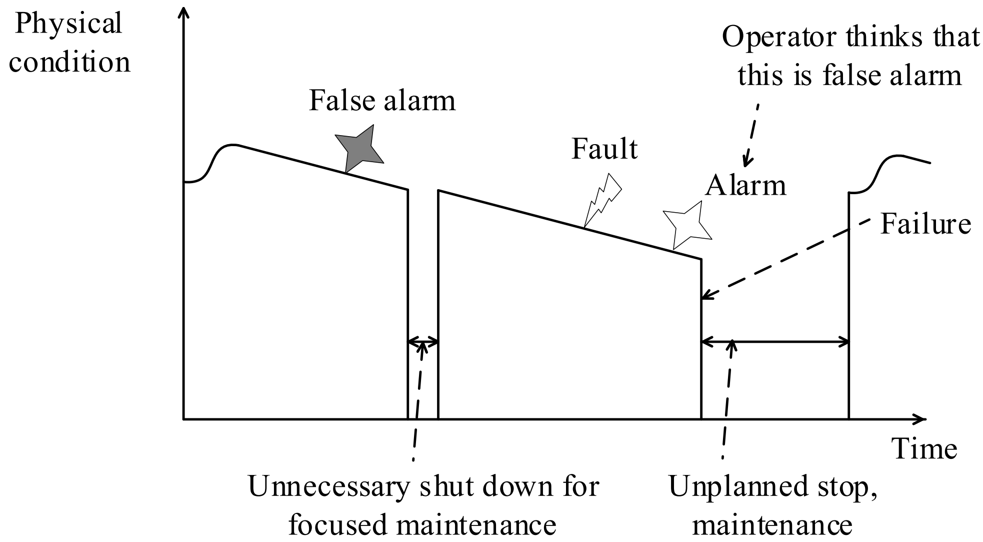


Figure 2.5: The effects of a false alarm

In condition monitoring systems False Positives are called *False alarms*. They occur when the monitoring system indicates a fault, while in reality there is no fault present in the monitored system. If a false alarm occurs is a decision might be taken to stop the process and perform a maintenance action during which it is discovered that there is actually no fault present in the system. The operator might lose trust in the CM system and the next alarm might be ignored, causing a real failure and unplanned shut down later on. A scenario with a false alarm is shown in Figure 2.5. If the CM system is very sensitive for noise the false alarm rate can become high, causing frequent stops for focused maintenance action, therefore there is a need to keep the false alarm rate low.

In condition monitoring systems False negatives are called *Missed alarms*. They occur when a fault is present in the monitored system and the condition monitoring system does not indicate any fault. Missed alarms might lead to failure, an unplanned shutdown and cause extra maintenance costs. A scenario with a missed alarm is shown in Figure 2.6. If a missed alarm leads to a failure, the condition monitoring system has failed and the advantages of condition-based maintenance are not exploited. Therefore there is a need to keep the missed alarm rate low. As equipment failures can have greater costs and longer shutdowns, usually the diagnostic systems are designed to be more tolerant of false alarms and less tolerant of missed alarms (Orkisz, 2017).

2.1.4. Economic feasibility of condition-based maintenance

There has been extensive research about whether CBM can be implemented in a profitable way. There are many case studies available in the literature when the use of CBM could achieve significant savings for its users. For example Rastegari and Bengtsson (2014) found that the application of CBM can gain significant paybacks in the manufacturing industry. Schön (2017) used the offshore oil and gas reliability data (OREDA, 2002) and found that CBM can reduce maintenance costs compared to the corrective maintenance strategy by about 70%. Sundin et al. (2007) have observed a number of cases, where savings

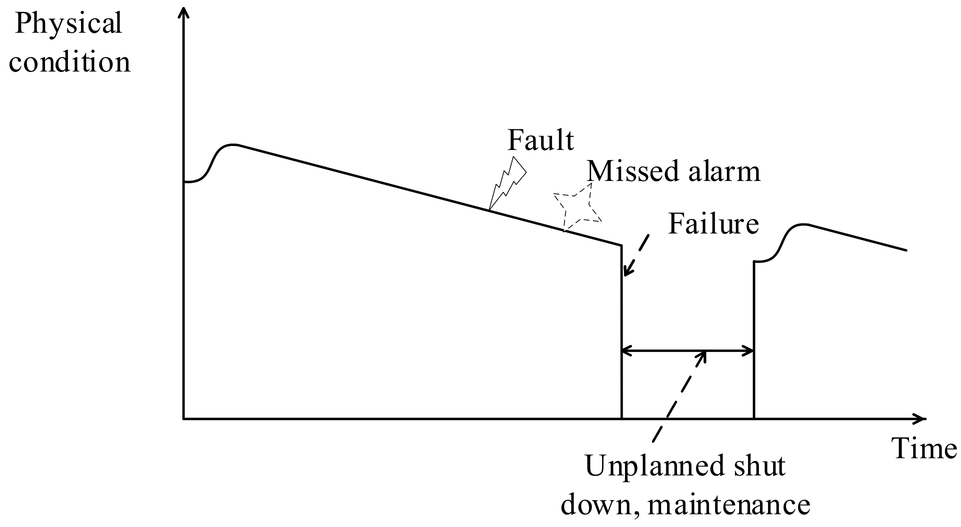


Figure 2.6: The effects of a missed alarm

were achieved by applying CBM in the pulp and paper industry. The costs and gains of CBM have been modelled by Al-Najjar and Alsyouf (2004) and it has been proved that CBM can become a profit center once false and missed alarms are reduced below a certain level. Therefore, new and accurate condition monitoring methods are needed to implement modern CBM systems which minimize false and missed alarms.

2.2. Data in condition monitoring

Recently there have been substantial improvements in sensing, connectivity and computing technologies. Industrial plants are now instrumented with a wide range of sensors, control, and data acquisition systems. With emerging AI, big-data and machine learning technologies and with the availability of a vast amount of data from all kinds of disparate sources, there are new opportunities in CM system development. Besides the traditional condition monitoring approaches focusing only on quantitative sensor data, other types of qualitative data sources can also be exploited and included in the condition monitoring framework for more accurate diagnostics and prognostics. There is a need for the data storage integration of computer maintenance management systems with condition monitoring systems, with alarm management systems and with advanced supervisory control and data acquisition (SCADA) systems (Galar et al., 2012).

The data used for condition monitoring can vary over a wide scale. In the literature two data types are described, the so-called sensor data and maintenance data (Jardine et al., 2006; Si et al., 2011; Heng et al., 2009). Sensor measurements can indicate either directly or indirectly a fault in the observed asset. Maintenance data contain historical data about the system, which describe the previous health states and maintenance actions (Jardine et al., 2006). Aside from sensor data and maintenance data, a few more types of data sources have to be mentioned for the sake of completeness: process condition data, alarm and event data, fleet data, design data, videos, and expert knowledge.

2.2.1. Sensor data

Sensor data are recorded through various sensor measurements and are the usual input data for diagnostics and prognostics algorithms. A wide array of sensors exist for condition monitoring purposes. The optimal choice of a sensor depends on the component, its properties, and its application. Here are some of the most typically used sensor types in CM applications:

- **Vibration measurements** are one of the most commonly used sources of data in CM. Vibrations are well suited for detecting faults in rotating machines, such as motors, compressors, and gearboxes (Randall, 2011). A rotating machine in normal condition has a certain vibration pattern, which changes due to degradation. There is an extended literature on vibration sensor types and fault detection techniques used for vibration-based condition monitoring, see for example Randall (2011), Carden and Fanning (2004) and Tandon and Choudhury (1999).

- **Acoustic measurements** can also be an indicator of deterioration and faults. Li (2002) gives an overview of how acoustics data can be used for tool wear monitoring. Acoustic signals provide measurements within a wide frequency range and do not require physical contact with the monitored asset. However, they may be prone to high background noise, attenuation, and reflections. Acoustic sensors for non-destructive condition monitoring can be found in refineries, power generation stations, aircraft, off-shore oil platforms, paper mills and bridges among others.

- **Electrical measurements**, such as current and voltage measurements, can also be indicative of developing faults in any systems which use electricity. For instance, they are frequently used for diagnosing induction motors by motor current signature analysis (Nandi et al., 2005; Benbouzid, 2000).

- **Lubrication analysis** using oil debris sensors provide information about the mechanical deterioration of the monitored system. Lubricants carry degradation information in the form of wear particles, where increased debris is an indicator of a deterioration (Tchakoua et al., 2013). Oil debris measurements are used for monitoring mechanical faults in rotating machines, such as bearings, induction motors, and gearboxes (Loutas et al., 2011).

- **Density measurements** play an important role in multi-phase flow monitoring, where the flow density may carry valuable information about the health state of the process and flow regime (Ruiz-Cárcel et al., 2015).

- **Temperature measurements** can also provide valuable insight into the health state of the equipment. Temperature sensors can detect fluctuations and unexpected rises in the surface temperature of the asset or in the liquid materials within the asset (Hellier and Shakinovsky, 2001). Thermocouples for high-temperature measurements and thermistors for low-temperature measurements are the most commonly used temperature sensors (Brun and Nored, 2006). Thermal imaging can also be applied to gather temperature data. The advantage of thermal imaging is that there are no sensors needed to be installed on the equipment. Bagavathiappan et al. (2013) gives an extensive overview of how thermal imaging can be applied to condition monitoring in various industrial applications. Janssens et al. (2015) shows that thermal imaging can be an efficient condition monitoring method for rotating machinery fault diagnosis.

- **Pressure measurements** play a significant role in process condition monitoring. Usually, pressure sensors are installed on process pipelines and if there is any blockage or leakage, the pressure drop or increase will be a good fault indicator (Jardine et al., 2006; Si et al., 2011).

- **Flow measurements** are also important in process condition monitoring. Flow sensors are installed on process pipes and they provide measurements about the flow rates, often together with flow density measurements (Isermann, 2011). Ruiz-Cárcel et al. (2015) established a case study on a multiphase flow facility with flow measurements, where these measurements were used to detect process faults, such as slugging.

- **Level measurements** can be found in process plants, where storage tanks are part of the production process. Level sensors monitor the liquid levels in storage tanks, which may also provide information about the operating condition and status of the process (Ricker, 1995).

2.2.2. Process condition data

The condition and lifetime of a component in a plant highly depends on its operating and environmental conditions. Using such data as inputs to the condition monitoring system can lead to better diagnostics and prognostics results (Mauricio et al., 2018). Process condition data is very useful for standardizing the sensor data and creating a baseline for the normal operation, especially in the case when the process is operated in multiple operating regimes.

- **Operating condition data** can contain information about the loading conditions, speed, and production rates depending on the asset. For example, for an induction motor, the operating condition may be defined by the load or the drive (Zarri et al., 2013; Martin-Diaz et al., 2018), while for a process pipeline the operating condition is given by the flow rate and composition (Ruiz-Cárcel et al., 2015). Standard monitoring approaches require measurements at similar operating conditions (ISO 17359:2018). However in industrial applications often there is not enough historical data from all of the possible operating conditions of the system, therefore operating condition data can help to build more accurate diagnostics and prognostics models (Zhao and Huang, 2018).

- **Control data** can be valuable when there are automatic control loops in the process. The control loops of the system might be able to compensate the effects of an unwanted fault for a while. However, faults might cause control quality degradation (Czajkowski and Patan, 2016). In compensating the effects of the fault, the controller response and feedback may contain information about the fault and its dynamics. Hence, an advantage of these signals that they might be used for condition monitoring purposes without requiring additional sensors (Orkisz et al., 2009). Furthermore, control loops can become untuned, leading to oscillations and steady state errors which can become apparent in other signals recorded from the plant (Thornhill et al., 1999).

- **Environmental condition data** may also play an important role in the CM framework, as environmental conditions can influence the operation of the assets and speed up its degradation under harsh conditions. Sensors might also have faulty behaviors under certain environmental conditions. Jardine et al. (2006) list moisture data, humidity data and weather data as condition monitoring data. The list can be extended with acoustic, magnetic and vibration effects originating from external sources to the process. These environmental conditions not only influence the behavior of the system, but can also bias the sensor measurements.

2.2.3. Alarm, event, and change data

Alarm management is usually integrated within the SCADA system, it is closely connected with process safety and it has its established practices and standards (IEC 62682:2014). Event and change data is often collected from SCADA systems in process condition monitoring.

- **Alarm data** are typically recorded when a pre-set safety threshold of a monitored variable has been crossed causing a threshold-based alarm. In the case of a failure, many alarms can be triggered in a short period of time causing an alarm flood. Alarm data can be effectively used for fault classification, as different faults have different alarm flood patterns (Lucke et al., 2018).

- **Event data** contain records of events, which happened to the monitored system during its operation. For example, a sensor lost its connection, or an alarm was triggered which caused a partial automatic shut-down in the plant.

- **Change data** correspond to a special type of event, which was triggered by a change in the process made by the operator. SCADA systems, such as ABB Ability Symphony + (ABB, 2019b) and ABB Ability™ System 800xA (ABB, 2019a), often keep an automatic log of all changes containing any new process inputs, operating conditions or set-points.

2.2.4. Maintenance data

Maintenance data describe the previous health states and maintenance events from the lifetime of the asset. It can include the following types (Jardine et al. (2006)):

- **Installation data** include information about the conditions, time and actions taken during installation and commissioning.

- **Logs of previous failures** contain records about any previous breakdowns or failure that occurred during the lifetime of the asset. If there was a failure the log may contain contain information on the part of the asset that failed, its failure mode and the impact it might have on other parts of the system.

- **Maintenance logs**, which are a comprehensive collection of past maintenance actions, can contribute to both diagnostics and prognostics. They can contain logs from periodic maintenance actions and detailed descriptions with the condition and faults of the asset during maintenance and the parts which were necessary to be replaced or upgraded. It may also contain a collection of past failures and the actions taken to correct them. It can include previous overhauls and previous shutdowns as well. These logs can be logged automatically or logged manually by maintenance personnel in a computer database or they can be handwritten by maintenance personnel. In the case of handwritten logs, extra effort is needed to integrate the data in the condition monitoring systems.

2.2.5. Fleet data

Fleet data contain information which is valid for a series or set of assets. They are not always available, either because the asset is a relatively new product with not much condition monitoring history recorded yet or there is only a single unique asset under observation. However, if fleet data are available, they can significantly help the prognostics and diagnostics algorithm.

- **Unit-to unit variability** refers to information on how assets from the same manufacturer with the same parameters may degrade differently even from the same batch due to their diversity in their working environment (Zhang et al., 2015). If the manufacturer has historical knowledge of this, unit-to-unit variability can contribute to the condition monitoring system in the form of confidence intervals associated with the remaining useful lifetimes.

- **Failure modes** of a component can help fault detection and diagnostics. If historical data are available from which the probability distributions of the different failure modes can be inferred or data about the most common failure modes on the fleet level, they can improve the accuracy of the prognostics and diagnostics algorithms by including this prior knowledge in the condition assessment.

- **Reliability data** describes the ability of an asset to function as specified for a given period of time (Geraci et al., 1991). There are several key performance indicators (KPI) , which are meant to describe asset reliability. Mean-Time-To-Failure (MTTF) indicates the time interval from new operation till failure. Another reliability indicator is the Mean-Time-To-Repair (MTTR) , which specifies the necessary time interval for maintenance actions, once a failure happened. Lastly, Mean-Time-Between-Failure (MTBF) is the sum of MTTF and MTTR measuring how often a system is going to fail (Frangopol et al., 2001). There might also be associated uncertainty intervals for the MTTF, MTTR and MTBF indicators available (OREDA, 2002).

2.2.6. Design data

Design data refer to the documentation and specifications of the observed object. They can help to identify potential failure modes based on the documentation. The tolerance or healthy operational limits of the object can potentially help in defining certain preliminary alarm and warning thresholds in the condition monitoring system.

- **System design** contains the detailed schematic of the process and its parts, the list of sensors and data acquisition systems installed and the description of the control system. It is the starting point of the design of the condition monitoring system.

- **Connectivity** information refers to data on how the components of the system are connected to one another. These components might interact with each other, their faults can propagate in the system from one sub-system to another (Jaramillo et al., 2017; Ruiz-Cárcel et al., 2016). The knowledge about connectivity can be an additional input to the design of the condition monitoring system.

2.2.7. Videos

Cameras are often installed in plants for surveillance, security and monitoring purposes. If installed near critical equipment, videos can become a new source of condition monitoring data to be considered. Image processing once implemented in real time can also be applied to videos. A recent study successfully implemented real-time image processing of videos recorded from a pantograph catenary system for railway monitoring purposes (Karakose et al., 2017).

2.2.8. Expert knowledge

Expert knowledge is one of the most valuable and qualitative sources of information. It can contain knowledge about the system design, topology, connectivity, typical fault modes and typical fault signatures in the measurements. They are usually hard to quantify and integrate into the CM system. However, they can hugely help the design of the CM systems by specifying for example the failure modes, monitoring thresholds or expected operating conditions.

2.3. Design of a condition monitoring framework

A typical condition monitoring framework is composed of three main parts: *Data acquisition*, *Data-processing* and *Maintenance decision support* (Jardine et al., 2006). Data-processing is composed of *Pre-processing* and *Feature design*. *Maintenance decision support* consists of *Diagnostics* and *Prognostics*, as described in Section 2.1.3, resulting in an actionable insight about the current and future health state of the monitored system. Figure 2.7 gives a schematic depicting the design steps of a condition monitoring framework. All of the data described in Section 2.2 are useful inputs when designing the monitoring framework. The data needed for accurate monitoring of the asset determines the data acquisition set-up. Data acquisition is comprised of sensor selection and placement, determination of sampling frequencies and selection of the data acquisition policy. Once the data acquisition system is installed, the gathered data are pre-processed before they are ready for further analysis. Data preparation, cleaning, and verification are the typical pre-processing steps, which may be further facilitated with data exploration and visualization. During the design of the CM system, the necessary pre-processing methods have to be determined based on the data types, historical data, and system behavior. Feature design determines a set of features which are the most representative for a monitoring problem. It may involve feature extraction, multivariate data reduction, and feature selection. The diagnostics level determines the current health state of the monitored asset. If there is enough historical data and expert knowledge available the prognostics level can be designed to give a prognosis about the future health state of the monitored asset. It has to be pointed out that designing a condition monitoring framework may be an iterative process. For example, the feature design and the design of maintenance decision support levels may influence the data acquisition and pre-processing levels. In this section, the previously listed design steps are described with the most commonly used methods and considerations.

2.3.1. Data acquisition

Condition monitoring systems work on the basis of data recorded from the monitored asset, therefore a data acquisition system is required. In order to achieve sufficient data quality, a few considerations

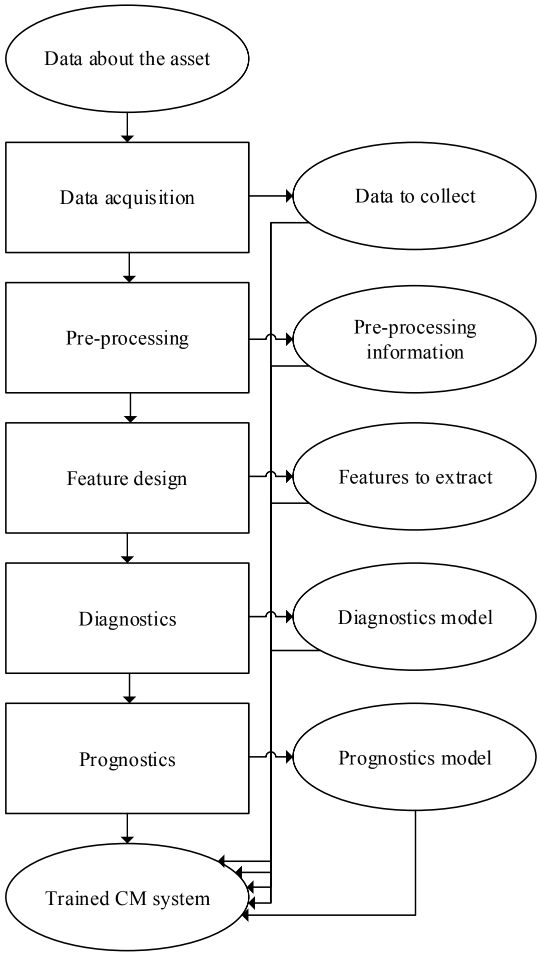


Figure 2.7: Design of a condition monitoring framework

have to be taken into account when designing both the hardware and the software of the data acquisition systems.

Sensor selection and placement

Sensor selection and placement are key for successful diagnostics. Sensors are usually selected and placed based on expert knowledge about the monitored assets. For example for rotating machinery the associated sensor mounting locations and guidelines are summarized in two ISO standards (ISO 10816-3:2009 and ISO 10816-1:1995). Similarly, there is literature available summarising optimal sensor placement strategies for process monitoring applications such as multiphase flow monitoring (Kawaguchi et al. (2013)) or gasification power plant monitoring (Lee and Diwekar, 2012).

Sampling frequency

It is very important to acquire data at the correct sampling frequency in order to ensure that all relevant information available from a sensor is retained for analysis. The Nyquist-Shannon sampling theorem provides guidelines for choosing the minimal sampling frequency. It states that if the maximum frequency of interest in the measured analog signal is f, then it has to be sampled with at least $2f$ times per second for all the information to be retained (Shannon, 1948). In order to determine the sampling frequency for a typical application, the maximum frequency of interest has to be estimated. Faults often have characteristic signatures that may be analyzed in the frequency domain. In some cases these signatures are somewhat deterministic (e.g. harmonics of rotating speed or supply frequency) in which case it is sufficient to identify which signatures to include in the analysis. Sometimes the frequencies are not deterministic (e.g. resonances or fluid flow oscillations) in which case best practice, advanced simulations or dedicated experiments might be used to ascertain the required frequency bands.

Data acquisition policy

Data can be gathered actively when an action is taken by the operator to record additional data, and passively when the data acquisition system is set-up for automatic data collection.

- **Passive, continuous**: The data acquisition system is set-up in such a way that it continuously gathers data during the operation of the monitored asset. This configuration enables the implementation of an online condition monitoring systems, safety measures and automatic control systems based on sensor measurements and other sources of data. The storage place for recording such historical data is often limited, therefore, sampling frequencies for continuous passive measurements are typically low.

- **Passive, periodic**: If there is no possibility to record data continuously at higher sampling rates, data acquisitions systems can be designed in such a way that periodic measurements are taken from certain sensors during the operation of the asset or systems. The data may be acquired from the sensors at a relatively high sampling frequency but with a relatively short signal length. Between data acquisitions there may be a longer period of inactivity where no data is acquired from the sensors.

- **Passive, triggered**: Often the data acquisition systems have built-in data recording options that are triggered sporadically on the basis of an event in the system. Such triggers can be for instance a crossed safety threshold, an alarm or a failure of a component or sensor.

- **Active, on-demand**: On-demand data acquisition happens when maintenance personnel requires extra data about a certain part of the process, which might be faulty, or about a certain (new) operating condition such as start-up, shut-down, or high loads. Extra sensors can be installed temporarily by maintenance personnel for on-demand monitoring. Once the system has suffered a failure, on-demand data acquisition might be needed before the replacement or repair of the failed component.

- **Active, periodic**: Passive data acquisition systems can be complemented with periodic maintenance check-ups on certain parts which might happen when the system is out of operation. The additional active measurements and the maintenance actions can also serve as a source of data for the data acquisition system.

2.3.2. Pre-processing

Once data is gathered by the data acquisition system, it has to be pre-processed so that meaningful insight can be extracted. The steps of pre-processing might vary depending on the data type, the format of the data, the sampling frequency and the length of the recorded data (Xu et al. (2015)), therefore,

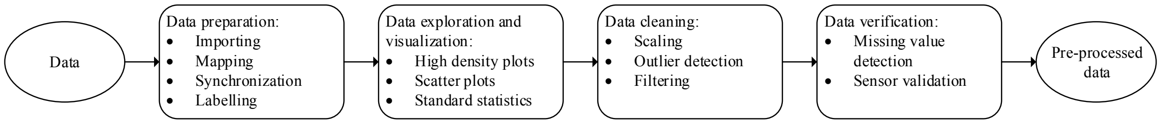


Figure 2.8: Steps of pre-processing

here only a general overview is given about the most common pre-processing steps without an attempt of being exhaustive. The selection of the necessary pre-processing methods is done after initial data analysis. The steps of preprocessing are shown in Figure 2.8.

Data preparation

- **Import and map**: The first data preparation step is import and map. The dataset is imported to a processing environment and the tag names are aligned with the variable names.

- **Synchronization**: If data were collected from different data acquisition systems, synchronization or time alignment is needed to align the time stamps of the measurements from the different systems. This synchronization can be performed manually or with the help of time alignment and trajectory synchronization methods. Some of the most well-known and representative techniques for synchronization are truncation and padding, linear time scaling, dynamic time warping and correlation optimized warping (Xu et al., 2015).

- **Labelling**: Data preparation also consists of labelling when the observations are matched with health states, faults, fault severities, and process conditions. Labelling is applicable to historical data when there is knowledge about the condition of the system. Such knowledge can be available either from maintenance records or from the fault detection and diagnosis outputs of the condition monitoring system.

Data exploration and visualization

It is a good practice to perform data exploration and visualization after data preparation. Some of the most simple and popular methods are calculating the standard statistics (root mean square (RMS), variance, standard deviation, higher order statistics), visualizing time domain plots, scatter plots, bar plots and high-density plots. The observations made during the exploration can also help to develop hypotheses that might subsequently guide the design of data pre-processing.

Data cleaning

- **Scaling**: Raw sensor measurements may require scaling to obtain the desired units by adding an offset and multiplying the raw measurement with a scaling factor.

- **Outliers**: Unusual observations that are not consistent with the rest of the observations from a statistical point of view are called outliers (Barnett and Lewis, 1974). Their detection and removal or substitution can improve the monitoring performance. Outlier detection methods span over a wide range in terms of complexity from simple univariate methods like the 3σ rule, which can be applied to normally distributed univariate data, to more complex multivariate methods like the minimum covariance determinant estimator (Hodge and Austin, 2004).

- **Filtering**: Signals may contain noise, which is an undesirable random or periodic component in a signal. The signal-to-noise ratio (SNR) can be a good indicator to inspect and compare the level of the measured signal to the level of background noise. In the case of small SNR, filtering methods can be applied to remove unwanted noise from the signal. Simple filtering methods include filters like the widely used moving average filter. Model-based methods include the Kalman filter and its variations, while data-driven methods include filters, such as digital filters, wavelet filters, and the Savitzky-Golay filters (Xu et al., 2015).

Data verification

- **Missing value detection**: Sensor or communication failures can cause data inconsistency in raw measurements. Data can contain missing values or false sensor readings. To avoid disruption in the condition monitoring performance, data verification is necessary to check the consistency of the data. There are several methods described in the literature to treat missing data (Little and Rubin, 2014). The easiest and simplest method is to apply variable deletion or time stamp deletion to eliminate the missing values. If only a fraction of the data is missing there are more sophisticated methods which are able to estimate and replace the missing values with regression replacement, mean replacement, interpolation replacement, maximum likelihood estimation and with other machine learning methods (Xu et al., 2015).

- **Sensor validation**: To indicate false sensor readings, sensor validation methods are necessary. Sensor faults can include soft faults and hard faults. In case of soft faults, such as bias, drift, gain, and precision degradation, once the actual type of the fault is recognized and the parameters causing the faults are known, it is still possible to calculate the correct measurement values with the appropriate scaling and calibration. In case of hard faults, when the sensor completely fails, measuring only constant value or noise, it is not possible anymore to recover the actual measurement values (Yi et al., 2017; Kullaa, 2013).

2.3.3. Feature design

An ideal dataset contains representative, non-redundant variables which are easy to interpret by experts. However, large datasets are rarely ideal, often containing correlated and redundant variables. Hence, feature extraction is often necessary to find those properties which best represent the system for a certain monitoring problem. A feature is defined as an individual measurable property of a monitored system or asset (Chandrashekar and Sahin, 2014). Feature extraction methods vary by data type, sensor type and by monitoring purpose, requiring domain knowledge about the sensors, the system and its components, and their possible fault modes (Jardine et al., 2006). If domain knowledge is not available, a standard feature set can be applied and further refined with multivariate methods and feature selection methods reducing its dimension, eliminating correlation and redundancy. The possible steps of feature design are shown in Figure 2.9.

Feature extraction

There are a range of signal processing methods, such as signal averaging and time domain, frequency domain or time-frequency domain analysis, which can be applied for feature extraction from the raw sensor measurements.

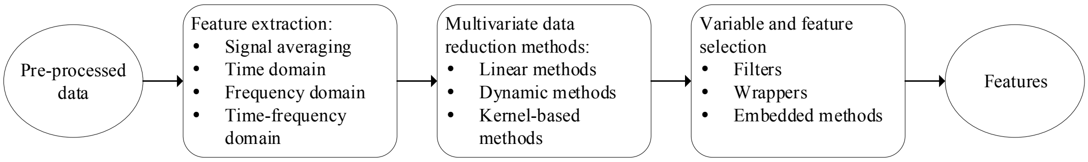


Figure 2.9: Steps of feature design

- **Time domain analysis**: Some examples of the most common time domain features are standard statistical features like root mean square, variance, standard deviation, skewness, kurtosis and the maximum peak to peak value. Besides standard statistical features, signal averaging is also a useful time domain analysis tool.

- **Signal averaging**: The motivation for signal averaging is the removal of unwanted noise and also the extraction of periodic signals from the original data. The most common signal averaging methods in condition monitoring are moving average, exponential averaging and time synchronous averaging (TSA) , which are described in detail by Braun (2008). They are widely applied in rotating machinery diagnostics, especially TSA, which is able to separate the signature of the different rotating components (McFadden, 1987; Bechhoefer and Kingsley, 2009).

- **Frequency domain analysis**: A time domain signal can be expressed by a summation of sinusoids, each with a particular amplitude and phase. This is called the frequency-domain representation or the spectrum of the signal. Frequency domain analysis involves the transformation of the signal into the frequency domain and the extraction of features from the spectrum at certain frequencies or from certain frequency bands. The most common method for transforming a signal into the frequency domain is the fast Fourier transform (FFT) . In rotating machinery diagnostics it is a very common tool, as certain faults have clearly identifiable patterns in the frequency domain with distinguishable peaks at certain frequencies (Goldman, 1999). These frequencies of interest can be easily calculated based on the physical parameters of the system, such as the rotational frequency, pole pairs, and supply frequency, which are examples of typical parameters of induction motor monitoring (Nandi et al., 2005). Another frequency domain method is Power Spectral Density (PSD) analysis, which is also commonly used in monitoring applications where the signal power distribution carries the useful information for fault diagnostics instead of looking for peaks at the certain frequencies. Such applications are common in process condition monitoring applications, where different fluid flows have different PSD features (Santoso, 2012; Abbagoni and Yeung, 2016).

- **Time-frequency domain analysis**: Although frequency domain methods are a good choice for analyzing stationary signals, for non-stationary signal analysis time-frequency methods should be considered which investigate the signals both in the time and frequency domain. The most popular time-frequency transformations are based on the short-time Fourier transform (STFT) (Benbouzid, 2000) and the Wigner distribution function (Staszewski et al., 1997). They are able to map the one-dimensional time-domain signal to a two-dimensional function of time and frequency. Another more advanced time-frequency method family are based on the wavelet theory. The wavelet transform can be used for multi-scale signal analysis through dilation and translation of the signal for effectively extracting time-frequency features. Both continuous and discrete wavelet transforms have been successfully applied in literature for diagnostics, a review of their applications is summarised by Peng and Chu (2004).

Multivariate data reduction methods

Data might contain spatially correlated measurements when a large number of sensors are installed to monitor the same system. However, the variability in data is often lower dimensional than the number of original variables. Multivariate approaches aim to handle dependent, correlated and redundant measurements by transforming the time-series measurements to a less redundant feature space which is still representative of system behavior (Chiang et al., 2000). Principal component analysis (PCA) (Kresta et al., 1991) transforms the features to a lower dimensional uncorrelated linear feature space, which may be used to reduce the dimensionality of the dataset by keeping only the first few principal components as features. It has been used for feature extraction, process monitoring and FDD purposes (Yin et al., 2012). Partial least squares (PLS) (Wise and Gallagher, 1996) is a multivariate method which is closely related to PCA. PLS finds a linear regression model by projecting variables to a new linear space (Joe Qin, 2003). It has been used for feature extraction, model building and in FDD applications. Independent component analysis (ICA) (Lee et al., 2004b) transforms the features to a statistically independent and linear feature space. It has been applied in process monitoring and FDD applications with non-Gaussian data (Yin et al., 2012). Canonical Variate Analysis (CVA) (Odiowei and Cao, 2010) finds the maximum correlation between any two multivariate datasets. CVA is also able to account for temporal correlations, which makes it suitable for monitoring applications where the representation of the system dynamics is of importance (Ruiz-Cárcel et al., 2016). Dynamic and kernel-based methods provide non-linear alternatives to handle the complexities and temporal correlation for the above-listed methods (Lee et al., 2004a, 2007).

Feature selection

Even when domain knowledge is available, feature extraction might yield correlated, redundant and sometimes irrelevant feature sets. Feature selection methods can help in selecting the relevant and informative features, reducing the feature set to a smaller subset, where the features are less correlated and less redundant (Guyon and Elisseeff, 2003). The three main types of feature selection methods are filters, wrappers, and embedded methods. Filters select features without optimizing the performance of a predictor by ranking the features using a relevance index. Wrappers iterate through subsets of features with a learning algorithm to find the best subset of features using predictive performance as scores. Embedded methods perform feature selection during the training and validation of a learning algorithm, therefore, they are similar to wrappers (Chandrashekar and Sahin, 2014). Guyon and Elisseeff (2006) summarizes the most common methods used in the literature from several application domains.

2.3.4. Diagnostics

Many fault detection and diagnosis methods have been described and implemented in several engineering applications, as they are the first key element towards successful maintenance decision making in a condition-based maintenance framework. The most common diagnostics methods are shown in Figure 2.10 grouped according to their system models and according to how they conduct the diagnostics task, which is further described in the following section.

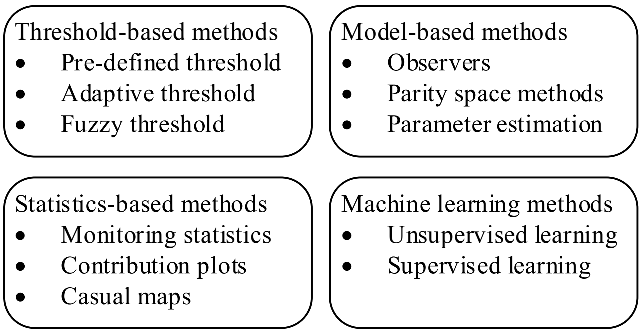


Figure 2.10: Examples of diagnostics methods

Threshold-based methods

The most traditional fault detection methods are the threshold-based methods, which are used for monitoring whether a variable or feature exceeds a threshold. The thresholds can be based on pre-defined values which can be set in an adaptive way which automatically adjust to the operating conditions (Höfling and Isermann, 1996; Isermann, 2006) or they can be based on fuzzy thresholds when the faulty and normal conditions are not clearly defined (Isermann, 1998). Threshold-based methods are simple, easy-to-use and are transparent, directly providing the indicator which caused the detection. However, in complex systems, they are less practical. If several indicators have exceeded their thresholds at the same time causing a so-called alarm flood, the root cause of the problem becomes less evident (Isermann, 2006).

Statistics-based methods

Statistics-based methods use time-series features extracted by multivariate approaches for fault detection purposes. They aim to create a single fault indicator which is typically based on the Hotelling's T^2 statistic or the squared prediction error (Joe Qin, 2003). Statistics-based methods are similar to the traditional threshold-based approaches, as a threshold called the control limit is set-up and monitored. A fault is detected when the fault indicator crosses its control limit. Instead of having thresholds for many process variables, only one health indicator needs to be monitored to determine if there is a fault in the system. Hence, statistics-based methods are popular for fault detection in process condition monitoring.

Contribution plots are fault diagnostics tools among statistics-based methods (Chiang et al., 2000). They can help to pinpoint the variables, which contribute the most to a faulty condition, based on which the fault type and the root cause can be identified (Miller et al., 1998). The root cause of the fault and its propagation path in the plant can be further traced by building the causal map of the variables in the system (Chiang and Braatz, 2003; Bauer and Thornhill, 2008).

Model-based methods

Model-based methods predict the expected system behavior based on a model, which is then compared with the real system behavior. The difference between the two, the so-called residual, is considered as a fault indicator. Model-based methods can be very efficient FDD tools if there is enough information available about the physics of the system to build an explicit mathematical model from first principles or from system estimation. However, as the system complexity increases, estimation and modeling becomes more difficult. There are several ways to predict system behavior and residuals. Observer-based techniques estimate the internal state of the system from the measured inputs and outputs, and are robust against uncertainties and disturbances. Parity space methods provide a model, which only depends on inputs and the outputs and is generated by using system redundancies. Parameter estimation predicts non-measurable parameter values and internal state variables based on the measurable inputs and outputs (Tidriri et al., 2016; Baranowski et al., 2017).

Machine learning methods

Machine learning methods used in diagnostics applications make assumptions about the current health state of the system on the basis of historical data without building an exact model about the physics of the system. If prior knowledge is available about the structure and physics of the system, the parametrization of the methods can be adjusted accordingly. Machine learning methods have two main categories: unsupervised and supervised methods.

Often there is no prior knowledge available about the health state of the system. In this case, it is extremely difficult to reliably label historical data according to health states and process conditions. Unsupervised methods can offer a solution to such problems where the objective is to construct decision boundaries for newly recorded observations based on unlabelled historical data. Unsupervised methods mostly rely on clustering algorithms, which are able to group the historical data corresponding to health states and operating conditions. After a sufficient training period, the unsupervised methods are able to identify a potential failure condition when a new cluster is formed and persists for some time (Schoen et al., 1995).

If historical data contain information about health state and process conditions the data or feature set can be easily labeled. In this case, it is possible to set-up a baseline of conditions and use supervised methods to detect and compare the newly recorded observations to the labeled historical data. Some of the most popular supervised approaches for fault diagnostics are Bayesian methods (Da Silva et al., 2008; Jaramillo et al., 2017; Stief et al., 2019c), Support Vector Machine (SVM) classifiers (Widodo and Yang, 2007), Artificial Neural Network-based classifiers (ANN) (Li et al., 2000) and k-nearest neighbours classifiers (Liu et al., 2018b; Jardine et al., 2006).

2.3.5. Prognostics

Prognostics aims to predict the future degradation of the asset based on the current state and historical degradation path, therefore the success of prognostics relies on the accuracy of diagnostics. The literature on prognostics is still much smaller than the literature available for diagnostics.. In this section a review of the most typical prognostics approaches is provided based on surveys, such as Heng et al. (2009); Peng et al. (2010); Kan et al. (2015) and Sutharssan et al. (2015). The two main types of prognostics methods fall into Physics-based methods and Data-driven methods. Hybrid methods also exist, which combine the strength of the two types, however, they are not discussed here.

Model-based methods

- **Physics-based methods** focus on building mathematical models, which are able to describe the most common physical causes of degradation, such as crack propagation, spall growth, stiffness-based damage, and fatigue. The inputs of the prognostics algorithm, such as system-specific mechanical knowledge, defect growth equations and sensor data are combined in order to obtain accurate prognostics results. These methods are hard to build and are very application, failure mode and degradation specific. Examples of physics-based methods are commonly found in rotating machinery applications (Li and Lee, 2005; Oppenheimer and Loparo, 2002).

- **Knowledge-based models** are built on a expert knowledge using qualitative data to describe the physics of the system. Examples of knowledge-based model include include rule-based expert systems (Biagetti and Sciubba, 2004), finite-state machines (Kurien and Nayak, 2000) and qualitative reasoning (Weld and De Kleer, 2013).

Data-driven methods

Data-driven methods are less application specific. There is no need for accurate physical and degradation models, rather a vast amount of historical data is needed where the degradation of the asset was monitored. Such data can either be collected from real systems or from targeted experiments with accelerated-life time tests (Jung et al., 2006; Tsui et al., 2015; Deutsch and He, 2018).

- **Statistical methods** are used to build prognostics models based on historical sensor data. They aim to estimate the degradation, damage initiation, and progression based on previous examples of the degradation of similar assets. The monitored asset is then compared to the degradation model of a similar asset using statistical degradation indicators (Sikorska et al., 2011). The most common statistical methods are trend extrapolation (Engel et al., 2000), Auto-regressive Moving Average (Yan et al., 2004) and Proportional Hazards Modelling (Liao et al., 2006).

- **Machine learning methods** can build purely data-driven models for a wide range of prognostics tasks. Artificial Neural Networks are used for time series prediction, exponential projection and data interpolation with longer prediction horizons (Heng et al., 2009). Bayesian methods are used to calculate the failure probabilities in along the prediction horizon (Zhang et al., 2007). Hidden Markov Models are used for adaptive stochastic fault prediction (Zhang et al., 2005).

2.3.6. Actionable insight as a design consideration

Condition monitoring is typically used to support decisions on maintenance actions. The form and presentation of insights about the health state of the monitored system may depend on the end users and their roles and functions in the maintenance of the system. The end users of the condition monitoring system can be service personnel, operators, maintenance managers, plant managers among others. They are interested in the details of the health state to different extents. The choice of diagnostics and prognostics algorithms may depend on what kind of information the end user would like to receive. Do they require only a simple yes/no indicator if there is a fault? Are they interested in the fault type? Are they interested in the fault severity? Do they require the transparency of the CM system for easy root cause diagnosis? Do they need to know the remaining useful lifetime of the asset? The following list provides an overview of the possible end results:

- **Normal/faulty indicator**: The end user is only interested to know if the operation of the monitored asset is normal or faulty. This is the simplest output, which may be practical for component-level monitoring where only a few faults are present and only a few signals are used in the monitoring system. To determine the actual root cause of the fault further manual inspection might be necessary.

- **Traffic light**: The end user receives information if the monitored asset is healthy, has started to degrade or has a severe degradation (ABB (2018)). Traffic lights give a convenient way to interpret degradation, however, the actual health state and the root cause of degradation may require further investigation.

- **Fault type**: The typical output of the diagnostic layer. If only the fault type is given by the CM system, a further manual inspection may be necessary to identify the root cause.

- **Fault severity**: A complementary end result beside the fault diagnostics result describing the level of degradation within a fault type.

- **Probabilities of health states**: The current health state is not only given with the most probable faults, but all the pre-defined fault conditions are associated with probabilities. This way the end-user receives additional information about the confidence of the diagnostic result.

- **Remaining useful lifetime**: A measure of degradation to the end user before they schedule maintenance actions.

- **Root cause**: A complementary end result beside the fault diagnostics result. If it is automatically provided by the CM system, maintenance personnel can directly target the degraded part and execute focused maintenance.

2.4. Operation of a condition monitoring system

Figure 2.11 gives an overview of the typical structure of the operation of a condition monitoring framework. Data from the monitored asset and the trained condition monitoring framework is the input of the whole framework, while actionable insight about the current and future health state is the output, which is given to the end-user.

The pre-processing layer takes the pre-processing information from the trained framework. Data preparation during operation is only the map, import, and synchronization of the incoming data. Labelling is conducted once the current health state is determined by the diagnostics layer. If an online data visualization interface or HMI is available in the condition monitoring framework, the operators might observe collected data about the monitored asset during operation. If during the training scaling was necessary, the data might be scaled with a pre-defined scaling factor. If outliers and noise were present during the training, outlier removal and filtering might also take place during operation. Pre-trained missing value detection and sensor validation methods can contribute to the more accurate operation of the condition monitoring system. If the processing time of one observation is limited, these steps might be avoided to reduce the computation time.

The pre-processed data is passed to the feature extraction layer. During operation, feature design is limited to feature extraction only, as the features have been already defined and selected during the training of the method. The feature extraction layer calculates the necessary transformations according

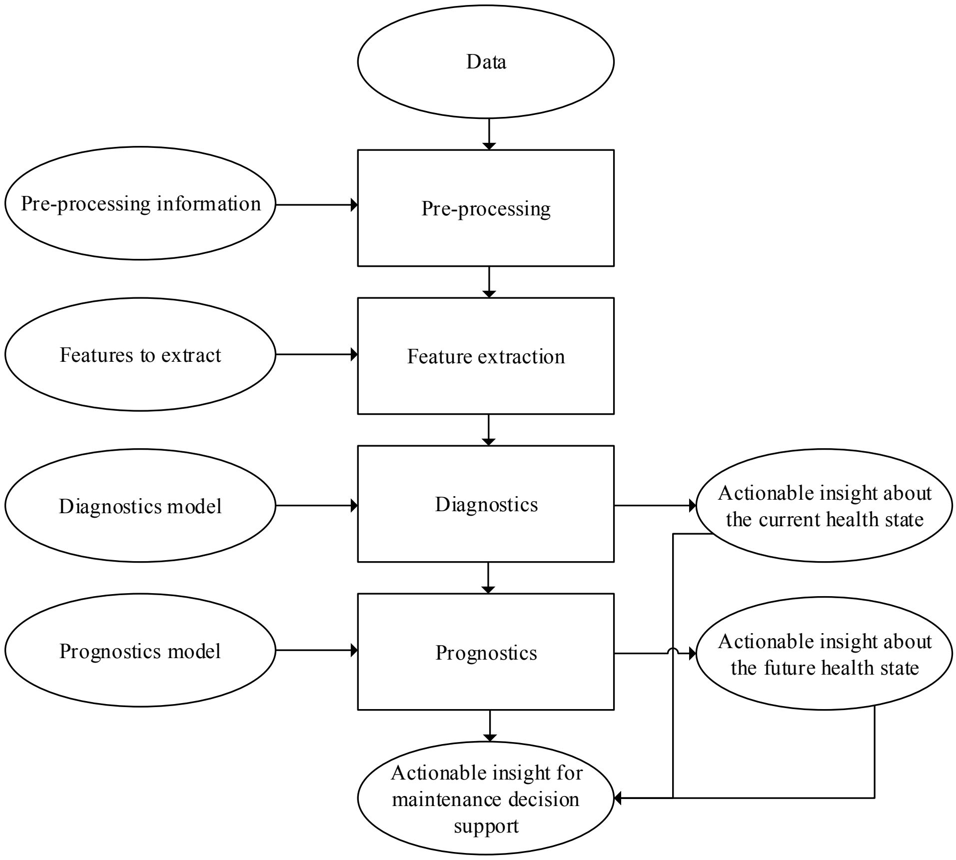


Figure 2.11: Operation of a condition monitoring system

to the pre-defined feature set. The features are fed to the diagnostics layer, which determines the current health state using a pre-trained diagnostics model. The prognostics layer determines the future health state based on the current health state and a pre-trained prognostics model. Finally, the current and future health state of the monitored asset is provided to the end-user.

2.5. Summary of condition-based maintenance

In this chapter, the terms and concepts of condition-based maintenance were introduced together with motivations, strategies, and tasks of condition monitoring. Several benefits of condition monitoring were described and it was also highlighted these benefits may only occur if the CM system is able to achieve low false and missed alarm rates. Therefore, accurate condition monitoring methods, which are able to give valuable insight into the maintenance decision-support, are required.

Monitoring complex and well-instrumented industrial plants with CBM systems can have a number of advantages in terms of safety and cost. They are monitored with a wide array of sensors, with several different monitoring systems generating disparate data. However, to achieve the economic feasibility of the CBM system, the following considerations have to be taken into account:

- The false and missed alarm rates should be low to provide optimal plant efficiency, safe operation and reduced downtime.

- It is necessary to consider data-fusion methods to effectively incorporate all the information in the monitoring systems. These methods is further explored in the upcoming Chapter 3.
- The algorithms have to use design and topology information to best suit the topology of complex industrial plants.
- Ideal condition monitoring systems are modular and scalable in order to easily adapt to new or changing systems. If a new sensor is added or an old sensor is removed the whole system should not need to be retrained.
- Transparency is a required characteristic of a condition monitoring systems. If the end user questions the diagnostics or prognostics result, it should be possible to trace back the result to the root cause.
- The data acquisition system has to be optimally set-up in terms of redundancy, sensor placement, and robustness so that the data quality is sufficient for a given monitoring problem and measurement noise does not result in reduced monitoring performance.
- The monitoring algorithms have to adapt to changing operating and environmental conditions.

The design and operation of condition monitoring systems have to accommodate heterogeneous data, not only on the pre-processing and feature design stage but also on the diagnostic level. Hence, novel data fusion methods are needed to tackle the challenges and exploit the opportunities in condition monitoring.

3. Data fusion

In this section, an overview of the fundamental terms of data fusion is given. Data fusion types, abstraction levels, methods and challenges of data fusion are reviewed with special attention as to how data fusion is applicable in the field of condition monitoring. The literature on data fusion is extremely wide as the potential application of data fusion is vast. It is used for example in the fields of medicine, biology, neuroscience, information technology, military applications involving target recognition, economics, robotics and also in condition monitoring. Data fusion can be approached from many different viewpoints, as connected to the various research fields. In condition and process monitoring, data fusion is used to improve the accuracy of the diagnostics and prognostics results. Fusion can help to leverage the strengths and mitigate the weaknesses of sensors, signals, data types and methods to obtain improved information about the monitored system or asset.

3.1. Data fusion for condition monitoring

Industrial plants often contain a vast array of potential sources of data, ranging from quantitative sensor data used for the control and monitoring of the system, through to more disparate and qualitative sources of data such as event logs, operation logs, and maintenance data. The quantity of data that may be recorded and stored from a variety of disparate sources for condition and process monitoring purposes is increasing due to improvements in computing and sensing technologies. Heterogeneous data offers a number of opportunities for developing more reliable and robust monitoring algorithms (Lu et al., 2014). Heterogeneous data often contain complementary information, which facilitates the modelling of complex system interactions, non-linearities, different failure, and fault propagations and different process conditions (Hou and Bergmann, 2012). In order to leverage the advantages of heterogeneous data, extract meaningful and actionable insights and guide maintenance decisions, data fusion techniques are required to manage, fuse and process the data from disparate sources.

One of the first and most commonly used definitions of data fusion was given by Hall and Llinas (1997): "*Data fusion combines data from multiple sources to achieve improved accuracies than could be achieved by the use of a single data sources alone about the monitored system.*" This definition is further detailed by Niu and Li (2017), who added that the aim of data fusion is to obtain improved information in the meaning of higher quality or more relevant information. However, many other definitions of data fusion also exist in the literature. An attempt has been made by Boström et al. (2007), who reviewed and evaluated more than thirty different definitions of data fusion.

There are several surveys in the literature about data fusion, such as (Luo and Kay, 1989; Hall and Llinas, 1997; Dasarathy, 1997; Pohl and Van Genderen, 1998; Nakamura et al., 2007; Esteban et al., 2005; Khaleghi et al., 2013; Gravina et al., 2017) based on which, the types, abstraction levels, and methods are introduced in the rest of this section.

3.2. Types of data fusion

Data fusion can be applied for a multitude of purposes. There are several types of data fusion described in the literature on the basis of what data are being fused. Some data fusion applications use the same input data with various methods, others use data from several sensors, and there are also applications which fuse heterogeneous data.

- **Ensemble of classifiers, Mixture of experts, Ensemble based systems** refer to the case when several learning systems or classifiers are built using the same input data, each of them providing a monitoring result, which is then fused on the decision-level. They have been shown to outperform single-classifier systems in a broad range of applications and under a variety of scenarios (Polikar, 2006).
- **Multi-sensor data fusion** combines data from multiple sensors to achieve improved accuracies and more specific inferences that could be achieved through the use of a single sensor alone (Hall and Llinas, 1997).
- **Heterogeneous data fusion** fuses data from disparate sources. Input data of the fusion can contain not only quantitative sensor data (sampled at the same or different sampling rates), binary or categorical data, such alarms, and events, but also qualitative data, such as maintenance, fleet, and design data (Yuan et al., 2018).

3.3. Levels of data fusion

Data fusion is categorized into three groups based on the abstraction level, as data-, feature- and decision-level fusion (Hall and Llinas, 1997):

- **Data-level fusion** combines multiple homogeneous sources of raw data into a health index or feature. It is only possible if the raw data are synchronized, aligned and sampled at the same sampling frequency observing the same monitored asset or system Gravina et al. (2017).
- **Feature-level fusion** combines features extracted from different sensors and sources. Extracted features are then combined into a single feature vector which is the input to the fusion algorithm. The output of the fusion is the inference drawn from the whole dataset. Data alignment and synchronization must be considered and the joint feature vector must contain homogeneous values of either numeric, binary or categorical variables (Hall and Llinas, 1997).
- **Decision-level fusion** combines the decisions of several local classifiers or monitoring systems into one final result. The aim of decision-level fusion is to leverage the strengths and mitigate the weaknesses of the local monitoring systems by reinforcing the correct decisions and suppressing out the incorrect ones (Ghosh et al., 2011).

Another work categorized data fusion methods according to the inputs-outputs of the fusion (Dasarathy, 1997):

- **Data In - Data Out fusion** is usually referred to as data processing. It overlaps with the data-level fusion of the previously described categorization.
- **Data In – Feature Out** is usually referred to as feature extraction. It overlaps with the data-level fusion of the previously described categorization.
- **Feature In – Feature Out** is usually referred to as feature selection, which will be further discussed in Chapter 4.
- **Feature In – Decision Out** is referred as feature-level fusion in the previous categorization.
- **Decision In – Decision Out** is referred to as decision level fusion in the previous categorization.

The rest of the thesis uses the data-, feature-, decision-level categorization of the abstraction levels. The choice of the abstraction level depends on the information carried by the different signals or data types. If the system incorporates multiple homogeneous sensors measuring the same physical phenomena, then sensor data can be directly fused. If the signal types are significantly different and carry complementary information feature or decision-level fusion may yield better results. If the data types are not synchronized and are even more disparate, decision-level fusion may be the optimal choice for data fusion.

3.3.1. Data-level fusion

Data-level fusion directly combines raw sensor signals into a monitoring index. Figure 3.1 shows a schematic representation of data-level fusion. In a condition monitoring context, its aim is to better represent the health state of the system. While it is not the most researched area in condition monitoring or in sensor fusion, it has been successfully applied for degradation modelling and prognostics (Liu et al., 2013; Yan et al., 2016; Song et al., 2018).

The main advantage of data-level fusion is that the health indicator obtained by combining multiple signals better represents the level of degradation in the monitored component and is more interpretable by the end user. Hence, the originally multivariate monitoring problem can be reduced into monitoring a

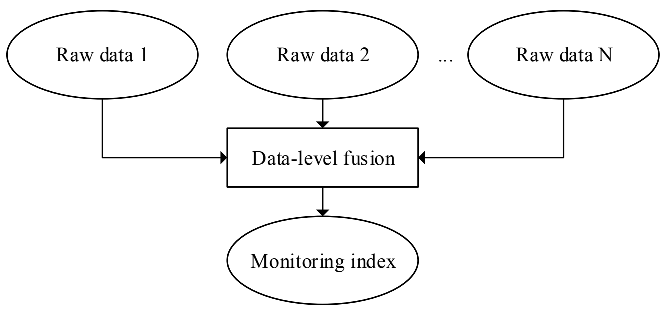


Figure 3.1: A schematic showing the inputs and outputs of data-level fusion

single indicator. The health indicator can be treated as a new sensor signal from the component, which may be further fused with other sensor data or features on the feature-level.

The methods of data-level fusion can be divided into two main categories: linear and non-linear fusion. Linear fusion, which uses a linear transformation to obtain the fused result, has been applied for example for monitoring the degradation of an aircraft gas turbine engine (Yan et al., 2016; Liu et al., 2013). While linear fusion can work for simple problems, complex systems might have non-linear relations with the degradation process and the operating conditions, hence non-linear methods show better potential for real-life problems. A non-linear kernel-based data-level fusion has been proposed in (Song et al., 2018), fusing temperature, level and speed signals for aircraft gas turbine degradation monitoring. They have observed that the sensors in condition monitoring systems usually contain only partial information about the health states, therefore better diagnostics results can be achieved by fusing multiple sensor signals into health indicators.

3.3.2. Feature-level fusion

Feature-level fusion is an efficient way to incorporate feature data from multiple sources into the diagnostics framework. Before the fusion, the variables or extracted features are joined in a higher dimensional feature vector, which is fed into a diagnostics algorithm performing the feature-level fusion. Figure 3.2 shows the inputs and outputs of feature-level fusion. The fusion provides the monitoring result, which is the current status of the monitored asset, often in the form of the diagnosed fault class. The advantage of feature level fusion is that it is possible to use different features from different data sources in order to incorporate a wider range of information into the monitoring algorithm. However, data with different sampling rates or unsynchronized data can cause extra pre-processing workload or even make the feature-level fusion impossible due to misaligned and missing observations in the joint feature set (Hall and Llinas, 1997). Over-fitting can occur in smaller datasets when the number of features is comparable to the number of observations. In such cases feature set reduction methods might be considered before the feature-level fusion to avoid poor diagnostics results.

Various condition monitoring methods which aim to increase the accuracy and robustness of fault detection via multi-sensor data fusion have been reported, where sensor fusion is conducted on the feature level. Feature extraction and selection might be conducted individually for the different sensor types resulting in a unified feature set, where ideally each observation is represented with all of the available features extracted from the different sensors.

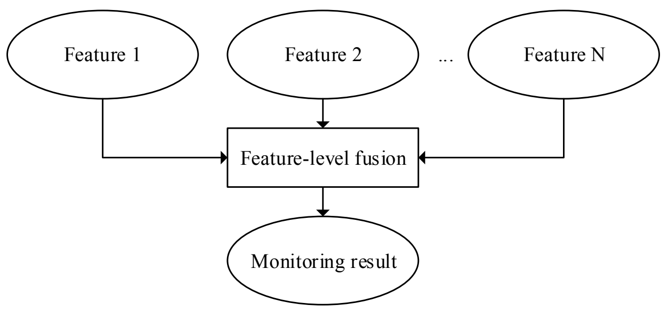


Figure 3.2: A schematic showing the inputs and outputs of feature-level fusion

A K-Nearest Neighbor (KNN) classifier was applied in (Safizadeh and Latifi, 2014) where accelerometer and load signals were fused for bearing fault diagnostics, and in (Dhami and Pabla, 2018) where acoustic and vibration signals were fused for gear fault diagnostics. Both papers showed that each signal type has its unique strengths and weaknesses when diagnosing various faults. They also compared the fusion results with the individual sensor diagnostics results concluding that multi-sensor feature-level fusion resulted in improved ability to detect, characterize, and identify fault conditions. Other rotating machinery diagnostics applications of feature level-fusion include the fusion of vibration, acoustic and oil-debris signals (Loutas et al., 2011), vibration and current signals (Arellano-Espitia et al., 2018) and vibration and process data (Ruiz-Cárcel et al., 2016). All of this research concluded that the diagnostic capability and reliability of the condition monitoring scheme were improved by the fusion.

Feature-level fusion in tool condition monitoring for milling also has great potential. The fusion acoustic and vibration signals in tool condition monitoring application has shown to achieve higher classification accuracy as compared to the individual data sources (Srinivasan et al., 2016). Zhou and Xue (2018) applied multi-sensor fusion acoustic, vibration, electric for tool condition monitoring in milling, using a kernel extreme learning machine and concluded that feature-level sensor fusion could provide more accurate information regarding tool condition. However, it was observed that there was a risk in the fusion that valuable information was overwhelmed by redundant information. Therefore, in multi-sensor fusion frameworks, sensor placement and feature selection is a critical component. Sensor fusion was investigated in the context of vibration analysis for tool condition monitoring in milling using Dynamic Bayesian Networks by He et al. (2018). They found that different sensor arrangements should be adopted for different monitoring problems and for various monitoring accuracy requirements of a machining process.

3.3.3. Decision-level fusion

Integrating data from different sources with fusion techniques into existing classification algorithms can improve the generalization and robustness of algorithms. Decision-level fusion combines monitoring results from monitoring systems to obtain a final monitoring decision. Figure 3.3 shows the inputs and outputs of decision-level fusion. It can be used to fuse monitoring results that may originate from monitoring systems using different data sources, using the same input data with different monitoring methods or from monitoring systems using data from different system components.

One of the advantages of decision-level fusion is that different data types of the monitoring systems

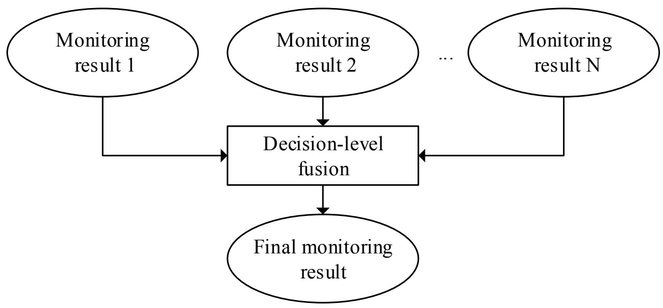


Figure 3.3: A schematic showing the inputs and outputs of decision-level fusion

do not have to be sampled at the same rate, do not have to be in the same format, and do not have to be fed into the same type of individual monitoring system. Hence, it is naturally well-suited for heterogeneous data and sensor fusion. Another important advantage is that by exploiting the strengths and mitigating the weaknesses of each data type, the final monitoring performance can be improved compared to the individual monitoring systems.

The most suitable methods for decision-level fusion in condition monitoring and FDD applications are voting-based methods, Bayesian methods, and the Dempster–Shafer method (Ghosh et al., 2011). Decision-level fusion is also applied in other fields, such as Structural Health Monitoring (SHM) where data is available from multiple sensors and they are placed at various locations of the monitored structure (Farrar and Worden, 2006). For SHM applications, besides the previously described decision-level fusion strategies, fuzzy inference (Wang et al., 2006) is also often applied. In other application fields, such as image processing and object recognition, Fuzzy sets (Zhu and Basir, 2006), Random sets (Mahler, 2004) and Rough sets (Yong et al., 2004) are also used for decision level-fusion, however, these methods have several disadvantages making them less appealing for condition monitoring applications requiring transparency. Fuzzy sets are limited solely to fusion of vague data, Random sets are relatively new and not well studied yet in the fusion community and Rough sets are only applicable to data with appropriate level of class separation (Khaleghi et al., 2013).

Decision fusion can be beneficial for building modular monitoring frameworks. It is less researched than feature-level fusion, especially multi-sensor data fusion on the feature-level. The available decision fusion methods are more limited. In this section, a review and short description is given about the currently existing and applied decision-level fusion methods in condition monitoring applications.

Voting-based fusion

A voting-based fusion framework consists of a set of classifiers, whose class predictions are aggregated on the decision-level to one final class prediction. Voting-based fusion has different versions based on how the class labels contribute to the final class vote (Ghosh et al., 2011). There are differences between the versions based on the level of agreement between the classifiers and if there is any order of importance, ranking or weighting between the classifiers.

- **Unanimous voting**: All classifiers have to agree on the final class label, otherwise the final class is not determined. This type of fusion is able to achieve low false alarm rates, however, it is often not practical due to the associated high missed alarm rates. Heterogeneous data might contain complementary information about the health state of the system and 100% agreement between the classifiers might not be possible if some of the classifiers do not receive sufficient information for diagnosing a certain fault.

- **Simple majority voting**: The final class label is the one that more than half of classifier has predicted reaching consensus. The false alarm rate is very low in this case, as simple majority guarantees reliable prediction. An observation may be classified as undetermined, if the classifiers did not achieve a simple majority consensus on any of the class labels. Missed alarms can also occur, however, the rate of missed alarms for the simple majority voting is lower than for unanimous voting.

- **Majority voting**: The final class prediction is the one that the majority of the classifiers agree on. An observation may be classified as undetermined when the classifiers vote for two or more labels with equal number of votes. If the number of classifiers is higher than the number of fault classes, this method can work well with a relatively low rate of undetermined class predictions.

- **Weighted voting**: In weighted voting, the votes of the classifiers do not equally contribute to the final class label prediction, rather each vote has its own assigned contribution weight. The weights can either be assigned to each classifier or to each classifier-predicted class combination based on the classifier performance on the training dataset or on a separate validation dataset. The weights represent the credibility or accuracy of the decision of the classifiers for a certain problem and can be assigned based on the performances of the classifiers during the training of the algorithm. Often the local classifiers not only give a class prediction but also a set of class probabilities, which can be used when assigning the weights. A review of weighted voting-based methods can be found in (Rahman et al., 2002), where a comprehensive summary is given about weight selection methods.

The first three voting-based methods treat the local classifiers equally without considering their individual performance. However, in reality, one classifier might outperform another due to more accurate training or due to a better classification model. Weighted-voting is able to take into account the accuracy of each classifier, as well as provide a solution for avoiding undetermined class predictions. Therefore, weighted-voting is superior to the first three methods. However, accurate classifier weighting is only possible if there is enough representative training data available to calculate the weighting.

Voting-based fusion methods are popular for decision fusion, as they are simple and easy to implement. They are applied to classification problems, which are commonly found in condition monitoring at the diagnostics level (see Chapter 2.3.4). A simple additive voting-based decision-level fusion was proposed for wind turbine fault diagnostics by Zappalá et al. (2019) fusing electrical and mechanical signals. They reported increased detectability of faults, increased robustness of the monitoring framework and increased trust of the operator in the monitoring system. Data fusion is also present for process condition monitoring applications. Majority voting is the most commonly used voting-based decision level fusion (Ghosh et al., 2011)) It is found, for example, in tool condition monitoring applications for the fusion of data from different sensors (Cho et al. (2010)), where it was found to outperform the simple feature level-fusion of sensor data. In (Zhang et al., 2018a) a novel method was proposed for monitoring a manufacturing process. The process was instrumented with numerous sensors, which were grouped into several clusters based on their correlation. The sensor data were fused on the feature-level with multivariate methods into a single monitoring statistic per group. The largest monitoring statistic of each group then formed one global monitoring statistic. This method is analogous to majority voting-based fusion when the result is in the form of monitoring statistics, not as fault class label. Condition monitoring applications of voting-based fusion can also be found when using an ensemble of classifiers with the same input data. Weighted-voting of classifier ensembles was applied by Dou et al. (2017) for rotating machinery fault diagnostics resulting in a better classification accuracy when compared to a single classifier. This was due to the fact that even if single classifiers provide the wrong classification, the ensemble classifier was able to obtain the correct result as long as the other classifiers obtained the correct result with higher weight.

Bayesian fusion

Bayesian fusion is a robust evidence-based method for decision fusion with an efficient classifier conflict resolving mechanism. Bayesian fusion combines the Bayes probabilistic model with a decision rule based on prior knowledge and the likelihood functions of class-specific fault diagnosis performance of the local classifiers. The final fault class decision is often selected based on the maximum a posterior probability (Ghosh et al., 2011).

Bayesian fusion in condition monitoring has also been applied to combine the results of diagnostics methods using the same input data, which is different from the heterogeneous data fusion approach where different sources of data are combined on the decision-level. A generic Bayesian decision-level approach combining the results of model-based and data-based diagnostics methods was proposed by Slimani et al. (2018). They showed that the advantage of the combination of heterogeneous diagnosis methods that it is efficient and easy to implement.

Fault diagnostic frameworks that use only a single source of data are prone to signal noise resulting in reduced diagnostics accuracy (Choi et al., 2009). Bayesian fusion has been described as a suitable way to tackle noisy data and implement robust diagnostic frameworks. Senanayaka and Robbersmyr (2018) implemented a decision-level fusion of current and vibration signals using a Bayesian decision-level strategy for synchronous motor fault diagnosis with and without added Gaussian noise to the sensor data. The method improved the diagnostics results and proved to be robust against noise. Another work also used the Bayesian method on the decision-level for the purpose of monitoring motors (Niu and Li, 2017) and fusion was found to over-perform the individual classifiers with a single source of data.

Bayesian decision-level methods are recently getting further attention in condition monitoring applications. They are modular, flexible and scalable providing transparency of decision for the end user. It can be applied not only for the fusion of different data types, but also for diagnosing multiple fault cases in a complex system, composed of several components where the faults may propagate between the components. Jaramillo et al. (2017) proposed such a framework for the diagnostics of coupled rotating equipment, which fuses the individual equipment data on the feature-level to get a component diagnostics results. The component diagnostics results are then fused on the decision-level to obtain the final system diagnostics result.

Dempster-Shafer evidence theory

The Dempster–Shafer theory is a generalization of the Bayesian probability theory based on evidential reasoning developed by first Dempster (1968) and later by Shafer (1976). It assigns belief and plausibility functions to decision results and fuses them according to Dempster's combinational rule to obtain the final conclusion. The belief function is the minimum probability of a decision being correct. The plausibility functions is the maximum probability of a decision being correct. These functions allow imprecision and uncertainty to be present in the data and in the fusion. Therefore, the main advantage of the method is the capability of robustly dealing with incomplete, imprecise data and uncertainty. However, there are also some disadvantages to the method, such as misleading results when fusing conflicting data or its high computational costs (Barnett, 2008).

Dempster-Shafer theory has been applied for fusing decisions of different diagnostic methods. Liu et al. (2018a) created a framework, which fused the diagnostic results for two data-driven feature-level fusion methods to diagnose failures of proton exchange membrane fuel cells. Lu et al. (2016) applied weighted Dempster–Shafer evidence theory for fusing sub-decisions of a model-based and a data-driven method for gas turbine engine diagnosis. Both papers report that such an approach can significantly improve the speed, robustness and diagnostic accuracy compared to a single method applied for feature-level fusion.

Sensor fusion algorithms where the classifiers are built on sensor types have also successfully applied using the Dempster-Shafer theory for the decision-level fusion. The fusion of vibration and acoustic signals resulted in more precise diagnostics along with reduced false and missed alarm rates when diagnosing faults in planetary gearboxes (Khazaee et al. (2014)). The fusion of vibration and current signals showed that these signal types are complementary to one another when diagnosing mechanical and electrical induction motor faults. The accuracy of the classification in the sense of reduced false and missed alarms is improved compared to the two individual classifiers when using Dempster-Shafer theory for their fusion (Yang and Kim, 2006).

Decision-fusion using Dempster-Shafer theory has also been successfully applied to a condition monitoring set-up, where a monitoring system is built for each individual sensor. The diagnostics results of the sensors are then fused on the decision level. Jiao et al. (2017) implemented such a framework for diagnosing faults in rolling bearings based on multi-source vibration sensors. The decision-level fusion of individual sensors improved significantly the overall monitoring accuracy compared to the classifiers built on the individual sensors.

Variations of the Dempster-Shafer can also be found. The belief function in (Yao et al., 2018) is weighted according to sensors. Weights are assigned to sensors using expert knowledge and the decisions originating from that sensor are weighted before fusion. The method was for the monitoring of centrifugal pumps based on vibration and pressure data and it proved to improve the classification accuracy compared to feature-level sensor fusion. The proposed method seems promising when expert knowledge is available for appropriate sensor weighting.

Comparison of decision-level fusion methods

Based on the review of the three different decision-level fusion methods, Table 3.1 gives a summary about the advantages and limitations of each. From the three methods, clearly the voting-based methods are the most simplistic. If the local classifiers do not reach consensus, voting-based methods may produce undetermined class labels as the fused result. Furthermore, unanimous, simple majority and majority voting techniques take into account all of the local classifiers equally without considering their performance. In case of incomplete, noisy and imprecise data the fusion may show reduced accuracy. On the other hand, voting-based methods are simple and easy to implement with transparent decisions and low computational costs.

One of the major differences between voting-based methods and Bayesian and Dempster–Shafer-based fusion is the possibility to include prior knowledge into the analysis. The latter two methods are capable of including prior knowledge. Bayesian methods are based on fusing the priors with the newly available evidence, while the Dempster-Shafer approach includes prior knowledge in the form of belief and plausibility functions. Bayesian and Dempster–Shafer-based fusion are also able to handle uncertainty. In Bayesian fusion, uncertainty is represented by the conditional probabilities, while Dempster–Shafer evidence theory uses the belief and plausibility functions for quantifying uncertainty. However, these two approaches have a common disadvantage which voting-based methods do not have. They are built on the assumption that the fusion of the local classifiers are based on independent evidence.

Table 3.1: Comparison of decision-level fusion methods for fault detection and diagnosis

	Advantages	Limitations
Voting-based fusion	Simple and easy to implement	Undetermined class labels in case of conflict
	Transparent	Not able to handle incomplete data
	Low computationally costs	Not able to handle noisy, imprecise data
	No independent evidence assumption	
Bayesian fusion	Handles uncertainty well	Not able to handle incomplete data
	Can handle noisy, imprecise data	Independent evidence assumption
	Transparency and interpretable decisions	
	Includes prior knowledge	
	Low computationally costs	
	Efficient classifier resolving mechanism	
Dempster-Shafer evidence theory	Able to deal with incomplete data	Independent evidence assumption
	Able to deal with noisy, imprecise data	Higher computational costs
	Includes prior knowledge	Misleading results with conflicting data
	Handles uncertainty well	Less transparent and interpretable decisions

Bayesian methods are able to robustly deal with noisy and imprecise data, similarly to Dempster–Shafer-based methods. However, Bayesian methods may have performance decrease when working with incomplete datasets, while Dempster–Shafer-based fusion is able to account for this. Bayesian methods have an advantage over Dempster–Shafer-based fusion, namely their transparency and interpretablility. Dempster–Shafer-based fusion methods are less easy to interpret due to the complexity introduced by their belief and plausibility functions. Regarding computational costs, voting-based and Bayesian methods have a lower computational burden, than Dempster–Shafer-based fusion.

A comparison study of the different decision-level fusion methods was conducted by Ghosh et al. (2011), who evaluated of the three decision-level fusion methods using a distillation column column case study and the Tennessee Eastman benchmark case. In both cases they observed that Bayesian and Dempster–Shafer-based fusion significantly outperformed the voting-based approach, achieving high prediction accuracy, a wide range of diagnosable faults and short diagnosis times. These two cases studies further proves that Bayesian and Dempster–Shafer-based fusion are capable of achieving similarly high performance on the same diagnostic problem (Cobb and Shenoy, 2003).

3.4. Summary of the advantages of data-fusion

From the review of data fusion, it is clear that there are a number of potential advantages that data fusion can offer. The use of any individual data-, feature- and decision-level fusion approach does not exclude the possibility of using them in combination. If the complexity of data fusion is well-adjusted to the complexity and structure of the monitored system it can provide the following advantages:

- Improved accuracy is the most often reported advantage of data fusion (Song et al., 2018; Jiao et al., 2017; Yao et al., 2018; Choi et al., 2009). Accurate diagnostics and prognostics enable more efficient, safer and more reliable operation of process plants, which results in economically feasible CBM systems.

- Effective distinction between faults also contributes to the improved accuracy of the condition monitoring system with low misclassification rate between faults (Safizadeh and Latifi, 2014; Dhami and Pabla, 2018).

- Reduced missed and false alarms are necessary for achieving accurate diagnostics and prognostics results and they also contribute to building the trust of the end-user in the decision support system (Yang and Kim, 2006; Khazaee et al., 2014).

- A wider range of diagnosable faults brings the possibility of more accurate diagnostics and prognostics (Loutas et al., 2011; Arellano-Espitia et al., 2018).

- Improved robustness of the condition monitoring system against changing operating conditions, noise or other external factors is also a key to achieve accurate results (Arellano-Espitia et al., 2018; Lu et al., 2016; Senanayaka and Robbersmyr, 2018).

- Well-informed maintenance decisions are important to achieve reduced downtimes and optimal system performance (Yao et al., 2018).

- Actionable insights about the monitored system are important to take well-informed maintenance decisions (Jaramillo et al., 2017).

3.5. Challenges and opportunities in data fusion for condition monitoring

Based on the literature review, heterogeneous data fusion for condition monitoring is a promising area where there are still many issues to be solved before holistic CBM solutions can be fully adopted. Heterogeneous data have to be treated with extra care from pre-processing to decision-level fusion. Therefore, this thesis focuses on the investigation of how to leverage the strengths and mitigate the weaknesses of disparate data from feature selection to decision-level fusion for component-level and plant-level monitoring. The research goals are listed as follows:

- Develop feature design and feature selection method from multi-sensor systems to obtain data representations which best suit the condition monitoring problems.

- Develop transparent, modular and scalable heterogeneous data- and sensor-fusion frameworks for fault detection and diagnostics, which are able to provide accurate diagnostics results with low false and missed alarm rates.
- Create heterogeneous data- and sensor-fusion frameworks that can adapt to changing operating and environmental conditions, which will further enable the transferability of the methods
- Develop methods for fusing data with different sampling rates and types in order to incorporate a wider range of data types for improved robustness and diagnostics accuracy

The rest of the thesis investigates, proposes and applies methods that are suitable for achieving the above listed research goals.

4. Feature ranking and feature selection for condition monitoring

In this chapter, the importance of feature selection in successful fault detection and diagnosis is discussed. Feature selection methods are also reviewed from the perspective of their applicability for condition monitoring and data fusion problems. The ReliefF method which has been found to be a suitable fit for condition monitoring applications is further studied. ReliefF is extended to cope with the issue of feature redundancy and a ReliefF-based hybrid method is proposed for feature selection. This chapter builds on the findings of Stief et al. (2018a, 2019b).

4.1. Introduction to feature selection

Feature design and selection is one of the first steps towards successful fault detection and diagnosis. Data from disparate sources often contain complementary information about the systems being monitored. One sensor might be more adept at detecting one fault or operation mode than another sensor. This may be due to differences in the type of the sensor used or simply due to differences in the physical location of two identical sensors relative to the fault source. Similarly, one feature derived from a signal recorded from a particular sensor might be capable of detecting one type of fault, while a different feature calculated from the same source might be more successful at detecting a different type of fault. Therefore, methods which fuse features from multiple sources can often detect and diagnose a greater number of fault modes with higher confidence.

Domain knowledge is a necessity for feature extraction in complex systems. Extraction of relevant information helps to reduce the dimensionality of the raw data. However, even when based on domain knowledge, feature extraction might yield correlated and redundant feature sets. Hence, it might be necessary to rely on additional data-driven methods to reach the best subset of features tailored to a monitoring problem (Onel et al., 2018). In the case of multi-sensor fusion for fault detection and diagnosis, feature selection has great importance, as there is a risk that valuable information may be overwhelmed by redundant information (Zhou and Xue, 2018).

Feature selection aims to obtain a subset of features from the original feature set, which are less correlated, less redundant and more relevant for a given data analytics task (Guyon and Elisseeff, 2003). It is successfully applied in multiple fields besides fault diagnosis (Rauber et al., 2015; Cerrada et al., 2015), such as bioinformatics (Wang et al., 2016), image recognition (Yao et al., 2017) and text mining (Tutkan et al., 2016) among others.

Feature selection aims to reduce the complexity of the monitoring algorithm, omit noisy, irrelevant and correlated features while retaining only the ones providing useful information for a monitoring problem (Alkhadafe et al., 2016). It can lead to a better understanding of the dataset, better learning performance, lower computational cost and more interpretable models (Chandrashekar and Sahin, 2014; Miao and Niu, 2016). Feature Selection methods can be categorized categorized according to their outputs, their degree of supervision and their selection strategies (Cai et al., 2018).

4.1.1. Result of feature selection

Feature selection methods can be divided into feature ranking and subset selection methods according to the type of result they produce as output.

- **Feature ranking** methods take the original feature set $\mathbf{x} = \{x_1, x_2, ..., x_m\}$ of size m and calculate a relevancy index $R(x_i)$ for each feature x_i based on which an order of relevancy can be determined among the original features. To obtain a reduced feature set, the relevancy index can guide the selection process. If the relevancy index of a feature is higher than a certain value τ, the feature gets selected to the reduced feature subset, otherwise it is not used further in the analysis.

- **Feature subset selection** methods directly obtain the reduced feature subset $\mathbf{x}' = \{x_1, x_2, ..., x_{m'}\}$ of size m' where $m' \leq m$. They may select the best subset of features based on both feature-class label correlation and feature-feature correlation (Huang, 2015).

4.1.2. Degree of supervision

Feature selection methods can be divided into supervised, unsupervised, and semi-supervised methods using labelled, unlabelled, or partially labelled datasets respectively.

- **Supervised selection** methods are convenient for classification problems as they use features and class labels of the observations to find a subset of the original features which better represent the dataset for a certain classification problem. Given a labelled dataset $\mathbf{d}_l$ with m features, n observations and $\mathbf{c}$ class labels $\mathbf{d}_l = \{\mathbf{x}_{n\times m}, \mathbf{c}_{1\times n}\}$, the supervised selection method aims to find a good feature subset $\mathbf{x}' \in \mathbf{x}$ within the original feature set $\mathbf{x} = \{x_1, x_2, ..., x_m\}$ that maximizes the classification accuracy (Cai et al., 2018).

- **Unsupervised selection** methods use only unlabelled data $\mathbf{d}_u = \{\mathbf{x}\}$ for reducing the dimension of the original dataset and finding features, which best represent the dataset. There are two approaches described in the literature. The feature-based clustering method aims to find clusters of correlated features and then select features from each cluster, which best preserve the structure of the original dataset (Pacheco et al., 2016). The observation-based clustering approach aims to find clusters among the observations, label the data by clusters and turn the unsupervised feature selection into a supervised feature selection problem. The feature-based clustering approach is usually substituted with multivariate approaches in condition monitoring applications, such as PCA, PLS or CVA, which are also able to consistently improve other traits of the dataset, such as ensuring that variables are independent and uncorrelated with one another, besides reducing the dimension of the dataset. The observation-based clustering approach has identical selection methods as supervised feature selection (Miao and Niu, 2016). Sometimes the two approaches are used together both clustering the features and the observations forming a bi-clustering problem (Liu et al., 2006).

- **Semi-supervised selection** methods work with datasets $\mathbf{d} = \{\mathbf{d}_l, \mathbf{d}_u\}$, which contain both labelled $\mathbf{d}_l = \{\mathbf{x}, \mathbf{c}\}$ and unlabelled $\mathbf{d}_u = \{\mathbf{x}\}$ data subsets. Semi-supervised feature selection methods are practical for cases where there is a small number of labelled data available and a larger amount of unlabelled data. They create an assumption about the optimal feature set using the labelled data subset $\mathbf{d}_l$, which is further refined using the unlabelled data $\mathbf{d}_u$. Although semi-supervised filter methods can achieve good classification performance and generalization ability, they are not commonly used in condition-monitoring applications due to their scalability and computational cost issues, which appears when applied to high-dimensional feature selection problems (Hu et al., 2017). A review of semi-supervised feature selection methods can be found in (Sheikhpour et al., 2017).

This thesis focuses on supervised diagnostic problems in condition monitoring applications where, based on the previously known fault conditions and health states, the current health state of the asset is determined. Hence, the rest of this chapter will discuss supervised feature selection methods to fit the supervised diagnostic problem. Unsupervised and semi-supervised feature selection methods are considered as out of the scope of this thesis.

4.2. Selection strategies

Feature selection strategies can be grouped into filter[1], wrapper, and embedded methods, which are introduced in the following section.

4.2.1. Filters

Filters rank and select the features without considering the optimization of the learning performance of the selected feature subset. They calculate a relevance index based on the association between the features and the class labels. Given a feature set $\mathbf{x} = \{x_1, x_2, ..., x_m\}$ and data $\mathbf{d}_l = (\mathbf{x}, \mathbf{c}))$, the filter calculates a relevance index $R(x_i)$ for each feature x_i, which expresses the relevance of the feature x_i for a given classification problem. The original features are ranked according to relevance indices $\mathbf{x}' = \{x'_1, x'_2, ..., x'_m\}$ in a way that the relevance indices are in order: $R(x'_1) \geq R(x'_2), ..., \geq R(x'_m)$. The selected features are the first k features with the highest relevance index. Filters are popular due to their low computational cost and statistical scalability, even though other methods may produce better feature subsets (Lazar et al., 2012). Based on the method of calculating the relevance index, filters can be categorized into similarity-based, information theory-based, and statistics-based methods (Li et al., 2018).

- **Similarity-based filter methods** use a similarity measure among features by class labels. They aim to select those features which are similar for the same label and are different for other labels, therefore they are able to distinguish between the class labels. Examples of similarity-based filter methods include those based on the Fisher score (Duda et al., 2012), those belonging to the Relief family using k-nearest neighbor based relevance ranking (Kira and Rendell, 1992b; Robnik-Šikonja and Kononenko, 2003) or those which use the trace ratio criterion (Nie et al., 2008). Similarity-based filter methods are able to efficiently work with continuous and discrete

[1] The term filter is used in the context of feature selection throughout this chapter and should not be confused with signal processing filters.

data. They can account for the multivariate contextual information between features. However, they may produce redundant features.

- **Information theory-based methods** use the entropy, conditional entropy, information gain and conditional information gain between features and class labels to measure the importance of features. Feature selection methods using mutual information aim to maximize feature relevance, where the relevance of a feature is the measure of its correlation with the class labels. Features which are correlated with class labels may be correlated with each other, therefore feature redundancy minimization is another aim of information theory-based methods. The Minimum Redundancy Maximum Relevance method (Peng et al., 2005) considers both the feature relevance and feature redundancy when producing feature subsets. Other Information theory-based methods used mutual information as a distance measure given by the Kullback–Leibler divergence (Torkkola, 2003). This family of methods works well with discrete features. For continuous features, extra quantization effort is required.

- **Statistics-based methods** calculate the relevance indices of features using various statistical measures such as correlation coefficients, which assess the degree of dependence of individual variables with the class labels. A variety of classical test statistics (T-test, F-test, Chi-squared, etc.) are suitable for relevance index calculation. Gini index is another measure which is able to quantify if a feature is able to provide good class separation (Gini, 1912). One of the simplest and most popular statistics-based criteria is the Pearson correlation coefficient which is only able to detect linear dependencies between variable and target (Guyon and Elisseeff, 2003). Statistics-based methods may also directly provide feature subsets using correlation-based feature selection. They rank the feature subsets according to a correlation-based heuristic evaluation function (Hall and Smith, 1998). Statistics-based methods are simple with very low computational costs, however, they often cannot handle feature redundancy and they only work with discrete data.

4.2.2. Wrappers

Wrapper methods select subsets of features and evaluate their predictive performance using a classifier as a black box evaluation function. The feature subset achieving the highest classification accuracy is selected as the final feature set. Wrapper methods can be divided according to the search strategy used into those which use exhaustive searches, forward selection, backward selection, heuristic, and random search. The computational effort can greatly differ between them. Additionally, the selected subset of features also may differ (Chandrashekar and Sahin, 2014). The most commonly used classifiers for wrappers are Support Vector Machine, k-Nearest Neighbors, Artificial Neural Network, and Bayesian classifiers (Ang et al., 2016).

- **Exhaustive search** evaluates all possible feature subsets. If there are m features in the dataset, an exhaustive search would require all of the 2^m subsets of features be evaluated. The exhaustive search leads to an exponentially growing computational time with the increasing number of features, which is infeasible even if there are more than 30 features to be searched. Therefore, suboptimal search strategies are needed to perform feature selection with limited computational efforts (Wang et al., 2016).

- **Forward selection** strategies start with an empty feature set. The relevant features are then subsequently added until the stop criterion (e.g. a pre-defined classification accuracy was achieved) is fulfilled resulting in the optimal feature set (Guyon and Elisseeff, 2006).

- **Backward selection** strategies start with a full feature set. The irrelevant or redundant features are subsequently removed until the optimal feature set is reached. Backward selection may achieve better classification performance compared to forward selection at the expense of possibly larger feature sets (Chandrashekar and Sahin, 2014).

- **Random search** strategies take a randomly selected subset from the original feature set, evaluate the selected subset and make a decision about either keeping the features or discarding and choosing a new subset (Wang et al., 2016). It may happen that random search quickly finds a global optimum. In less lucky scenarios, the computational cost of finding the optimal subset may be close to exhaustive search. The unpredictable computational cost of random search makes it less applicable to problems with high-dimensional feature sets.

- **Heuristic search** algorithms are used to heuristically test feature subsets to gather experience. The sub-optimal feature subsets are then improved by searching similar feature subsets in the state space. Heuristic search is also capable of innovations to include less optimal feature sets which allows the search to escape a sub-optimal subset until an optimal feature set is found. An example of a heuristic search is the Genetic Algorithm-based feature selection method (Punch et al., 1993). The objective function of feature subset evaluation is based on the predictor performance (Chandrashekar and Sahin, 2014; Wang et al., 2016).

Wrappers have huge computational demands compared to filters, however, they are able to achieve higher classification accuracy and produce smaller feature sets. They are prone to over-fitting and are highly dependent on the selected classifier. The selected features may achieve poor performance when used with other classifiers.

4.2.3. Embedded methods

Embedded methods select features as part of the training of a learning algorithm (e.g. classifier). The selected feature subset is created once the training has finished. They differ from wrappers in how the feature selection and learning algorithm interact, as embedded methods do not separate the training of a learning algorithm from the feature selection part. Embedded methods can be divided into pruning methods and sparse learning-based methods (Huang, 2015).

- **Pruning methods** train a learning algorithm with all features and remove features by setting the feature coefficients associated to zero while optimizing learning performance. Recursive feature elimination with SVM is an example of a pruning method which removes features recursively using SVM as the predictor black box (Guyon et al., 2002). The relevance of a feature is given by its SVM weights. Feature selection removes iteratively the least important feature according to their weights. Feature selection-perceptron is another embedded method based on a feed-forward neural network, where the weights of the perceptrons are used as relevance indices (Mejía-Lavalle et al., 2006).

- **Sparse learning-based methods** consider feature selection as an optimization problem. These methods work with a sparse regularization term to minimize the fitting errors of a classification problem and are capabale of achiving good performance and interpretability. The sparse regularization term calculates the feature coefficients, which can be treated as a relevance index. If a feature coefficient is very close to zero, the corresponding feature is not selected as part of the final feature subset. The least absolute shrinkage and selection operator (LASSO) method using l_p-norm regularization is suitable for binary classification problems (Tibshirani, 1996). The outputs of its sparse model can be interpreted as probabilities. Its extended version the $l_{p,q}$ regularizer is suitable for multi-class problems (Peng and Fan, 2016).

Embedded selection can often lead to good performance for a specific type of learning algorithm. Whilst higher than that of filter-based methods, the computational complexity of embedded selection methods is smaller than that of wrapper methods. Their drawback is similar as of wrappers, as the selected features do not necessarily achieve good performance in other learning tasks.

4.3. Feature selection for condition monitoring applications

This thesis is focuses on heterogeneous data fusion methods for condition monitoring applications, particularly for fault diagnostic problems with the assumption that historical datasets are labelled according to a health state. If not only the features within the dataset but class label information is also available for algorithm development, intuitively supervised methods are preferred, which are able to incorporate class label information from previous health states.

Filters are efficient due to their small computational burden and statistical scalability. Embedded methods and wrappers are more computationally expensive, as they use a black box method for evaluating various features subsets and selecting the features based on classification performance. Furthermore, they lack transparency. Hence, a filter approach is selected in this thesis. Statistics-based methods are simple with very low computational costs. However, they often cannot handle feature redundancy and only work with discrete data. Information-theory-based methods work well with discrete features. In the case of continuous feature data, which is usually the form that data recorded from sensors in condition monitoring applications take, an extra quantization effort would be required in order to convert the continuous data into discrete data. Similarity-based filter methods are able to efficiently work with continuous and discrete data. Similarity-based filter methods from the Relief family can account for the multivariate contextual information between features and they are able to deal with noisy and incomplete data.

4.4. A supervised feature ranking method: ReliefF

The original Relief ranks the features for two-class classification problems (Kira and Rendell, 1992b), while ReliefF extends Relief to multiclass problems (Kononenko, 1994). The Relief family is efficient, reliable and powerful in estimating the quality of attributes (Huang, 2015). It is also relatively simple and computationally cheap, making it attractive as a feature selection choice.

ReliefF was chosen for further investigation of the feature selection task for condition monitoring. It is a supervised, multivariate, feature selection filter which calculates relevance indexes for all features using a nearest neighbor-based joint relationship with the classification target. ReliefF usually performs better relative to other filter methods due to the fact that feature ranking is performed with a non-linear k-nearest neighbor search algorithm, which is able to account for the non-linear relationships between features (Sun, 2007). They are also a popular choice for feature ranking and selection due to their simplicity, low computation complexity, and effectiveness (Deng et al., 2010; Bolón-Canedo et al., 2013). These methods are able to correctly estimate the quality of features in classification problems with strong dependencies between features (Robnik-Šikonja and Kononenko, 1997; Huang, 2015). ReliefF is able to handle conditionally dependent and independent features with incomplete and multi-class data sets (Kononenko et al., 1997). It is also robust against noisy features (Kononenko, 1994).

The Relief algorithm takes all of the features and class labels for each observation as inputs and outputs a relevance index for each of the features. It starts by randomly choosing an observation and for each feature, it searches for the nearest neighbor in the same class (nearest hit) and the nearest neighbor in different classes (nearest miss). A relevance index is then calculated based on the Manhattan distance between the chosen and the found observations. Greater weights are given to those features which are close to one another in the same class, less weight is given to those features, which are close to one another in the different class. The resulting ranking is based on how well the features differentiate the observations of different classes,

$$W(x) = W(x) - \frac{\text{diff}(x, R_i, H_j)}{m} + \frac{\text{diff}(x, R_i, M_j)}{m} \tag{4.1}$$

where R_i is the chosen observation, H_j are the nearest hits, M_j are the nearest misses, m is the number of iterations defined by the user and x denotes a feature (Robnik-Šikonja and Kononenko, 1997).

The $\text{diff}(x, o_i, o_j)$ function for numeric features calculates the Manhattan distance between the values of the feature x for two observations o_i and o_j using Equation 4.2. It is also used to calculate the distances between observations in order to find nearest neighbors.

$$\text{diff}(x, o_i, o_j) = \frac{x(o_j) - x(o_j)}{\max(x) - \min(x)} \tag{4.2}$$

The multi-class ReliefF takes the k-nearest neighbors, and calculates the weights for each feature using Equation 4.3 (Robnik-Šikonja and Kononenko (2003)):

$$W(x) = W(x) - \sum_{j=1}^{k} \frac{\text{diff}(x, R_i, H_j)}{mk} + \sum_{c \neq \text{class}(R_j)} \frac{\left\{ \frac{P(c)}{1 - P(\text{class}(R_i))} \sum_{j=1}^{k} \text{diff}(x, R_i, M_j) \right\}}{mk} \tag{4.3}$$

The formula for calculating the weights is similar to the original Relief, except that the contribution of all hits and all misses are averaged, weighted with the $P(\mathbf{c})$ prior probability of the hit class and divided with $1 - P(\text{class}(R_i))$ prior probability of the other miss classes. Selection of k hits and misses ensures greater robustness against noise compared to the original Relief.

The ReliefF algorithm has three parameters: m, k and τ. Parameter m represents the number of iterations used for training the algorithm. If the training set is relatively small, it is a common choice to set m to the number of observations creating an exhaustive search (Kononenko, 1994). The number of nearest neighbors k is used for calculating the nearest k hits and k nearest misses, controlling the locality of the estimates. If k is too small, the algorithm may not be robust enough to deal with noise, if k is too large, the algorithm may not be able to capture the local dependencies between the features. The default value proposed in the literature is $k = 10$ (Robnik-Šikonja and Kononenko, 2003). Parameter τ is the feature relevancy threshold, which determines which features are selected based on the feature ranking. If the relevancy of a feature is below τ it is considered unimportant and is not selected. The upper bound for τ was proposed by Kira and Rendell (1992a) as

$$\tau \leq \frac{1}{\sqrt{\upsilon \cdot m}} \tag{4.4}$$

where υ is the probability of accepting an irrelevant feature as relevant. However, this bound for τ is conservative and in most of the cases much smaller values should be sufficient.

4.5. ReliefF for condition monitoring applications

Even though ReliefF is able to efficiently produce relevant feature sets, these features may be redundant and correlated (Zeng et al., 2013; Bolón-Canedo et al., 2013). In the literature some attempts have been reported to overcome this issue by combining ReliefF with information-theory-based methods (Zhang et al., 2008; Hancer et al., 2018), with multivariate methods, (Zeng et al., 2013; Khelf et al., 2012) and with statistics-based methods (Ding et al., 2018). However, these approaches do not consider the heterogeneous fusion data aspect and the design consideration that condition monitoring system is often required to be transparent and modular, as highlighted in Section 2.5. In this section modifications of the ReliefF algorithms are proposed to better suit the design considerations of condition monitoring applications using data from heterogeneous sources.

4.5.1. ReliefF with correlation removal for feature ranking

Ding et al. (2018) proposed a redundancy analysis-based feature selection method. Firstly, they applied ReliefF for feature ranking and then used a feature subset selection method based on Pearson's linear correlation coefficient. For feature x_a and x_b, the correlation coefficient is expressed by Equation 4.5. The correlation coefficient corr(x_a, x_b) is calculated for two randomly selected features. If the corr(x_a, x_b) value is greater than a predefined value, the feature with the higher ReliefF weight is selected. Ding et al. (2018) did not describe any stop criteria for the redundancy analysis, hence it is difficult to determine the applicability of the method in real-life applications. Another drawback may be due to the pair-wise comparison and randomness of the selection, which may result in feature subsets still containing redundant and correlated features.

To solve the correlation and redundancy issue of ReliefF in a systematic way, a new Pearson's linear correlation coefficient based re-ranking approach is proposed in this thesis based on the results of ReliefF. Pearson's linear correlation coefficient is the most commonly used linear correlation coefficient. It is defined between feature vector x_a with mean μ_a and feature vector x_b with mean μ_b as shown in Equation 4.5.

$$\text{corr}(x_a, x_b) = \frac{\sum_{i=1}^{m}(x_{a,i} - \mu_a)(x_{b,i} - \mu_a)}{\left\{\sum_{i=1}^{m}(x_{a,i} - \mu_a)^2 \sum_{i=1}^{m}(x_{a,j} - \mu_b)^2\right\}^{1/2}} \tag{4.5}$$

The correlation of two features have a value on the $[-1, 1]$ interval, where 1 indicates a total positive linear correlation, 0 means no linear correlation, and -1 indicates a total negative linear correlation.

The original feature set $\mathbf{x} = \{x_1, x_2, ..., x_m\}$ is ranked with ReliefF giving a rank R and weight W for each feature in the feature set: $\mathbf{x}\{W, R\} = \{x_1(w_1, r_1), x_2(w_2, r_2), ..., x_m(w_m, r_m)\}$. The weights and the ranks are modified using Equation 4.6 resulting in a re-ranked feature set $\mathbf{x}'\{W', R'\} = \{x_1'(w_1', r_1'), x_2'(w_2', r_2'), ..., x_m(w_m', r_m')\}$.

$$W_{\text{new}}(x_j) = W_{\text{old}}(x_j) \cdot \frac{\sum_{i=1}^{j-1} |(1 - \text{corr}(x_i, x_j))|}{j-1} \tag{4.6}$$

Firstly, the feature with the highest weight is selected. Then the correlation of the next feature with the highest weight is compared to the already selected features and this process is repeated for all features. The re-ranking process ensures that relevancy receives priority, however, features which are correlated with the more highly ranked features are penalised by reducing their weights. The top k features are selected above the feature relevancy threshold τ resulting in a reduced feature subset containing relevant and less redundant features. It has to be emphasized that this method only proposes a re-ranking algorithm and the actual feature selection task is still to be conducted by the user. The structure of the proposed method is shown in Figure 4.1.

The example shown in Figure 4.2 has four simulated features x_1, x_2, x_3, x_4 for two fault classes F_0 and F_1. The simulated features are drawn randomly from normal distributions with a standard deviation

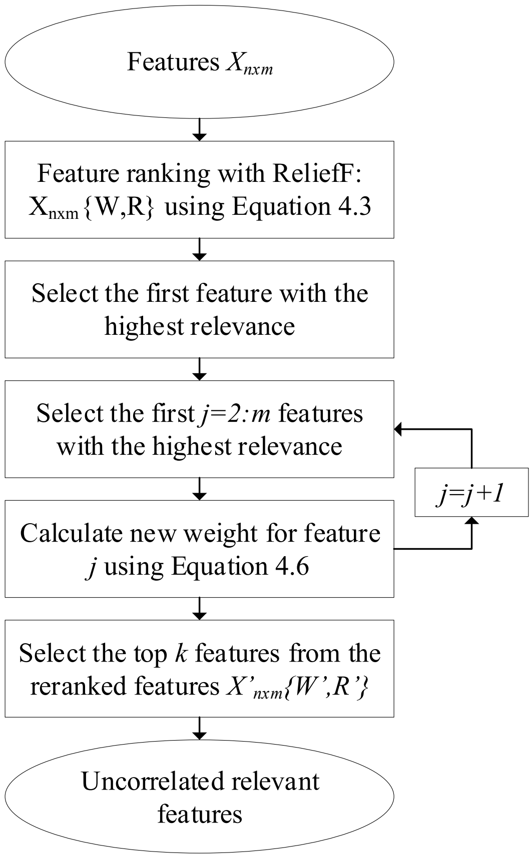


Figure 4.1: ReliefF with correlation removal for feature ranking

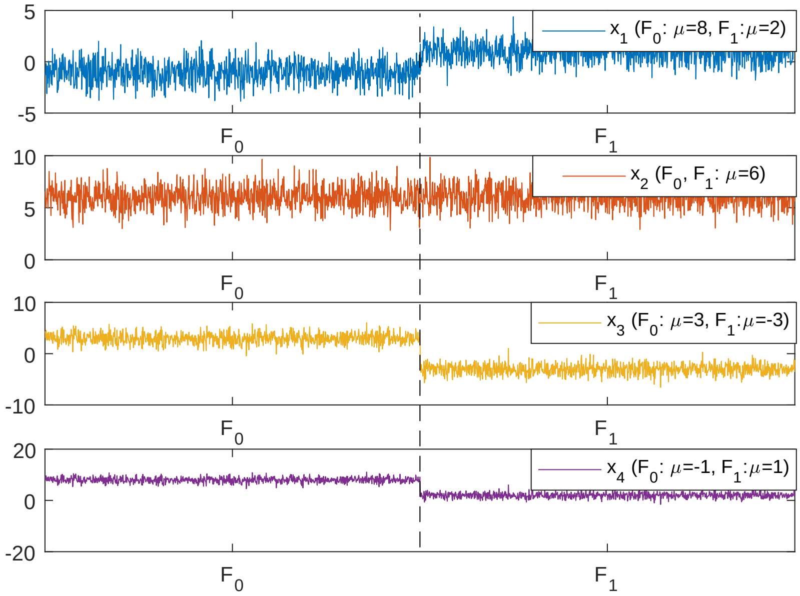


Figure 4.2: Simulated features for two fault classes F_0 and F_1

Table 4.1: Correlations between the simulated features

	x_1	x_2	x_3	x_4
x_1	1	-0.0026	-0.6649	-0.6649
x_2	-0.0026	1	0.0346	0.0346
x_3	-0.6649	0.0346	1	1
x_4	-0.6649	0.0346	1	1

Table 4.2: Ranks and weights of the simulated features with ReliefF, with and without correlation removal

	Rank with ReliefF	Weight with ReliefF	Rank with ReliefF and correlation removal	Weight with ReliefF and correlation removal
x_1	3	0.0943	2	0.0316
x_2	4	0.0309	3	0.0302
x_3	1	0.3113	1	0.3113
x_4	2	0.3113	4	0

of 1. The mean values for the features for fault class F_0 and F_1 differ from feature to feature, except for x_2, which has the same mean for both fault classes. Feature x_3 and x_4 are from the random same draw, with a mean shift between the two features. The correlation coefficients between the simulated features are presented in Table 4.1.

The features x_1, x_2, x_3, x_4 were both ranked with ReliefF with and without correlation removal. The calculated weights and ranks are shown in Table 4.2. The original implementation of ReliefF ranks x_3 and x_4 as the first two most important features, even though they are 100% correlated and their calculated weights are equal. ReliefF without correlation removal is able to account for the correlation between these two features and x_4 is ranked as the least important feature, as it does not provide any new information to the analysis.

4.5.2. A ReliefF-based hybrid approach for feature selection

Being unable to account for redundancy and correlation, ReliefF also only provides a feature ranking based on which the top k features are selected. Equation 4.4 describes τ as a rule-of-thumb method for determining k, the risk of selecting too many or too few features is an existing issue. In the literature, this consideration has received less attention, although an embedded version of ReliefF was proposed where the parameter optimization of the learning algorithm takes place during feature subset selection (Zhang et al., 2018c).

A hybrid filter-wrapper approach may be used for parameter optimization purposes finding the optimal number of features k to select (Guyon and Elisseeff, 2006). The features are ranked with the filter method and then the number of selected features is determined using a classification algorithm providing prediction accuracies, similarly as in the wrapper setting. This approach is computationally less expensive than the full implementation of the wrapper approach, as the evaluation of the classification performance is only calculated m times if there are m features in the original feature set. It was also proved that such hybrid filter-wrapper approaches are considerably less prone to over-fitting compared to wrapper methods (Ng, 1998).

Following this argument, the selection task in ReliefF is possible with such a hybrid filter-wrapper

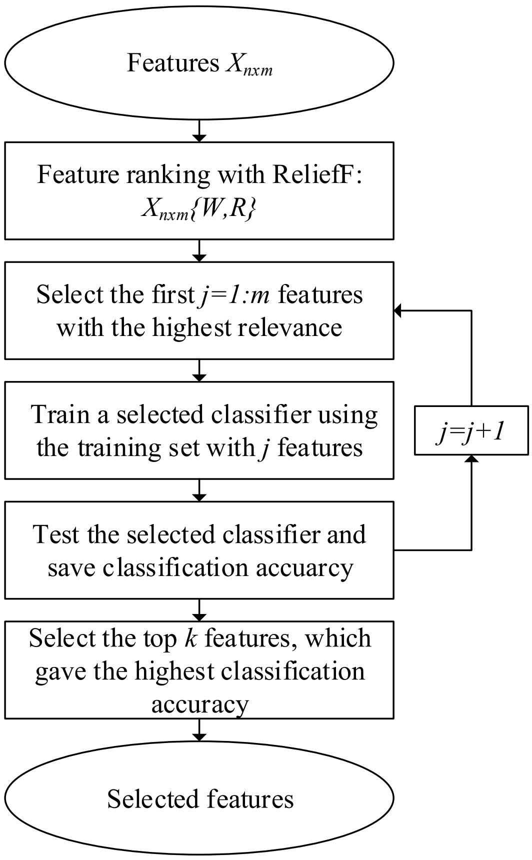


Figure 4.3: ReliefF-based hybrid approach for feature selection

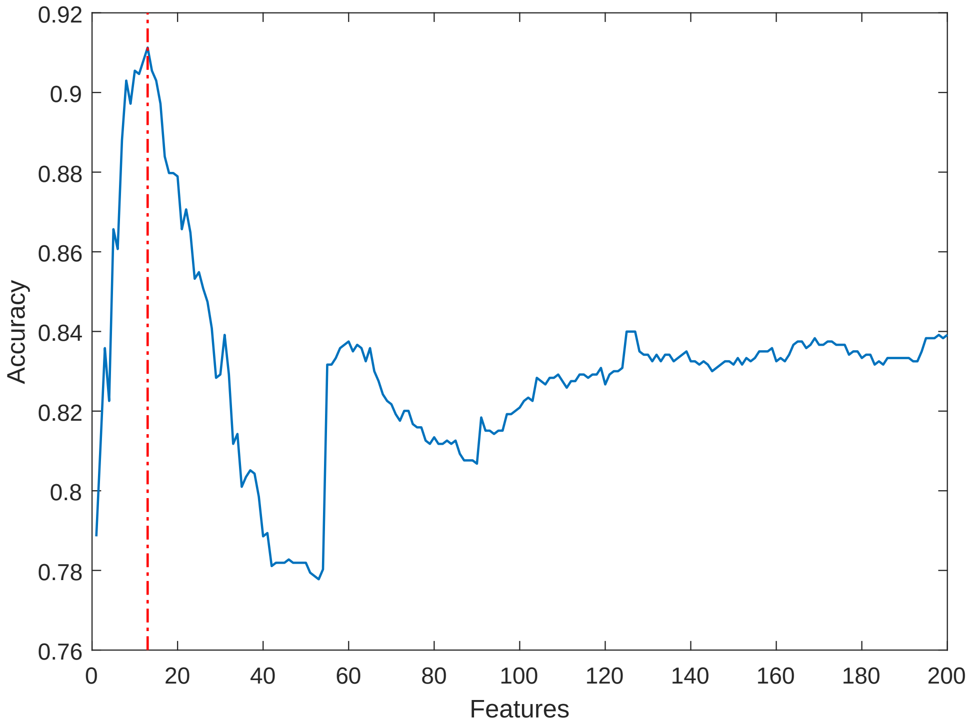


Figure 4.4: Classifier accuracies with respect to the number of features on an example after the ReliefF-based hybrid approach has ranked the features

approach. If there is enough data available, a validation set may be formed to find the number of features to select. Once the standard ReliefF method has ranked the features, a selected classifier may be trained and tested using the first k features and the classification accuracy of the classifier evaluated for a given feature subset. This may be subsequently repeated multiple times, with each successive model incorporating an additional feature, ordered in terms of ranking, until all of the features are selected. The feature selection is conducted based on the top k features which have achieved the highest classification accuracy. The flow diagram of the hybrid filter-wrapper approach is shown in Figure 4.3.

In Figure 4.4, an example shows the type of results the hybrid approach may produce after it has evaluated all of the features added one by one to the feature set. The classification accuracy is at its highest (91.1%) when the first 13 features from the ranking are included in the analysis. Hence, the optimal choice for the number of features to select is 13 in this example. If there is more than one maximum with the highest accuracy, the hybrid approach would select the feature set, which contains the least features.

4.6. Summary of feature selection

In this chapter, the concept and purpose of feature selection have been discussed. The importance of feature selection has been discussed with reference to condition monitoring and data fusion applications, where data may be redundant, irrelevant or noisy. After a review of various methods, ReliefF has been selected for further investigation. It is a computationally inexpensive, multivariate and supervised filter, which has been described to be efficient, reliable and powerful in estimating the quality of attributes (Kononenko, 1994; Sun, 2007; Huang, 2015). Even though ReliefF has such several favourable properties, it is not able to take into account for the correlation between the features and it is only able to provide a ranked feature set. Hence, in this chapter modifications have been proposed to solve the issue of redundancy and feature selection. The correlation and redundancy aspect of ReliefF is addressed using the Pearson's linear correlation coefficient to re-rank the features and produce non-redundant feature sets. Furthermore, a hybrid filter-wrapper approach is proposed for efficient feature selection. These modifications enable the ReliefF algorithm to better suit the design considerations of condition monitoring applications using data from heterogeneous sources.

5. Bayesian data fusion for diagnostics

This chapter focuses on Bayesian data fusion methods. The Bayes theorem, its relation to Bayesian hypothesis testing and probability estimation are introduced along with Naive Bayes classifiers. The different applications of Bayesian methods are discussed with relevance to condition monitoring applications. The challenges of Bayesian fusion are discussed, such as naive independence assumption of features, setting of priors, computational costs and the training of a Bayesian framework. A generic two-stage Bayesian framework is proposed, which is composed of a feature-level fusion and a decision-level fusion stage of the feature-level fusion results. The various feature- and decision-level fusion methods are also introduced. The challenges of two-stage Bayesian framework are discussed with regards to avoiding over- and under-fitted models. Furthermore, two methods are proposed to account for the operating condition dependency of signals and features when using the two-stage Bayesian framework, which is a typical condition monitoring challenge. Finally, the chapter concludes with a summary of the proposed two-stage Bayesian framework with a discussion on the applicability of the different feature- and decision-level fusion algorithms. This chapter builds on the work reported in Stief et al. (2017, 2019c, 2018c,b, 2019a).

5.1. Introduction to Bayesian methods

Bayesian methods are a well-known and powerful tool for reasoning under conditions of uncertainty. Bayesian inference has been described as a suitable method for fault detection and fault classification in condition monitoring systems, where uncertainty is often present (Heng et al., 2009; Tidriri et al., 2016). Bayesian methods quantify uncertainty in the form of probabilities. These methods use prior knowledge and historical data to obtain the probability of an event and update the posterior probability of an event every time there is a new piece of information or evidence available. For example, if the health state of an asset is related to the RMS of vibration measured by a sensor, the RMS of a vibration sensor can be used to assess the probability that a monitored asset has a fault using Bayes theorem. This concept will be further investigated for condition monitoring applications with regards to data available from multiple sensors with multiple features.

5.1.1. Bayes theorem

Bayes theorem is named after Thomas Bayes, who defined conditional probability to use evidence for the calculation of an unknown parameter (Bayes et al., 1763). However, it was Pierre-Simon Laplace, who used conditional probability to update a posterior probability from a prior probability creating the basis of Bayesian inference (Laplace, 1812). Bayesian inference provides a formulation to draw statistical conclusions about a parameter, hypothesis or assumption A. These statistical conclusions are based on observed data and prior knowledge (Gelman et al., 2013). Given assumptions, A and B, if the probability of A is greater than zero, $P(A) > 0$, then the probability of assumption B given assumption A is called the conditional probability and is expressed as (Kolmogorov, 1956):

$$P(B|A) = \frac{P(AB)}{P(A)} \tag{5.1}$$

where $P(AB)$ is the joint probability:

$$P(AB) = P(B)P(A|B) = P(A)P(B|A) \tag{5.2}$$

The Bayes theorem expresses how the probability of an assumption changes with the availability of new evidence. Given assumption A with a prior degree of belief $P(A)$ and evidence B, the Bayes theorem provides a formula to calculate how the probability of assumption $P(A)$ changes to $P(A|B)$ with B evidence:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{5.3}$$

$P(B|A)$ is the *likelihood* function or the conditional probability of evidence B being observed when assumption A is true. $P(A|B)$ is the *posterior probability* of assumption A given that evidence B has been observed. $P(A)$ is the *prior probability*, which represents the initial degree of belief in A. $P(B)$ is the probability of observing evidence B. Furthermore, supposing $A_1, A_2, \ldots, A_n$ are a set of mutually exclusive assumptions and whose sum forms the certain event ($P(S) = 1$),

$$S = \sum_{j=1}^{n}(A_j) \tag{5.4}$$

then if any of them have a probability greater than zero $P(A_i) > 0$, the following marginal probability is true

$$P(B) = \sum_{j=1}^{n} P(B|A_j)P(A_j) \tag{5.5}$$

The Bayes theorem can be obtained by the conditional probability and marginal probability:

$$P(A_i|B) = \frac{P(B|A)P(A)}{\sum_{k=1}^{n} P(B|A_j)P(A_j)} \tag{5.6}$$

In this form, the Bayes theorem provides a method to obtain the posterior probability of an event from prior assumptions based on newly observed evidence. The Bayes theorem can be adopted for a fault diagnosis example. If the prior probability of a particular fault A_i is known and the likelihood or conditional probability of an observed symptom B given the fault A_i can be estimated from historical data, then it is possible to compute the posterior probability $P(A_i|B)$ of fault A_i given the symptom B.

5.1.2. Bayesian hypothesis testing

Suppose that there are two alternative hypotheses H_0 and H_1. Both hypotheses have a prior degree of belief $P(H_0)$ and $P(H_1)$ and they are mutually exclusive $P(H_0) + P(H_1) = 1$. A random variable

x was observed and its distribution is known for the two hypotheses based on historically available data: $P(x|H_0)$ and $P(x|H_1)$. With the help of the Bayes theorem, it is possible to obtain the posterior probabilities of H_0 and H_1.

$$P(H_0|x) = \frac{P(x|H_0)P(H_0)}{P(x)}, P(H_1|x) = \frac{P(x|H_1)P(H_1)}{P(x)} \tag{5.7}$$

In classical statistics, hypothesis testing is formulated in such a way that a decision is made about the acceptance or rejection of the null hypothesis H_0 based on the significance level. In Bayesian hypothesis testing to decide between the two hypothesis H_0 and H_1, it is possible to compare the posterior probabilities $P(H_0|x)$ and $P(H_1|x)$, accepting the one with the higher posterior probability. This decision rule is called the maximum a posteriori (MAP) test. H_0 is chosen if and only if

$$P(H_0|x) \geq P(H_1|x) \tag{5.8}$$

which can be further rewritten as

$$P(x|H_0)P(H_0) \geq P(x|H_1)P(H_1) \tag{5.9}$$

The MAP test can be generalized to cases when there are more than two probable hypotheses. The MAP test will again decide on the hypothesis H_i which have the highest posterior probabilities $P(H_i|x)$ from all the hypotheses or equivalently has the highest $P(x|H_i)P(H_i)$. For classification problems a hypothesis H_i will be called a class c_i.

5.1.3. Probability estimation

Given a set of $\mathbf{x} = \{x_1, \ldots, x_m\}$ observations, which from now on are referred to as features, and two possible hypotheses, which from now on are referred to as classes $\mathbf{c} = \{c_1, c_2\}$. To estimate the probability of which class **x** belongs, we can formulate a binary classification problem. If we not only want to predict the correct class label but also obtain the probability, then, in this case, the problem is called binary class probability estimation or binary classification. This probability estimation may be conducted using Bayesian probability estimation or logistic regression (Gelman et al., 2013).

Bayesian probability estimation

Bayes theorem states that the posterior probability of $P(c_1|x_1, \ldots, x_m)$ can be obtained from the prior class probability of $P(c_1)$ and the likelihood function $P(x_1, \ldots x_m|c_1)$ of the features $\mathbf{x} = \{x_1, \ldots, x_m\}$ observed given class c_1. The same equation can also be obtained for class c_2.

$$\begin{aligned} P(c_1|x_1, \ldots, x_m) &= \frac{P(c_1)P(x_1, \ldots x_m|c_1)}{P(x_1, \ldots, x_m)} \\ P(c_2|x_1, \ldots, x_m) &= \frac{P(c_2)P(x_1, \ldots x_m|c_2)}{P(x_1, \ldots, x_m)} \end{aligned} \tag{5.10}$$

where the numerators are the joint likelihoods. They can be rewritten in the following form:

$$\begin{aligned}
P(c_1, x_1, \ldots, x_m) &= P(x_1, \ldots, x_m, c_1) \\
&= P(x_1 \mid x_2, \ldots, x_m, c_1) P(x_2, \ldots, x_m, c_1) \\
&= P(x_1 \mid x_2, \ldots, x_m, c_1) P(x_2 \mid x_3, \ldots, x_m, c_1) P(x_3, \ldots, x_m, c_1) \\
&= \ldots \\
&= P(x_1 \mid x_2, \ldots, x_m, c_1) P(x_2 \mid x_3, \ldots, x_m, c_1) \ldots P(x_{m-1} \mid x_m, c_1) P(x_m \mid c_1) P(c_1)
\end{aligned} \tag{5.11}$$

A full Bayesian model have the exact joint likelihood functions between the features and classes, however sometimes a simplification is considered. If we suppose conditional independence between each two features within each two classes, then the following form is true:

$$\begin{aligned}
P(x_i | c_1, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_m) = P(x_i | c_1) \\
P(x_i | c_2, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_m) = P(x_i | c_2)
\end{aligned} \tag{5.12}$$

This is called the naive conditional independence assumption, which further simplifies Equation 5.10.

$$\begin{aligned}
P(c_1 | x_1, \ldots, x_m) = \frac{P(c_1) \prod_{i=1}^{m} P(x_i | c_1)}{P(x_1, \ldots, x_m)} \\
P(c_2 | x_1, \ldots, x_m) = \frac{P(c_2) \prod_{i=1}^{m} P(x_i | c_2)}{P(x_1, \ldots, x_m)}
\end{aligned} \tag{5.13}$$

Since the denominator is constant given the input, the following class probabilities are estimated:

$$\begin{aligned}
P(c_1 | x_1, \ldots, x_m) \propto P(c_1) \prod_{i=1}^{m} P(x_i | c_1) \\
P(c_2 | x_1, \ldots, x_m) \propto P(c_2) \prod_{i=1}^{m} P(x_i | c_2)
\end{aligned} \tag{5.14}$$

Observation **x** may be classified as c_1 or c_2 based on the posterior class probabilities using the maximum a posteriori test (Equation 5.8).

Probability estimation with logistic regression

Regression techniques can be thought of as Bayesian posterior inference based on a uninformative priors not assuming any prior knowledge about the distributions for the parameters of a normal linear model (Gelman et al., 2013). A binary classification problem can also be formulated as an optimisation problem, aiming to find a function that minimises the error of misclassification on observations drawn from the same distribution that generated the training data. Binary logistic regression (Cox, 1958) decides between two possible classes $\mathbf{c} = \{c_1, c_2\}$ based on the observed features $\mathbf{x} = \{x_1, \ldots, x_m\}$, by estimating the logarithm of the odds of two events with a multiple linear regression function. The function has a value of 0 if class c_1 is predicted as with a probability of 1 and has a value of 1 if class c_2 is predicted with a probability of 1. The log-odds are then converted to probability using the logistic function.

The formulation of logistic regression given there previously were n observations of m features:

$$\text{logit}(p) = \log \frac{P(c_1)}{1 - P(c_1)} = \beta_0 + \beta_1 x_{i,1} + \beta_1 x_{i,2} + \cdots + \beta_m x_{i,m} \tag{5.15}$$

where $i = 1, \ldots, n$. The β_i parameters of the model can be obtained by an iterative optimization algorithm called the gradient descent. For further reading on logistic regression used for multi-class problems, the readers may refer to Gelman et al. (2013).

Bayesian methods and logistic regression

Even though both Bayesian methods and logistic regression are able to estimate the posterior probabilities of an observation with m features belonging to a certain class, the two methods use different assumptions. In the case of a full Bayesian model, the exact joint likelihood functions between the features and classes $P(x_i|c_1, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_m)$ have to be known. This assumption is overcome by a strong independence assumption of features of the naive Bayes formulation. In the case of logistic regression, feature independence is not assumed. Bayesian methods calculate likelihoods based on the historically observed features, while logistic regression fits a model to the observed data using an optimization algorithm. Hence, from this perspective, Bayesian probability estimation provides better model transparency and modularity. Furthermore, Bayesian inference is naturally able to deal with a multi-class problems, while logistic regression works for binary-class problems in its original form. For more information on Bayesian methods and logistics regression, the readers are guided to (Gelman et al., 2013; Ng and Jordan, 2002; Duda et al., 2012).

5.2. Naive Bayes classifier

Classification is a fundamental problem in fault diagnosis, where the goal of the diagnostics algorithm is to construct a classifier given a set of historical observations with known fault classes and to give correct fault diagnostics results from new observations. The Naive Bayes (NB) classifier has been described as an accurate fault diagnostics method (Heng et al., 2009; Tidriri et al., 2016). The NB classifier is a set of algorithms based on the Bayes theorem. The classifier is called naive as it assumes conditional independence between every two features within a class (Equation 5.12), which greatly simplifies the learning problem. Although the independence assumption is often not valid in real-world applications, in practice NB classifiers are able to achieve comparable results to the results of more complex classifiers, such as neural networks (Rish, 2001; Mitchell, 1997). Zhang (2004) gives a detailed theoretical analysis of why naive Bayes classifiers work well in real life applications. Furthermore, NB classifiers provide results in the form of class probabilities and the results can be traced back to the root-cause of the decision.

The Naive Bayes classifier is defined as follows for multi-class fault detection problems. An observation $\mathbf{x} = \{x_1, x_2, ..., x_m\}$ with m features belongs to class c_i out of fault classes $\mathbf{c} = \{c_1, c_2, ..., c_p\}$ with a posterior probability given by

$$P(c_i|\mathbf{x}) = \frac{P(c_i) \prod_{j=1}^{m} P(x_j|c_i)}{\sum_{k=1}^{p} P(c_k) \prod_{j=1}^{m} P(x_j|c_k)} \tag{5.16}$$

where $P(c_i)$ is the prior probability, $P(x_j|c_i)$ is the likelihood function (conditional probability that a data point x_j belongs to class c_i.) Since the denominator is constant in Equation 5.16 given the input, it can be simplified to:

$$P(c_i|\mathbf{x}) \propto P(c_i) \prod_{j=1}^{m} P(x_j|c_i) \tag{5.17}$$

The predicted fault class c_{pred} is calculated as follows:

$$c_{\text{pred}} = \arg\max_i \left\{ P(c_i|\mathbf{x}) \right\} \tag{5.18}$$

5.2.1. Likelihood functions

The likelihood function can be formulated in numerous ways according to the assumed distribution of the features within a class. The various versions of the naive Bayes classifier differ mainly by how these $P(x_j|c_i)$ likelihood functions are calculated. In the following section, three possible formulations are introduced, the first two versions assume a Gaussian and Bernoulli distribution, while the last one is using a non-parametric method called the Kernel Density Estimation to obtain the likelihood functions. Other versions of the NB classifier such as the Multinomial Naive Bayes classifier (Kibriya et al., 2004), the Complement Naive Bayes classifier (Rennie et al., 2003) are not discussed here, as they are mostly used in text classification applications.

Gaussian Distribution

A Gaussian Naive Bayes (GNB) classifier assumes that all features are conditionally independent between each classes and that they are distributed according to Gaussian distributions. The classifier learns the $P(x_j|c_i)$ conditional probabilities that a given feature value x_i belongs to class c_i from the training dataset. By assuming a Gaussian distribution of the features, the conditional probabilities may be obtained using the values of mean and standard deviation of the labelled training data for each class:

$$P(x_j|c_i(\mu_{i,j}, \sigma_{i,j}) = \frac{1}{\sigma_{i,j}\sqrt{2\pi}} e^{\frac{-(x_j-\mu_{i,j})^2}{2\sigma_{i,j}^2}} \tag{5.19}$$

For further references describing GNB classifiers, readers are guided to, for example, (Duda et al., 2012; Mitchell, 1997; Friedman et al., 1997).

Bernoulli Distribution

A Bernoulli Naive Bayes classifier assumes that features are conditionally independent between each classes, that they follow multivariate Bernoulli distributions and that the features have binary values $(0/1)$. The classifier learns the $P(x_j|c_i)$ conditional probabilities that a given feature value x_i belongs to class c_i from the training dataset. The likelihood functions may be obtained as:

$$P(x_j|c_i) = P(i|c_i)x_j + (1 - P(i|c_i))(1 - x_j) \tag{5.20}$$

which not only takes into account the presence of a binary feature i when calculating the posterior probability of class c_i but also it takes into account the case when feature i is not present.

Kernel Density Estimation

Let **x** $= \{x_1, x_2, \ldots, x_n\}$ be a sample drawn from a distribution with an unknown density f. To estimate the shape of f using the **x**, its kernel density estimator can be calculated in the following way for any data point x:

$$f(x) = \frac{1}{nh} \sum_{k=1}^{n} K\left(\frac{x - x_k}{h}\right) \tag{5.21}$$

where n is the number of samples, h is the bandwidth. The bandwidth controls the degree of smoothing of the estimation. If the underlying density being estimated is normal, the optimal choice for h is given by Equation 5.22 (Bowman and Azzalini, 1997; Silverman, 1986),

$$h = \left(\frac{4\sigma^5}{3n}\right)^{-1/5} \approx 1.06\sigma n^{-1/5} \tag{5.22}$$

where σ is the standard deviation of the input data and n is its size.

$K(z)$ is a non-negative Kernel function. In this work the normal density function is used:

$$K(z) = \frac{e^{-z^2/2}}{\sqrt{2\pi}} \tag{5.23}$$

The $P(x_j|c_i)$ likelihood functions for each feature x_j in each fault class c_i may be calculated with KDE using the following formulation:

$$P(x_j|c_i) = \frac{1}{nh} \sum_{k=1}^{n} K\left(\frac{x_j - x_k}{h}\right) \tag{5.24}$$

where the training data are used to determine the bandwidth h and the shape of the KDE function.

5.2.2. Advantages of Naive Bayes classifiers

Naive Bayes classifiers have a number of advantages that particularly make them well-suited to condition monitoring applications. Each new observation from the data acquisition system can incrementally decrease or increase the confidence of a hypothesis regarding the health state of the monitored asset. In this way uncertainty, which may arise due to conflicting information or lack of information, is accounted for in the condition monitoring system in a formalized manner. Therefore, Naive Bayes classifiers provide a more flexible way to handle uncertainties than methods that immediately eliminate a hypothesis if it is found to be inconsistent with the rest of the observations (Mitchell, 1997).

Bayesian methods are similar to human reasoning, they are able to incorporate prior knowledge with observed data to obtain a final hypothesis. Bayesian methods are able to combine hypotheses. If there are many features available, new observations may be classified by combining the predictions of the multiple hypotheses of features. The outcome of a Naive Bayes classifier may be used as an input of another classifier (Mitchell, 1997). Hence, Bayesian formulation is well-suited for not only feature-level fusion but also decision-level fusion.

5.2.3. Challenges of Naive Bayes classifiers

Besides the numerous advantages, Naive Bayes classifiers also face a few challenges, mostly related to the calculation of likelihood functions, computational costs, the selection of priors and the naive conditional independence assumption. Based on historical data the likelihood functions of features may be determined for each fault class. However, these features may have non-Gaussian distributions. Furthermore, in condition monitoring applications, features are often load dependent. The available training data may not be representative for new operating conditions and the data may not be labelled by the operating condition. Therefore, the selection of the calculation method of the likelihood functions may depend on prior knowledge about the monitored system, its behaviour, and its expected operating conditions. If the expected distribution of the likelihood functions is known, the likelihood functions can be directly calculated. If no fair assumption can be made about their expected distribution, the likelihood functions have to be approximated.

A practical difficulty of Naive Bayes classifiers is the significant computational cost required to determine the probability density functions and the posterior probabilities if the number of features is high or the training set contains many observations (Mitchell, 1997). This issue is addressed later in Section 5.5.4.

Another challenge of Bayesian methods is the setting of prior probabilities. When the priors are not known in advance, they are often estimated based on historical data and expert knowledge, and assumptions are made about the form of the underlying distributions. In a Bayesian formulation, the prior probabilities define the trade-off between false positives and false negatives (Gorinevsky, 2015). The priors, depending on their information content, may be divided into three categories: uninformative, weakly informative and informative priors.

Uninformative priors do not assume any prior knowledge about the probability of occurrence of the various fault cases. Uninformative priors are uniform priors where each fault class has equal probability. If there is no historical data or expert knowledge available to define the priors, uninformative priors are often used in Bayesian frameworks.

A weakly informative prior expresses partial information about a variable or the distribution of the fault classes. Such priors can be used for regularization to constrain inferences within a reasonable interval. These type of priors might be used when expert knowledge is available in the form of an approximate estimation of the fault distributions, for example, it is known that a type of fault is the most common among all the fault classes but its exact ratio is not available. A weakly informative prior may also be used when estimating a variable with a known physical limit to its value. For example, a level measurement cannot be negative, then this piece of information can be incorporated into the Bayesian framework in the form of a weakly informative prior.

If the historical data is a representative sample for the occurrence of faults, priors can be set proportional to their occurrence in the historical data. These priors are called informative priors. Another example of an informative prior is when the distribution of an estimated variable is known beforehand and the prior can be set according to this distribution.

Lastly, it has to be mentioned that the naive conditional independence assumption between features, which naive Bayes classifiers are based on, almost never holds for real-world applications and data sets (Zhang, 2004; Lewis, 1998; Rish, 2001; Domingos and Pazzani, 1997). In condition monitoring applications such conditional independence is hardly possible where features are originating from the signals which are often monitoring similar and dependent physical attributes of the same asset. Even though this independence assumption is often not true, in practice naive Bayesian classifier may perform well even when feature dependences are present (Domingos and Pazzani, 1997; Zhang, 2004). However, datasets with strong feature dependence may lead to over-fitted models, especially when the rate of dependence between the features are not similar within the different fault classes.

5.3. Bayesian methods for condition monitoring

There are various Bayesian methods, which are based on the Bayes theorem (Equation 5.3), such as Naive Bayes classifiers, Bayesian Networks, Bayesian parameter estimation methods, Bayesian hypothesis testing, and Bayesian model selection among others. These methods all use Bayesian inference, which has been described as a suitable method for fault detection and fault classification in condition monitoring systems (Heng et al., 2009), (Tidriri et al., 2016). This section gives a short introduction about the various types of Bayesian methods used for condition monitoring applications.

5.3.1. Applications of Naive Bayes classifiers

The Naive Bayes classifier, which is based on a series of Bayesian inference steps, is referred to as a pragmatic approach for fault diagnosis in the literature. Its main advantages lie in its simple structure, low computation effort, and high accuracy. Furthermore, NB has a relatively good performance even with a small amount of training data (Sharma et al., 2015; Zhang et al., 2018b). NB has been applied for fault diagnosis in rotating machinery applications, such as bearing faults diagnosis (Zhang et al., 2018b; Kumar et al., 2014a), gearbox diagnosis (Yu et al., 2018; Vernekar et al., 2017) and stator fault diagnosis of induction motors (Asfani et al., 2013). NB classifiers are also present in process monitoring applications (dos Santos et al., 2014). A Gaussian Naive Bayes (GNB) classifier has been compared with other classification approaches on the TEP simulated dataset with missing data and it was found that the GNB approach is sensitive to the missing data imputation values resulting in decreased classification accuracies (Askarian et al., 2016). The applications of NB classifiers can also be found in the fault diagnosis of welded joints (Kumar et al., 2014b), fault diagnosis of power transmission lines (da Silva et al., 2018; Swetapadma and Yadav, 2016) and milling tool condition monitoring (Madhusudana et al., 2016).

5.3.2. Applications of Bayesian networks

A Bayesian network (BN) (Pearl, 2014), often referred to as a Bayesian belief network, is a probabilistic inference network suitable for dealing with problems containing uncertainty (Jensen et al., 1996). A BN is a directed acyclic graph consisting of a set of nodes. These nodes represent a set of random variables, which are connected by directional arrows quantifying the causal relationship between the nodes (Pearl, 2014). BNs are used in condition monitoring applications as they provide a convenient way to quantify expert knowledge in the form the causal relationship between faults and fault symptoms (Xu, 2012; Lampis and Andrews, 2009). A three-layer configuration incorporating expert knowledge, faults, fault symptoms, and operating conditions was proposed for detecting faults in flexible rotors by Xu (2012). The three-layer configuration is able to account for combinational fault modes, which means that more than one fault may occur in the same time. Another application of BNs implemented a fault diagnosis solution for heating, ventilation, and air conditioning systems (Verbert et al., 2017), where BNs were constructed for operating modes based on both expert knowledge about component interdependencies and conservation laws and on historical data. The results showed that infromation about component interdependencies included in the analysis had contributed to more accurate fault diagnosis. Lampis and Andrews (2009) proposed a generic fault diagnosis framework which was able to conveniently convert fault trees into BNs and represent the monitored system with a single network constituted of sub-networks for each system component. The probabilities of the component failures showed how each component contributed to the observed fault symptoms. BNs are also present in tool condition monitoring applications. A dynamic Bayesian network was proposed to monitor a machining process and analyse the effects of the sensor locations on the fault detection accuracy (He et al., 2018). It was shown that BNs were effective in modelling both dynamic and static signals with randomness and uncertainties when predicting surface roughness. Condition monitoring of photovoltaic, wind and hydroelectric energy systems also have used BN-based methods in the literature. A wide review of the applications of BNs for fault detection and diagnosis in the field of renewable energy can be found in (Borunda et al., 2016). BNs are also applied in process condition monitoring applications. A naive BN was proposed in (Verron et al., 2006) for process fault identification using the Tennessee Eastman process simulator.

5.3.3. Other applications of Bayesian methods in condition monitoring

Other applications of Bayesian methods are also present in condition monitoring literature. Bayesian hypothesis testing has been applied to fuse multivariate data for anomaly detection and fault isolation in jet engines (Gorinevsky, 2015). For each new observation hypothesis testing was used to decide if the observation was abnormal or healthy. The result presented in (Gorinevsky, 2015) showed an order of magnitude improvement in the fault detection accuracy compared to regression models. Bayesian hypothesis testing has also been used to remove noise from condition monitoring data to improve the fault diagnostics accuracy in turbo-machinery (Xu et al., 2016). Directly assessing imperfections in the vibration signals through the ratio of Bayesian posterior odds for noise removal proved to be a robust method against over-denoising. Another work (Loutas et al., 2013) applied Bayesian error bound estimation of noise to account for noise related uncertainties in the RUL estimation of rolling bearings.

Fluctuating loads and operating conditions is a significant challenge in condition monitoring, especially when relatively little prior knowledge is available about the monitored asset. An application of Bayesian model selection was proposed by Heyns et al. (2012) to remove the non-fault related components and obtain residuals from vibration signals which are robust against fluctuating load and operating conditions.

Prognostics application of Bayesian methods can also be found in the literature. In (Chen et al., 2012), a Bayesian solution was proposed to calculate the degree of belief in the prognostics results obtained by a neuro-fuzzy system prognostic to forecast the evolution of machine faults with time. The Bayesian estimation algorithm adopted the predicted data from a neuro-fuzzy system as prior information, combined it with online sensor data and recursively updated the degree of belief in the prognostics results using the Bayes theorem. The recursive computation of the posterior probability density function was conducted with particle filtering to approximate the optimal Bayesian solution (Chen et al., 2012).

5.4. Two-stage Bayesian framework to fuse heterogeneous data

Bayesian reasoning is an efficient tool for data fusion in condition monitoring applications, as it is modular, flexible and scalable, granting transparency about the final decision of the algorithm for the end user in a probabilistic form. As described in Section 3.3.3, Bayesian fusion provides a robust evidence-based method, which can also be applied for decision-level fusion. Furthermore, Bayesian fusion at the decision-level grants an efficient classifier conflict resolving mechanism. Recently, Jaramillo et al. (2017) proposed a two-stage Bayesian inference approach to monitor the condition of a system composed of several subsystems. A system of a motor connected to a gearbox was analyzed, considering each individual component as a separate subsystem. The first stage of the sensor fusion took place at the subsystem level, while the second stage fused the result of the first stage at the decision-level in order to determine the health state of the whole system. The method was efficient in diagnosing faults in complex systems composed of interacting components (Jaramillo et al., 2017).

In this thesis, a two-stage Bayesian framework is proposed to combine information from multiple, diverse condition monitoring systems for fault diagnostics purposes. The flow diagram of the framework is shown in Figure 5.1. The framework is in line with the typical operation of a condition monitoring system described in Section 2.4 and also shown in Figure 2.11. Data may originate from various sources differing only in terms of what characteristics of the system the sensors are measuring, as described in Section 2.2. Data may originate from various data acquisition systems providing various condition monitoring data differing in terms of sampling or in the ways these data are acquired, as described in

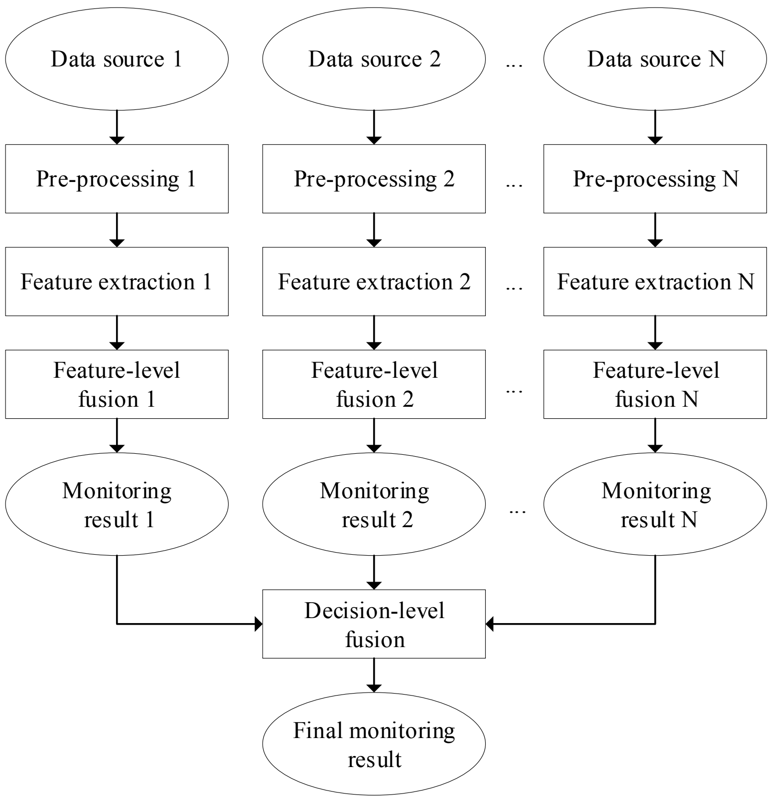


Figure 5.1: A two-stage Bayesian framework for fault diagnostics

Section 2.3.1.

After the raw data are grouped by data type, they are pre-processed and features are extracted separately. Each feature-level fusion algorithm takes the features from one data type as inputs and provides a monitoring result in the form of an assessment of the health state of the monitored asset. The monitoring results of the feature-level fusion algorithms are fused on the decision-level to obtain the final monitoring result.

5.5. Bayesian feature-level fusion

5.5.1. Bayesian feature-level fusion with thresholds

Recent work (Jaramillo et al., 2017) proposed a Bayesian inference approach, where the Bayes-theorem is interpreted in the following way: the probability that a fault F_i occurred in the system, given that a feature y_k crosses its threshold is

$$P(F_i|y_k) = \frac{P(y_k|F_i)P(F_i)}{P(y_k)} \tag{5.25}$$

where $P(y_k|Fi)$ contains the likelihood of feature y_k crossing its threshold, given it belongs to F_i. $P(F_i)$ is the prior probability of fault F_i and $P(y_k)$ is the probability of a feature crossing its threshold. The probability that an observation is classified as fault F_i is calculated in the same manner taking into account all of the features. If there are m features $\mathbf{y} = \{y_1, ..., y_m\}$ exceeding their respective thresholds the probability that fault F_i occurred is

$$P(F_i|\mathbf{y}) = \frac{P(\mathbf{y}|F_i)P(F_i)}{\sum_{j=1}^{m} P(y_j|F_i)} \tag{5.26}$$

The Bayesian interpretation of threshold-based feature-level fusion is further refined in this section by incorporating a posterior probability update step even for those features which did not exceed their respective thresholds.

Threshold setting with KDE

Threshold setting is one of the most crucial aspects of every condition monitoring system, as the accuracy of the end result is highly connected with setting the appropriate thresholds. The thresholds may be calculated using Kernel Density Estimation (see Section 5.2.1), which is described as an accurate method when the exact distribution of the data is unknown (Jaramillo et al., 2017; Chen et al., 1998; Odiowei and Cao, 2010).

Kernel density estimators are constructed from healthy data for each feature allowing a confidence interval to be determined. Confidence interval may be set either based on the expert knowledge or using the 1, 2, 3 σ rule resulting in a 68%, 95% or 99.7% confidence intervals. They also may be set on the basis of an optimization step that selects the value of the confidence interval using a validation set for which the highest diagnostics performance is observed. The end of each confidence interval is set as a threshold so that any feature value that exceeds the threshold would trigger an alarm.

Feature-level fusion

The posterior probability update formula (P_k^u), which is used for feature-level fusion, is based on the assumption that a feature either crosses its threshold with a probability of $P(y_{i,k}|F_i)$ or not with a probability of $(1 - P(y_{i,k}|F_i))$. Threshold setting allows the features to be transformed into binary values with a 0 or 1 outcome. As a result it is appropriate to use the Bernoulli Naive Bayes described by Equation 5.20. The following algorithm calculates the posterior probabilities, where n is the number of fault cases and $P^-(F_i)$ is the prior probability associated for fault case F_i. If N features out of M exceed their threshold, the prior probability is updated by the following equation:

$$\begin{cases} P_k^u = \frac{P(y_{i,k}|F_i)P^-(F_i)}{\sum_{n=1}^{N} P(y_{i,k}|F_i)P^-(F_{i,n})} \\ P^-(F_i) = P_k^u \end{cases} \tag{5.27}$$

If the $P^-(F_i)$ probabilities are only updated when a feature exceeds its associated threshold, the probabilities will be biased towards predicting faults more often, even in the case of healthy systems, leading to an increase in false alarms. Therefore, for the remaining $M - N$ features which do not exceed their threshold, the prior probability is updated by the following equation:

$$\begin{cases} P_k^u = \frac{(1-P(y_{i,k}|F_i))P^-(F_i)}{\sum_{n=1}^{M-N}(1-P(y_{i,k}|F_i))P^-(F_{i,n})} \\ P^-(F_i) = P_k^u \end{cases} \tag{5.28}$$

5.5.2. Bayesian feature-level fusion with Gaussian Naive Bayes classifier

Properly tuning alarm thresholds can be challenging, particularly when there are a large number of features in the data set, or when the thresholds themselves might optimally be described as a function of other parameters (e.g. operating conditions). Bayesian feature-level fusion does not always require monitoring thresholds to be set. The posterior fault class probabilities may be directly calculated if the likelihood functions are defined based on the distributions present in the training data. Often sensor measurements follow Gaussian distributions due to measurement noise (Liggins II et al., 2017), in these cases, a Gaussian assumption about the distribution of features can simplify the calculation of the likelihood functions. Gaussian Naive Bayes classifiers (Equation 5.19) provide a way to calculate the likelihood function from the historical data. It is very computationally efficient and does not require much storage space for the trained model, as the mean and standard deviations from the training set are enough to obtain the likelihood functions and posterior probabilities for any new observations (Stief et al., 2019c).

5.5.3. Bayesian feature-level fusion with KDE

Fault detection and diagnosis (FDD) methods often face the challenge that there is no prior knowledge about the distribution of the monitored variables and features. In order to simplify the problem, data-driven methods based on statistics often assume a Gaussian distribution (Stief et al., 2019c). However, this simplification might cause deterioration in the final fault detection accuracy. Hence, non-parametric methods are often preferred (Jaramillo et al., 2017). Kernel Density Estimation (KDE) is a powerful non-parametric method. Its main advantage lies in the fact that no prior assumption about the distribution of the data is necessary and it is able to handle multimodal distributions in an effective way. Once the $P(x_j|c_i)$ likelihood functions for each feature x_j in each fault class c_i are obtained using Equation 5.24, the posterior probabilities may be directly calculated for any new observations.

5.5.4. Bayesian feature-level fusion with Interpolated Kernel Density Estimate

KDE is a very computationally expensive method requiring linear storage proportional to the training data Zhou et al. (2003). Even if it is possible to store large amounts of data, the evaluation of KDEs requires a number of multiplications proportional to the size of the dataset, which can be infeasible due to high evaluation times (Heinz and Seeger, 2006). This is especially important for embedded applications.

There has been some ongoing research in the last two decades aimed at proposing Kernel Density Estimation methods which are less computationally expensive. These works focused mostly on applications using data streams with KDE. Data streams are a large volume of data arriving continuously. The data streams are analyzed online without storing them Zhou et al. (2003). Processing such data streams requires small and constant processing times for each incoming data record.

The first group of algorithms which were able to provide linear computational time and fixed size memory for KDE computation belong to the M-Kernel family first proposed by Zhou et al. (2003). The concept behind M-Kernels is that simple kernels are merged into M-Kernels by a weighting factor for computing KDEs over one-dimensional data streams. Variations of M-Kernels, such as (Heinz and Seeger, 2006), improve M-Kernels in terms of accuracy and processing time by introducing a new cost measure for optimal kernel merge. Other improved versions of KDE for faster data stream processing include Dm-Kernels (Xu et al., 2014), the SOMKE method (Cao et al., 2012) and cluster kernels (Heinz and Seeger, 2008). Recently Sodkomkham et al. (2016) proposed a method that compresses kernels on the basis of the distances between merging components and its most similar neighbors. Their aim was to create a new kernel compression method which is incremental, namely when a new data point arrives it can be directly included to the KDE model without retraining the whole model. They have reported low compression latency, small errors and improved computation time compared to the original KDE. These methods work well with data streams, however, in case of FDD problems, these methods are not applicable, as even in the case of online evaluation, the new observation are unlikely to form part of the training set.

In this section, an Interpolated Kernel Density Estimate (IKDE) is proposed to reduce the computational cost of KDE for FDD problems (Stief et al., 2019a). Furthermore, the proposed method aims to reduce the storage space required to store the trained KDE model. IKDE is based on interpolated KDE functions using Chebyshev polynomials and the barycentric formula to obtain the posterior probabilities. *Chebfun*, which an open source software package was used for the implementation of the method (Trefethen, 2013; Driscoll et al., 2014). The main advantage of this approach is the reduction in the stored data and the reduced number of multiplications needed for a single evaluation.

Chebyshev interpolation

Chebyshev interpolation is a type of polynomial interpolation that uses Chebyshev nodes in the determination of the interpolating function. The Chebyshev nodes are obtained by projecting equally spaced points on the unit circle down to the unit interval $[-1, 1]$. The Chebyshev nodes are the extreme points of the Chebyshev polynomials and can be expressed as

$$x_j = \cos(\frac{j\pi}{N}), j = 0, ..., N \tag{5.29}$$

Using Chebyshev nodes and sampling from function f, the Nth order interpolation polynomial is obtained, which is unique.

There are two main advantages of using Chebyshev interpolation. The first is that it is immune to Runge's phenomenon, i.e interpolation quality deterioration with the order of interpolation (Trefethen and Weideman, 1991). The second and more important advantage is the fact that Chebyshev interpolants are exponentially convergent (in a sense of supremum norm) to the analytic functions. In particular, if a function is analytic on a certain Bernstein ellipse (ellipse at a complex plane, with foci at $(-1,\ j0)$ and $(1,\ j0)$), the interpolation error fulfils

$$\|f - f_N\| \leq \frac{4M\rho^{-N}}{\rho - 1} \tag{5.30}$$

where ρ is the sum of semiminor and semimajor axes of the Bernstein ellipse, and M is the maximum value of f at that ellipse (Trefethen, 2013).

Barycentric formula

In order to efficiently compute values of an interpolation polynomial, it is convenient to use the Barycentric formula. Considering an interpolation polynomial with $N+1$ nodes x_j and sampled values $y_j = f(x_j)$, then the value of the interpolation polynomial at x is equal to

$$L(x) = \frac{\sum_{j=0}^{N} \frac{w_j}{x - x_j} y_j}{\sum_{j=0}^{N} \frac{w_j}{x - x_j}} \tag{5.31}$$

where w_j are the Barycentric weights which are unique for every set of nodes. In case of Chebyshev nodes in the $[-1, 1]$ interval they are given by (Berrut and Trefethen, 2004):

$$w_j = \begin{cases} 1/2, & j = 0 \quad \text{or} \quad j = n. \\ (-1)^j, & \text{otherwise.} \end{cases} \tag{5.32}$$

Regardless of the chosen precision, the computation of Equation (5.31) requires $O(2N)$ multiplications.

Interpolated Kernel Density Estimate (IKDE)

The Interpolated Kernel Density Estimate (IKDE) is proposed to reduce the computational cost and improve the processing times of the original KDE. IKDE approximates the KDE functions using Chebyshev interpolation. The barycentric formula is used to obtain the posterior probabilities. The flow diagram of IKDE is shown in Figure 5.2.

Firstly, the approximation coefficients ($\mathbf{x}_{cheb}$, $\mathbf{y}_{cheb}$, $\mathbf{w}_{cheb}$) are obtained when building the IKDE function for a data vector $x \in [a, b]$ from the training set.

1. The first step is to calculate the bandwidth with a selected method. As the focus of this section is not the bandwidth estimation, but the presentation of the new IKDE method, a simple and commonly

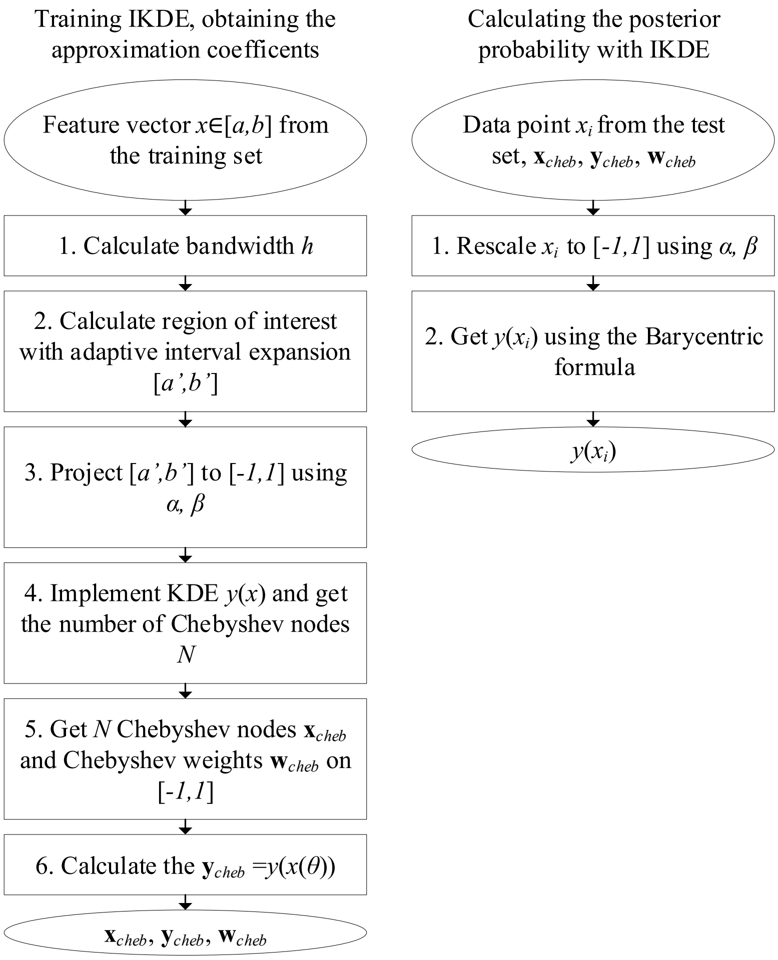


Figure 5.2: The flow diagram for IKDE: Training IKDE for a data vector **x** and determining the posterior probability for a data point x_i

used rule-of-thumb bandwidth estimation strategy is used to obtain h, as described by equation 5.22.

2. The region of interest is calculated for the data vector **x** by using an adaptive interval expansion strategy. Let $y(\mathbf{x})$ be the real kernel function of **x**, and $y(x_i)$ the posterior probability of a data point i belonging to the distribution $y(\mathbf{x})$. Let a be the minimum, b the maximum and σ the standard deviation of **x**. To achieve a certain accuracy ε is used as tolerance. In this thesis $\varepsilon = 10^{-15}$ is used. The adaptive interval expansion works in the following way:

$$\textbf{do}\text{: } a = a - \sigma \ \textbf{while}\text{: } y(a - \sigma) < \varepsilon$$
$$\textbf{do}\text{: } b = b + \sigma \ \textbf{while}\text{: } y(b + \sigma) < \varepsilon$$

Once the interval expansion has stopped, the interval of interest $[a', b']$ is obtained.

3. The next step is to project the $[a', b']$ interval to the $[-1, 1]$ interval. This step simplifies the calculation of the Chebysev nodes and weights. To describe the linear combination, which takes $x \in [a', b']$ to $\theta \in [-1, 1]$, $x(\theta)$, $\theta(x)$ and the projection coefficients have to be calculated in the following way:

$$x(\theta) = \alpha\theta + \beta, \quad \theta(x) = \frac{x - \beta}{\alpha} \tag{5.33}$$

$$a' = -\alpha + \beta, \quad b' = \alpha + \beta \tag{5.34}$$

$$\alpha = \frac{b' - a'}{2}, \quad \beta = \frac{a' + b'}{2} \tag{5.35}$$

4. The next step is to implement the $y(\mathbf{x})$ KDE function for the data **x** on the original $[a', b']$ interval using Equations (5.21) and (5.23). In this thesis the normal kernel function was used. Once the function is implemented, the number of Chebyshev nodes n is determined to approximate the given KDE function with an error less than 10^{-16} (machine precision).

5. Then, the $\mathbf{x}_{cheb}$ Chebyshev nodes (Equation (5.29)) and the corresponding $\mathbf{w}_{cheb}$ barycentric weights (Equation (5.32)) can be obtained on the $[-1, 1]$ interval.

6. Finally, the $\mathbf{y}_{cheb} = y(\mathbf{x}(\theta))$ KDE function values can be calculated using the rescaled $x(\theta)$ values in the $y(\mathbf{x})$ KDE function.

The outputs of the trained IKDE function are the interval scaling parameters α, β, the Chebyshev nodes $\mathbf{x}_{cheb}$, values $\mathbf{y}_{cheb}$ and weights $\mathbf{w}_{cheb}$, which can be used to calculate the posterior probability for any x_i in a fast and efficient way involving only two steps:

1. Firstly, x_i has to be rescaled to the $[-1, 1]$ interval using the scaling parameters α, β to obtain $\theta(x_i)$.

2. Then, the Barycentric formula (Equation (5.31)) is used to calculate the posterior probability $y(\theta(x_i)) = L(\theta(x_i))$:

$$y(\theta(x_i)) = \begin{cases} 0, & \text{if } \theta(x_i) \notin [-1, 1]. \\ \mathbf{y}_{cheb}(\theta(x_i)), & \text{if } \theta(x_i) \in \mathbf{x}_{cheb}. \\ L(\theta(x_i)), & \text{otherwise.} \end{cases} \tag{5.36}$$

Simulated numerical example with KDE and IKDE

A simulated dataset is used to compare how the computational time varies with the number of data points for KDE and the number of Chebyshev nodes for IKDE. The data points are randomly drawn from a normal distribution with 0 mean and 0.1 standard deviation. For KDE, the number of samples drawn for the training set is increased between 1 to 2000. The elapsed CPU times are measured and saved for the calculation of the posterior probability of each random data point. For IKDE, the number of Chebyshev nodes is increased between 1 to 2000. The Chebyshev nodes, weights and values are calculated. The elapsed CPU times are measured and saved for the calculation of the posterior probability of a random data point using the Barycentric formula. These simulations were repeated 100 times for both KDE and IKDE to obtain the average elapsed CPU times. Figure 5.3 shows the comparison results. For both KDE and IKDE, the median values of the computation times are shown with the upper and lower 5$^{\text{th}}$ percentile.

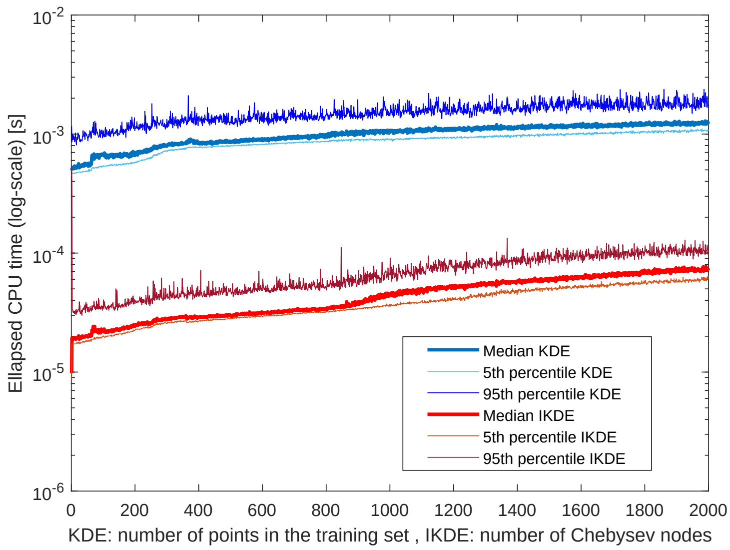


Figure 5.3: Comparison of the elapsed CPU time for calculating the posterior probabilities with KDE and with IKDE for one data point

It can be observed that in all cases the IKDE outperforms KDE, the CPU times being at least one order of magnitude smaller. It should be noted that IKDE is able to recreate the simulated distribution when the dataset contains 2000 samples with around 200 Chebyshev nodes.

Determining the order of the interpolated KDE

The superiority of IKDE over KDE with regards to computational time is due to the fact that the interpolation order N is much lower than the number of samples in training set n. For the comparison in Figure 5.3 a standard approach has been applied using *chebfun* (i.e. conversion of interpolating polynomial to the finite series in orthogonal Chebyshev basis and determining the interpolation order, for which series coefficients are indistinguishable under machine precision).

However, it is possible to deduce a formula for determining the order of the interpolated KDE with Gaussian kernels (Equation (5.23)). Firstly, the bound of M in Equation (5.30) has to be determined, i.e. the maximal value of KDE (Equation (5.21)) on the Bernstein ellipse. It can be observed that the KDE is bounded by the sum of identical components which add the most to the value at all points, so the sum and n can be dropped. The formula can be reduced to the problem of maximizing a Gaussian on a Bernstein ellipse, without loss of generality, focusing on Gaussians centered at zero. Then, the Chebyshev interpolant of KDE (Equation (5.21)) with kernel (Equation (5.23)) on the interval $[-1, 1]$ has an error that is not greater than ε if the interpolation order N is given by

$$N = \left\lceil \min_{\rho>1} \frac{\frac{1}{2h^2}\left(\frac{\rho^2-1}{2\rho}\right)^2 - \log\left(\frac{1}{4}\varepsilon\sqrt{2\pi}h(\rho-1)\right)}{\log \rho} \right\rceil \tag{5.37}$$

Its maximal value occurs at the imaginary axis, so the point of maximum is a semiminor axis b. The semiminor axis can be expresses as a function of ρ

$$b = \frac{\rho^2 - 1}{2\rho}$$

and applying it to Equation (5.30), taking its logarithm to obtain the bound, which is then rounded up to obtain an integer order. The minimization problem is well-posed for reasonable parameters (bandwidth and tolerance), but it is difficult to solve analytically, as it requires solving logarithmic equations. However, it is still a relatively unconstrained (there is a natural barrier at $\rho = 1$), differentiable, one-dimensional problem, so the solution can be found in several ways. It is important to note that this bound holds for the $[-1,\ 1]$ interval. In order to use it at different intervals, scaling is required, as described in the flow diagram in Figure 5.2. The bandwidth also needs to be rescaled when using it at at different intervals with standard deviation according to Equation (5.22). The bound will only hold for scaled bandwidths.

The method with Chebyshev series coefficients is slightly more efficient than the bound given by Equation 5.37. It requires multiple resampling of KDE and then computes the Fast Fourier Transform in order to obtain series coefficients. The calculation of the bound requires the solution of a one-dimensional optimization problem and results in a more conservative bound.

Feature-level fusion with IKDE

The $P(x_j|c_i)$ likelihood functions for each feature x_j in each fault class c_i may be calculated with IKDE using the same formulation as described for KDE with Equation 5.24. The training data are used to determine the α, β, $\mathbf{x}_{cheb}$, $\mathbf{y}_{cheb}$ and $\mathbf{w}_{cheb}$ values for each IKDE likelihood function and Equation 5.24 is computed as described by the right column of the flow diagram in Figure 5.3.

5.6. Bayesian decision-level fusion

Decision-level fusion takes the diagnostic results from the feature-level as inputs and fuses them into one final diagnostic result. If data type D_k predicted F_i and data type D_l predicted F_j, decision-level fusion provides a way to obtain an improved final prediction. The Bayesian formulation retains the transparency of the monitoring system. The likelihood functions on the decision-level represent the probabilities that a data type provides a correct prediction for F_i. The prior probabilities may be defined based on expert knowledge and the new evidence is the feature-level diagnosis result. A decision-level formulation is introduced. The method uses the predicted fault class from the feature-level diagnosis according to the maximum a posteriori test.

The prediction counts for each fault type are organized in an $M \times M$ confusion matrix G_{D_i} for each data or sensor type D_i where the rows represent the actual condition, the columns represent the diagnosed condition and the prediction counts by rows are divided by the total number of actual conditions for the fault type. The matrix elements can be interpreted as $P(F_i|F_j)$ conditional probabilities; given that the algorithm predicted F_j what is the probability that the actual fault condition is F_i? The $P(F_i|F_i)$ probabilities, located along the diagonal of the confusion matrix for each sensor type D_k, represent the probability that a data type diagnosed the corresponding fault correctly:

$$G_{D_k} = \begin{pmatrix} P(F_1|F_1) & P(F_1|F_2) & \cdots & P(F_1|F_M) \\ P(F_2|F_1) & P(F_2|F_2) & \cdots & P(F_2|F_M) \\ \vdots & \vdots & P(F_i|F_i) & \vdots \\ P(F_M|F_1) & P(F_M|F_2) & \cdots & P(F_M|F_M) \end{pmatrix} \tag{5.38}$$

If the fault class predicted by D_k is F_i and fault class predicted by D_M is F_j, then columns have to be selected in the following way from the corresponding confusion matrices:

$$G_{D_1,F_i} = \begin{bmatrix} P(F_1|F_i) \\ P(F_2|F_i) \\ \dots \\ P(F_M|F_i) \end{bmatrix}, ..., G_{D_M,F_j} = \begin{bmatrix} P(F_1|F_j) \\ P(F_2|F_j) \\ \dots \\ P(F_M|F_j) \end{bmatrix} \tag{5.39}$$

Finally, the predicted class label c_{pred} is determined:

$$c_{\text{pred}} = \arg\max_i \{P(c_i) \prod_{i=1,j=1}^{M,N} G_{D_i,F_j}\} \tag{5.40}$$

5.7. Training of the two-stage Bayesian framework

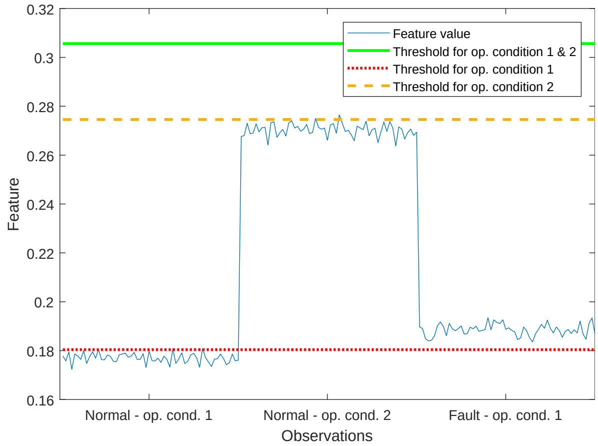
The two-stage Bayesian framework is a data-driven condition monitoring model composed of a set of Naive Bayes classifiers. To achieve a well-parametrized framework and good diagnostic performance with low false and missed alarm rates, it has to be trained, validated and tested using labelled historical condition monitoring data. In machine learning applications, three separate datasets are used to perform the parametrization of classifiers. The training set is used to fit the parameters of a classifier. A validation set may be used for tuning the hyper-parameters of a classifier. Finally, the test set is used to evaluate the performance of the trained classifier. The dataset used for the training and the way the dataset is split into training, validation and test sets can significantly influence the diagnostics performance. In this section, this issue is investigated with regards to how to avoid over-fitted and under-fitted models and how the operating condition dependency may be an issue when training condition monitoring models.

5.7.1. Avoiding over-fitted and under-fitted models

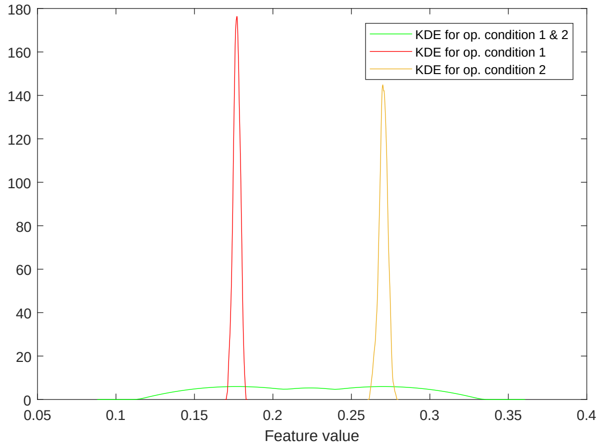
Over-fitted models contain more unknown parameters than can be justified by the data (Everitt and Skrondal, 2010). Such models tend to have reduced performance with unseen data compared to the performance achieved for training and validation sets. Under-fitted models are also unwanted, as they cannot properly capture the underlying data structure of the training data. Under-fitted models miss some parameters or terms that would appear in a correctly specified model (Everitt and Skrondal, 2010). To avoid over- and under-fitting the training and validation step is crucial for successful diagnosis performance.

The training of the two-stage Bayesian framework requires a training set to determine the likelihood functions of the feature-level fusion algorithms. The decision-level fusion of the monitoring results use the confusion matrices as likelihood functions. To determine each confusion matrix for each data type a validation set is required which is different from the training set. Finally, the test set is used to decide if the training of the algorithm has been successful and whether further parameter tuning is necessary for more accurate diagnostics results.

The 60-20-20% split of data has been described as rule-of-thumb split in the literature (Lever et al., 2016; Anifowose et al., 2017) for training, validation and test sets. Furthermore, by checking the diagno-



(a) Feature values and thresholds for two operating conditions and the threshold for both operating conditions



(b) The probability distributions calculated with KDE for the two operating conditions and for both operating conditions

Figure 5.4: Feature values for a normal condition and for a fault in two different operating conditions

sis accuracies of the trained model on the test set and comparing the results achieved using the training and validation set, it is possible to evaluate if the trained model is prone to over-fitting or under-fitting.

5.7.2. Operating condition dependency as an issue in diagnostics

It is often the case that features are dependent on operating conditions. This issue is illustrated in Figure 5.4 by means of a simulated example. Features are shown for a normal condition and for a fault under two operating conditions. In Figure 5.4(a), thresholds were calculated for two operating conditions for feature values calculated from data from the normal motor, described in Section 5.5.1. It can be observed that the threshold for operating condition 1 is much lower than the threshold for operating condition 2. However, it should be noted that even though the feature is reactive to operating condition changes, it provides valuable information about the health state. The fault results in increased feature values compared to the normal condition, which can only be detected if the thresholds are adjusted according to operating conditions. If the thresholds are calculated using simply combined data for operating condition 1 and 2, the result will be an even higher threshold if only operating condition 2 had been considered. Figure 5.4(b) shows the calculated likelihood functions for the three cases. If thresholds are calculated for each operating condition separately, then the fault will be detected, as it will always exceed the threshold of operating condition 1. Hence, if there is enough available training data labelled according to operating condition, operating condition models can be established by setting thresholds for each operating condition to solve this issue in the diagnostics framework.

Another difficulty caused by the dependence of features on operating condition is faced when the data is not labelled by the operating condition or there is data available from only one operating condition. In both of these cases, it is not possible to take into account and model the operating condition dependency of the features. Hence, new methods are necessary to tackle these challenges in an unsupervised way either by selecting features which are not dependent on operating conditions or transforming the features to a feature space, where the operating condition dependency is reduced.

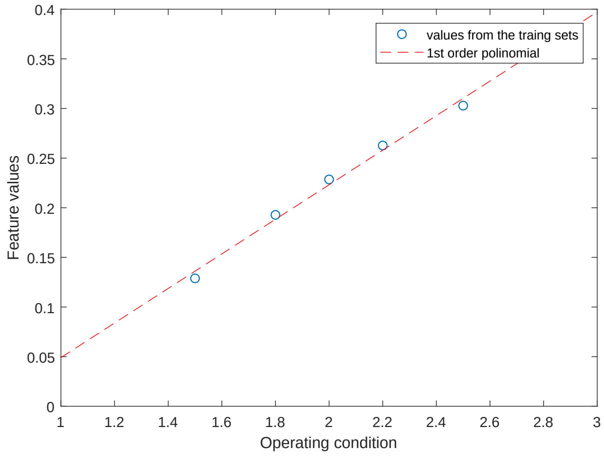


Figure 5.5: Calculating the load dependent threshold (Stief et al., 2017)

Operating condition model using polynomials

As highlighted in the previous example in Figure 5.4 feature values are often operating condition dependent and improved results can be achieved if the thresholds are also operating condition dependent. To solve this issue for the feature-level fusion method described in Section 5.5.1, an operating condition model using polynomials is proposed. If the training of the algorithm includes fitting of a first-order polynomial on the calculated thresholds for each feature for at least two different operating conditions among the normal training data then it is possible to determine thresholds for any given operating condition value. This method only works for cases when the operating condition is always known and the operating condition is determined by only one parameter. Figure 5.5 shows an example for such a case where the operating condition is dependent on only one parameter and the feature value is dependent on that one parameter. Fitting a first-order polynomial to the feature values allows the threshold to be identified for any operating condition.

Operating condition model using Principal Component Analysis

A typical challenge encountered when creating decision-level fusion algorithms is that there are often a large number of features relative to the number of observations. These features can be highly correlated, which ultimately can bias the results of a fault detection algorithm. A common method to reduce the correlation and the dimensionality of features is Principal Component Analysis (PCA) (Teti et al., 2010; Sutharssan et al., 2015).

For example in (Yang et al., 2006) the dimensionality of features extracted from several signals was reduced by PCA before applying genetic algorithms and an artificial neural network for classifying faults. It was found that the performance of the fault classifier was improved by adding PCA as a feature pre-processing step. In (Farajzadeh-Zanjani et al., 2017), several feature reduction and transformation methods including neighborhood component analysis, linear discriminant analysis (LDA), locally linear coordination and PCA were compared with maximally collapsing metric learning for fault diagnosis with particular focus given to the dimensionality reduction aspect. Feature reduction is also found in multi-stage frameworks, for example a recent work (Saucedo-Dorantes et al., 2017) applied PCA, LDA, a genetic algorithm and the Fisher score in a hybrid strategy to obtain a reduced and optimized feature set from vibration signals.

In this thesis, PCA was selected, as it is a well-established method for feature extraction, dimensionality reduction, data compression, and data visualization (Jolliffe, 2011). It is a common problem in data analysis that the features or attributes of the observation data are highly correlated. PCA transforms the correlated features to a linear space where the transformed features are uncorrelated and are ordered in a way that the first features retain most of the variation in the data. Singular Value Decomposition (SVD) or Eigenvalue Decomposition (EIG) are popular algorithms for performing PCA. In this thesis, SVD is considered, as it is numerically more robust when matrices are either singular or numerically very close to singular. If X is an $n \times m$ matrix with rank r, with n observations and m features, SVD is defined as

$$X = ULP^T \tag{5.41}$$

In Equation 5.41 U is an $n \times r$ orthonormal matrix, L is an $r \times r$ diagonal matrix and A is a $m \times r$ orthonormal matrix. SVD directly provides the required scores and loadings. UL is an $n \times r$ matrix, containing the transformed uncorrelated features in the principal component space, usually referenced as scores. P contains the principal components, sometimes called loadings (Jolliffe, 2011).

Fault indicators and features may be dependent on the operating conditions. By incorporating a multivariate statistical approach into the analysis, the correlations between operating conditions on the feature level can be accounted for. PCA is able to transform the correlated features to a linear space where the transformed features are less correlated and are ordered in a way that the first features retain most of the variation in the data. The feature-level fusion may suffer from bias due to the unevenly distributed data from various operating conditions resulting in reduced false and missed alarms.

PCA is proposed as a solution to reduce the correlations that are present in the extracted features and to reduce the influence of load conditions. It is also able to mitigate feature correlation which can bias the likelihood calculations. It is a linear method which yields a reduced and uncorrelated feature set. Instead of the original features, uncorrelated principal components are fused at the feature-level. The number of principal components considered for each data types is calculated using the validation set in a way that the performance of the algorithm is maximized whilst the false and missed alarm rates are reduced, using the detection accuracy as an optimization parameter. At the feature-level fusion stage, principal components of the features are fused with a NB classifier. The structure of the PCA - two-stage Bayesian fusion approach is shown in Figure 5.6.

The proposed algorithm is suited for condition monitoring problems where N data types provide data for the determination of the health state of the system. For training, the algorithm requires data that has been labelled with M fault conditions. If there is a test set available, the data has to be split into two separate data sets for training: the training set and the validation set. The training set is used for the training of the NB classifiers at the feature-level, while the validation set will produce the confusion matrices for the various data types at the decision-level.

Once the data are cleaned and selected features are extracted, the features are split by data type. At this stage, the training set takes the form of an $n \times m$ matrix, where n is the number of observations and m is the number of features. The μ_{x_i,D_j} means and σ_{x_i,D_j} standard deviations are calculated for each x_i feature and D_j data type. A normalization step transforms the features such that the means are zero and the standard deviations are one. PCA calculates the UL_{D_j} scores and P_{D_j} loadings for each data type. The scores, which might also be considered as the new features, are uncorrelated. The P_{D_j} loadings are calculated using the whole training set containing both healthy and faulty data.

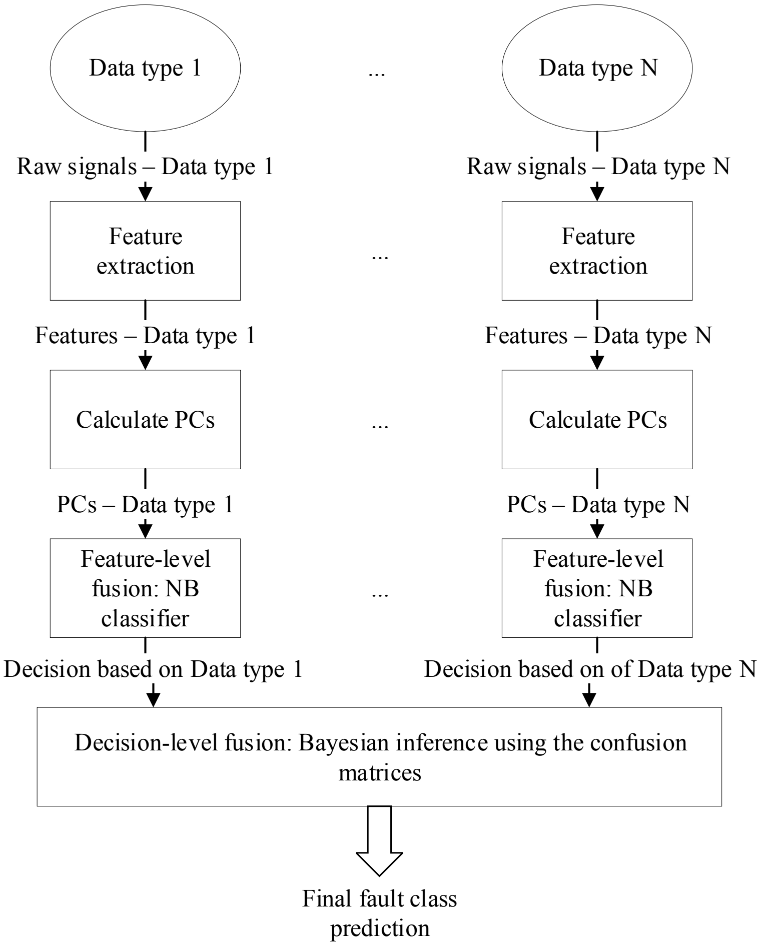


Figure 5.6: The flow diagram of the PCA - two stage Bayesian fusion approach (Stief et al., 2019c)

The validation set is used both to find the optimal number of principal components and to calculate the confusion matrices. The features in the validation set are normalized using μ_{x_i,D_j} and σ_{x_i,D_j}. The normalized features are transformed into the principal components space using the P_{D_j} loadings. To find the number of principal components for each D_j data type an iterative step is considered:

1. The first i principal components are used as features, calculating the posterior probabilities and class predictions for each observation in the validation set on the feature-level.
2. Count the correct predictions and save it for i.

Once the iteration has finished the value of i resulting in the highest number of correct predictions is chosen for the number of principal components used to calculate the predictions for each observation in the validation set.

5.8. Discussion and summary

In this chapter, Bayesian methods have been introduced in detail both for feature-level fusion and decision-level fusion. A two-stage Bayesian data fusion framework has been proposed, which is able to fuse data from disparate sources for condition monitoring purposes focusing on fault diagnostics. The two-stage Bayesian framework can be adapted to various monitoring problems depending on the distribution of the features, prior knowledge about the system and its fault modes.

Bayesian feature-level fusion with a Naive Bayes classifier can be implemented in several ways, depending on how the likelihood functions are calculated. In this chapter, four methods are described which are suitable for the feature-level task. The first method based on thresholds is the simplest. It requires minimal storage space only for storing the thresholds for each feature and the likelihoods of exceeding the thresholds in each fault class for each feature. Secondly, a Gaussian Naive Bayes classifier has been introduced, which is able to directly calculate the likelihood functions and the posterior probabilities for any new observation. The GNB classifier is also very computationally efficient and requires storage space only for the mean and variance values of each feature in each fault class. However, it is based on a Gaussian assumption among features, which is often not valid in real life applications. Therefore, a non-parametric KDE-based NB classifier was introduced. KDE is practical as no prior knowledge about the distribution of the monitored features is necessary. KDE is very computationally costly and also requires linear storage space proportional to the number of observations in the training set. Hence, IKDE has been proposed as a solution to use the KDE-based NB classifier in a less computationally expensive and faster way, which can also enable on-line implementations of the methods.

For the decision-level fusion, a NB formulation is proposed using a confusion matrix, which can be obtained during the validation of the feature-level fusion stage for each different data or sensor type. The method fuses the decisions from the feature-level fusion stage in the form of predicted class labels based on the maximum a posteriori probability.

The selection of the appropriate method for the feature-level fusion may also depend on whether the monitored system is employed in only one or many operating conditions. Two methods have been proposed for tackling the operating condition dependency of features. The first builds an operating condition dependent model and assumes that operating conditions only depend on one parameter and that there are data in the training set from a normal healthy state of the monitored system from at least two operating conditions. The second employs a PCA-based multivariate approach to transform the features to a less correlated and operating condition dependent feature space. This method may work with systems where the data is not labelled according to operating conditions.

To conclude, a newly proposed two-stage Bayesian framework has been described for fusing heterogeneous data, which aligns with the design considerations described in Section 2.5, such as transparency, modularity, and scalability. The proposed framework takes advantage of the properties of data fusion, as described in Section 3.4, such as improved accuracy and robustness. This framework can be easily adapted to several condition monitoring applications and to complex topologies of industrial plants. These applications are introduced in Chapter 7.

6. Case studies

This thesis investigates how heterogeneous data may be used for condition monitoring purposes. Data are often available from disparate sources, as described in Section 2.2. The fusion of disparate data has a number of advantages ranging from improved accuracy to improved robustness, as described in Section 3.4. A component of a system may be monitored by several sensors, while a complex process plant may be monitored using various monitoring systems. To develop and validate new diagnostics methods, case studies with heterogeneous data are necessary to show the effectiveness of the newly proposed algorithms. Two case studies are described in this chapter, one for component-level monitoring and one for plant-level monitoring to show that the newly proposed methods described in Chapter 4 and 5 are both applicable for component-level and plant-level monitoring. The two case studies are used later on in the thesis for data fusion and feature selection algorithm development and validation in Chapter 7.

The component level monitoring case study is based on induction motors monitored by several different sensors. The motors were operated under various loading and environmental conditions with and without induced faults. The plant level monitoring case study is based on a multiphase flow facility. The facility was instrumented with various sensors and various data acquisition systems. A heterogeneous dataset was recorded from the facility, which was working under various operating conditions with and without induced faults.

Both case studies contain heterogeneous data recorded under various health states and operating conditions. The case studies provide examples of how sensor and data fusion diagnostic methods may be applicable for both plant level and component level monitoring, therefore they are suitable for the validation of the algorithms developed in this thesis.

6.1. Component level monitoring: An induction motor case study

6.1.1. Introduction

Rotating machinery plays a significant role in industrial plants and is widely used in different applications and often operated under significant stress in harsh environmental conditions. Rotating machinery is often responsible for critical operations in the plants, where failures could result in a shut down of the production process. Ensuring reliable and safe operation of rotating machinery is, therefore, a key for high productivity in industrial plants.

The case study for component-level monitoring is based on an induction motor. Induction motors are commonly used rotating machines and their condition monitoring is a well-researched area with many standards and reports describing their possible faults and with a wide literature about existing diagnostic and prognostics methods. An IEEE motor survey (Albrecht et al., 1986) indicated that 44%

of failures in induction motors were due to bearing faults, 26% of the faults related to stator faults and 8% were designated as rotor, shaft or couplings related faults. A similar EPRI sponsored survey (IEEE, 1985) noted that 41% of induction motor failures related to bearing faults, 37% were due to stator faults and 10% were rotor related faults. The IEEE standard (IEEE 493-2007) summarizes and compares the results of the two studies in greater detail regarding motor sizes and failure rates. Induction motors can have electrical, mechanical and environmental-related faults (Karmakar et al., 2016), among which the most common are stator, bearing and rotor faults. These faults will result in various fault signatures being apparent in the mechanical, magnetic and electrical characteristics of the motors. One sensor might be applicable to diagnosing one fault mode, while not applicable for another fault mode, which might require an additional sensor in order to achieve good diagnostic results.

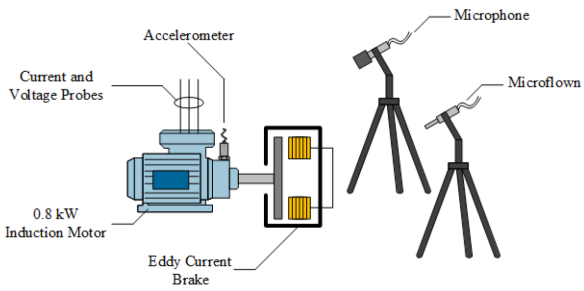
Specific induction motor faults can be diagnosed using various types of sensors (Nandi et al., 2005; Li and Mechefske, 2006; Mehrjou et al., 2011; Siddique et al., 2005), as some sensors are more suitable for detecting specific faults than others (Mehrjou et al., 2011; Siddique et al., 2005). The most commonly used sensor types for rotor and stator fault detection are acoustic, vibration and electric signals (Karmakar et al., 2016). Bearing faults lead to increased vibration and noise levels, hence acoustic and vibration sensors are the most sensitive for their detection. Broken rotor bar faults show specific patterns in electric signals, thus current signals are commonly used for their detection (Nandi et al., 2005). Nevertheless, acoustic signals can also be applicable for stator and rotor fault diagnostics of induction motors (Glowacz, 2019, 2018). As Li and Mechefske (2006) described, sensors which are responsive to a specific fault can provide information about other faults. Therefore condition monitoring systems fusing information from multiple sensors types can provide more accurate and comprehensive fault detection.

The induction motor case study was created for the development and testing of various condition monitoring techniques using various sensor data for the detection of several fault modes with various severities. It was conducted under various loading and environmental conditions to observe non-linear behaviors of fault pattern changes. The case study was used for the development and testing of feature selection methods and a sensor fusion framework. The measurement campaign of the case study was carried out by Maciej Sułowicz, Konrad Weinreb, Janusz Petryna, Arkadiusz Dziechciarz from Krakow University of Technology and Wojciech Batko, Maciej Kłaczyński, Jacek Wierzbicki, Tadeusz Wszołek, Jacek Frączek from AGH University of Science and Technology.

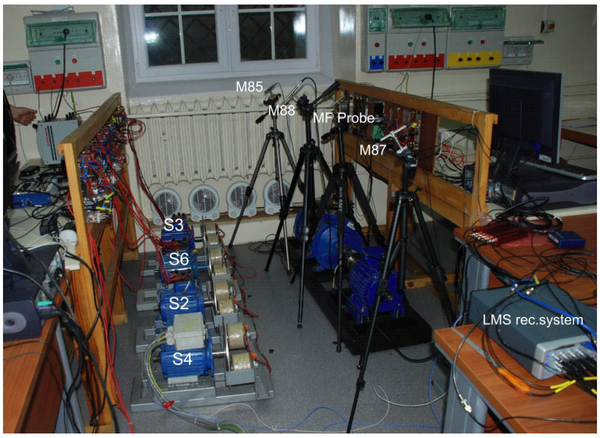
In this section, a detailed description is given about the case study, including the experimental set-up, fault modes, operating conditions, and data types. Based on this case study the following papers have been written Stief et al. (2017, 2018a, 2019c,a).

6.1.2. Experimental set up

A schematic of the experimental set-up for the case study is shown in Figure 6.1(a) (Stief et al., 2019c). Data were collected from three SZJKe 14a induction motors manufactured by TAMEL indicated as S4, S2, and S6 in Figure 6.1(b). The motor and bearing parameters are presented in Table 6.1 and 6.2. The three motors differed only in terms of health state, one motor was healthy, one had two broken rotor bars and one had an outer raceway fault in a bearing. It was also possible to seed stator faults into the healthy motor, as described in (Weinreb et al., 2016). An eddy current brake was used to load the motor. The measurements were conducted at steady-state operation under different loading and environmental conditions.



(a) Schematic of the measurement set-up (Stief et al., 2019c)



(b) Photograph of the measurement set-up (Batko et al., 2011)

Figure 6.1: The experimental set-up used in the induction motor case study

6.1.3. Faults and operating modes

For each fault case between three and five loading conditions were tested, resulting in stator currents of 68%, 81%, 90%, 100%, and 113% of the nominal current value. Measurements were recorded both with and without background noise generated by a separate shaker. Datasets were collected for eight

Table 6.1: Ratings of motor SZJKe 14a

Parameter	Value
Active power: P_N (kW)	0.8
Nominal voltage U_N (V)	380
Nominal current (A)	2.2
Nominal power factor cos phi_N (dimensionless)	0.74
Rotor speed (rpm)	1400
No load speed (rpm)	1497
Winding connection	Y
Number of pole pairs, p (-)	2
Nominal frequency (Hz)	50
Number of rotor bars (-)	22
Rotor inertia (kg m^2)	0.0025
Number of coils per phase	4
Number of turns in the coil	90

Table 6.2: Parameters of bearing SKF type 6304 ZZ CXSQ

Parameter	Value
Number of balls Nb (-)	7
Diameter of ball Bd (mm)	6
Pitch diameter of bearing Pd (mm)	32
Angle thrust alpha (rad)	0

different health conditions, denoted as F_0-F_7:

- F_0: Healthy motor
- F_1: Stator fault: Phase one bypassed in the first phase
- F_2: Stator fault: Phase one bypassed in half of the first phase
- F_3: Stator fault: Phase-phase short-circuit
- F_4: Stator fault: Phase-phase short-circuit with offset point
- F_5: Stator fault: Break of half of the phase one
- F_6: Rotor fault: Two broken rotor bars
- F_7: Bearing fault: Outer raceway defect

The tested motor was rewound in such a way that instead of coils for a given phase being directly connected to one another, the individual coils were connected to a switchboard allowing the winding configuration to be quickly changed. Furthermore, in six coils, special taps were created in order to allow different short circuits to be seeded. Such a configuration allows various stator faults to be seeded, as was investigated in (Weinreb et al., 2016) for the same SZJKe 14a induction motor. For F_1 and F_2 the first phase was bypassed by a 15Ω resistance causing a short circuit on the first phase winding. For F_3 and F_4 there was a short circuit of two stator phases in the taps connected in the middle of first coils by adding a 115 ohm resistance. In the case of F_5, part of the coil was not connected causing asymmetry in the winding, so that the current did not flow through a part of the winding. The two broken rotor bars (F_6) were located next to one another. The bearing fault (F_7) was caused by an incision through the outer ring of the bearing. The damage on the outer bearing was not made in a direction that is parallel to the rotation axis, rather at an angle.

6.1.4. Data types

Acoustic, electric and vibration signals were collected using 5 different sensor types, as shown in Figure 6.1(a). Three G.R.A.S. 46AE microphones were used to measure the sound pressure levels. A Model USP regular 3D Sound Intensity Microflown probe was also used to collect acoustic signals from the motors. The probe provided four measurement signals, three particle velocity signals in three orthogonal directions and a sound pressure signal. The vibration signals were measured by a 3-axis PCB ICP accelerometer Model No. 356B18 and a 1-axis PCB ICP accelerometer Model No. 353B32. The four vibration signals were measured in unit g [m/s^2]. The three-phase voltages were measured by LV 25-P voltage transducers providing signals directly for analysis of voltage characteristics. The motor currents were measured by LTS-6NP and LEM HY 5-P current transducers. The following signals were collected using a 16 channel LMS Scada Mobile System: 4 micro flown signals, 3 microphone signals, 2 current signals, 4 vibration signals, and 3 voltage signals. Data were collected with a 51.2 kHz sampling rate with 30 seconds of data being recorded for each configuration to capture a sufficiently long steady-state period for analysis. 58 datasets were obtained: one for each tested loading condition, both with and without additional background noise. The same background noise was applied over the tests.

The Microflown axis X probe measured an average 47.26 m/s particle velocity with no noise, while it measured an average 88.69 m/s particle velocity with noise for the healthy motor under nominal load.

6.1.5. Summary

The induction motor case study contains data recorded from several sensors mounted to induction motors operating under healthy and faulty conditions. The motor faults are different in nature and include stator faults, rotor faults and bearing faults. The motors were operated in steady state under various operating and environmental conditions. The data was recorded at a relatively high sampling rate, hence, feature extraction is relevant and feature selection methods may be tested and validated on this dataset. The dataset is suitable for testing and developing sensor fusion methods for component level condition monitoring. The sensors are expected to provide informative features for detecting various faults. Furthermore, the issue of changing loading and environmental conditions provides an additional condition monitoring challenge.

6.2. Plant level monitoring: A multiphase flow facility case study

6.2.1. Introduction

Reliable and fail-safe operations are key to achieve productive and profitable plants. Condition monitoring with early fault detection and diagnostics on the plant level is a research area with many challenges yet to be solved. Industrial processes, plants, and facilities are instrumented with numerous sensors and data acquisition systems for efficient plant level monitoring. Data generated and recorded from industrial plants are disparate, originating from heterogeneous sources, such as sensor measurements, alarm records, operation logs, maintenance records, videos and so on. Heterogeneous data offers a number of opportunities for improved reliability and robustness of monitoring algorithms (Lu et al., 2014), as complex system interactions can be modelled and taken into account, the strengths of one data type can be leveraged and its weaknesses mitigated (Hou and Bergmann, 2012).

One of the main challenges of developing industrial plant-wise monitoring systems is dealing with heterogeneous data which is now often available due to the recent improvements of technology in sensing, data storage, connectivity, and computing technologies (Tidriri et al., 2016). Developing such monitoring systems requires a benchmark dataset with heterogeneous data. Although there are a few available case studies and benchmark datasets, like the Tennessee Eastman process plant simulator (Ricker, 1995), an industrial-scale multiphase flow facility case study (Ruiz-Cárcel et al., 2015) or a carbon capture case study (Kachko et al., 2015), these only provide process data. Therefore there is a need for creating a benchmark data set, based on a plant level-monitoring case study with heterogeneous data to support the development and validation of advanced condition and process monitoring techniques.

This work aims to fill this gap by describing a case study based on an industrial scale multiphase flow facility with data available from several sources including process data, alarm data, and high frequency ultrasonic and pressure data, videos and an operation log. The case study was conducted under several operating conditions with induced faults commonly found in process plants (Datta and Sarkar, 2016), with different fault severities. The benchmark dataset also intends to promote the use of heterogeneous data for condition and process monitoring, therefore the dataset is publicly available online (Stief and Tan, 2018) with a detailed technical report (Tan and Stief, 2018).

The case study was conducted in the Process System Engineering Laboratory of Cranfield University together with Ruomu Tan. In this section, a detailed description is given of the case study including the experimental set-up, faults and operating conditions and data types. Based on this case study the following papers have been written: Stief et al. (2018c,b, 2019d,b).

6.2.2. Experimental set up

The case study was conducted on an industrial-scale, fully automated, high pressure, multiphase flow facility. The facility was designed for the investigation of the transportation, measurement, and control of multiphase flows comprised of water, air, and oil, which are found in offshore oil and gas process plants.

In this case study, only a two-phase flow comprised of water and air is studied. The schematic of the multiphase flow facility is shown in Figure 6.2 (Stief et al., 2018c). Water is supplied from the water tank $T100$ via a pump and air is supplied by an air compressor. Both air and flow rates are individually controlled and monitored by a SCADA system manipulating $FIC301$, $FIC302$, $FIC102$ and $FIC101$ automatic valves. The air and water are mixed in the mixing zone, then the two-phase flow is led to the horizontal section and then to the bottom of the riser unless the $U39$ manual valve is open. In this case, the two-phase flow is taken straight to the bottom of the riser. The two-phase flow is then directed to a

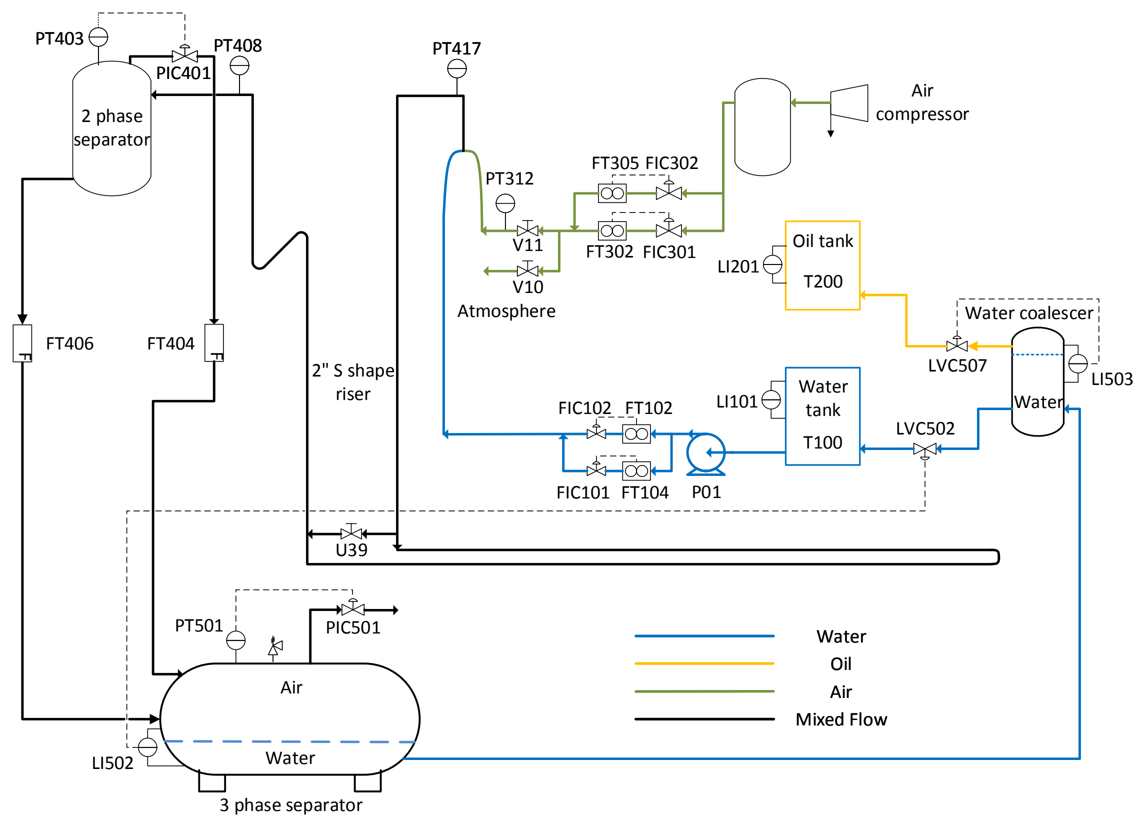


Figure 6.2: Schematic of the multiphase flow facility, (Stief et al., 2018c), © 2018 International Federation of Automatic Control. Reproduced with permission from the original publication in IFAC-PapersOnline, 51/18.

2″ vertical riser, which has an S-shape section in the middle. The mixed flow is separated on the riser top by the two-phase separator to liquid and gas phases. The gas phase is led to via $FT404$, while the liquid phase is led via $FT406$ to the three-phase separator, where water and air are separated again. The water is returned to the storage tank $T100$ via the water coalescer and the air is exhausted to the atmosphere via $PIC501$.

The facility is instrumented with various pressure, flow rate, temperature, density, and level sensors all connected to the SCADA system. Besides the flow control, the system has four additional control loops connected to the pressure in the two-phase separator ($PT403$), the level and pressure in the three-phase separator ($LI503$, $PT501$) and the level of the water coalescer ($LI503$). The facility is also instrumented with high-frequency pressure and ultrasonic flow measurements, as shown in Figure 6.3. The high-frequency pressure sensors are located along the pipelines from the mixing zone to the riser top while the ultrasonic sensor is located at the riser top. The pipeline has two transparent sections for the observation of the flow regime, one on the riser bottom and another on the riser top. Figure 6.4 shows several parts of the facility.

6.2.3. Faults and operating conditions

Operating conditions

The case study was conducted under twenty operating conditions, which were achieved by manipulating the air and water flow rates from the SCADA system, resulting in normal, stable flow and unstable flow called slugging. The air and water flow rates for the operating conditions are shown in Table 6.3.

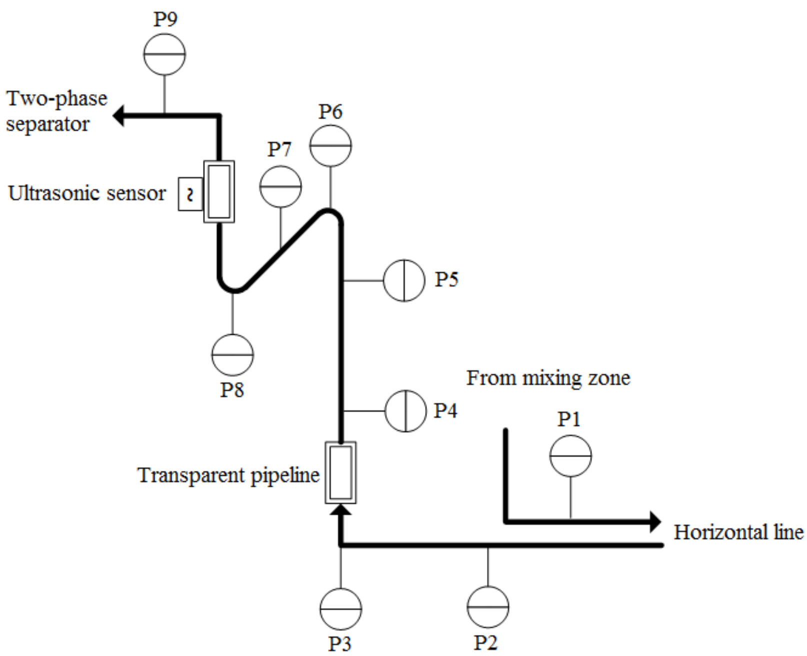
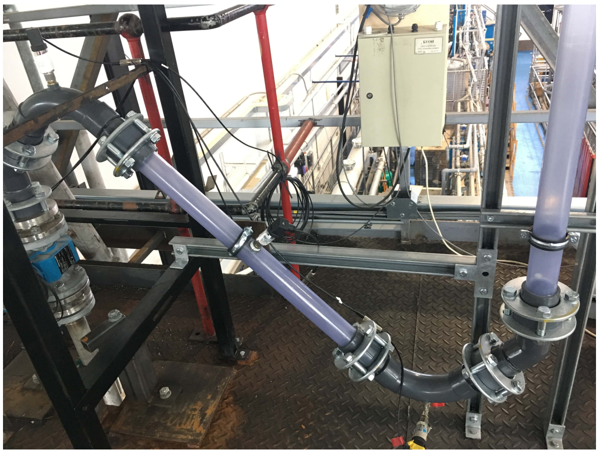


Figure 6.3: High frequency measurements, (Stief et al., 2018c), © 2018 International Federation of Automatic Control. Reproduced with permission from the original publication in IFAC-PapersOnline, 51/18.

(a) The three-phase separator

(b) Horizontal pipelines



(c) Overview of the S-shape riser

(d) The water storage tank

Figure 6.4: Photographs of the rig

Table 6.3: Operating conditions

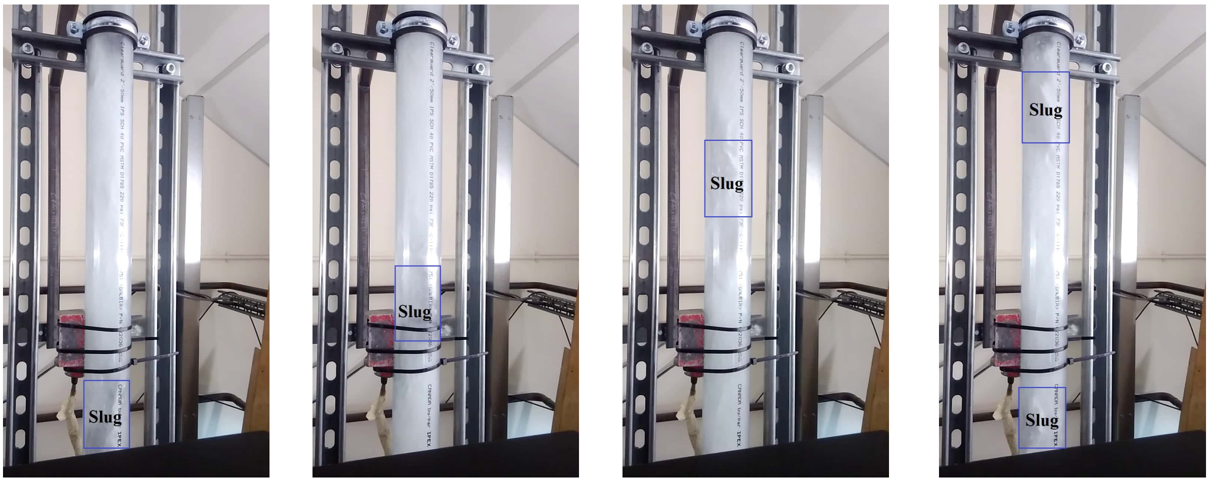
		Water flow rate (kg s^{-1})				
		0.1	0.5	1	2	3.5
Air flow rate (Sm^3 h^{-1})	20	slugging	slugging	slugging	slugging	normal
	50	slugging	slugging	slugging	normal	normal
	100	normal	normal	normal	normal	normal
	120	A: normal	-	-	-	-
	150	-	B: normal	-	-	-
	200	normal	normal	normal	-	-

Induced faults

Three faults were induced to the process for operating condition A and B (see Table 6.3). These faults were designed to simulate the most common process malfunctions, such as air leakage, air blockage, and diverted flow, which is incorrect operation of the system. Table 6.4 summarizes the faults and severities which were induced by manually opening or closing valves. All tests started from normal operating conditions after which the manual valves were gradually operated.

Table 6.4: Induced faults and fault severities

Induced fault	Operating condition	Valve	Severity by valve openings [degree °]		
			Mild	Moderate	Severe
Air leakage	A	V10	5	10	15
	B		5	10, 15	20, 25, 30, 40, 90
Air blockage	A	V11	80, 70, 60	50, 40	30 20, 10
	B		80, 70, 60	50, 40	30, 20, 10
Diverted flow	A	U39	5, 10, 15	20, 30	40, 50, 60
	B		10, 20	30, 40	45, 50, 60



(a) $t_0 = t$ (b) $t_1 = t + \Delta t$ (c) $t_2 = t + 2\Delta t$ (d) $t_3 = t + 3\Delta t$

Figure 6.5: An air slug moving upwards in the horizontal pipe (Operating condition: 20 Sm^3 h^{-1} air, 1 kg s^{-1} water flow rates)

Normal operating conditions

A representative normal dataset is essential for developing diagnostics algorithms, therefore thirteen normal datasets were recorded as shown in Table 6.3.

Slugging

Slugging occurs in multiphase flow risers due to relatively low gas and liquid flow rates, resulting in an unstable flow regime. It is an unwanted phenomenon, as it causes oscillations in the pressure, flow rate, and flow density through the riser (Jansen et al., 1996). Figure 6.5 shows the phenomena of slugging with four video frames, as it occurred at the top of the riser. Large air slugs are travelling within the two-phase flow causing pressure and flow rate oscillation in the pipes. Slugging was achieved by manipulating the input air and water flow rates until the fault pattern was observed: 7 slugging datasets were collected, the set points of which are shown in Table 6.3.

Air leakage

Air leakage was achieved by manually opening $V10$ according to valve openings shown in Table 6.4. As air leakage developed, the flow regime in the riser shifted from normal to slugging showing an interesting cyclic behaviour: first, there was normal flow regime, then the flow disappeared, then water appeared again with big air bubbles, then the cycle started again. When all of the air leaked out, the pressure dropped and continuous water only flow regime appeared.

Air blockage

Air blockage was achieved by manually closing $V11$ according to valve openings shown in Table 6.4. The flow regime remained continuous during the air blockage tests with mild slugging observed.

Diverted flow

Diverted flow was achieved by manually opening $U39$ according to valve openings shown in Table 6.4. The mixed flow was partially led straight to the riser and partially led into the horizontal pipeline. The diverted flow caused an only visible change in the flow regime at the bottom of the riser. There was no difference observed at the riser top.

6.2.4. Data types

Data was collected from disparate sources throughout the facility. The data types collected included process measurements, alarms, events and change logs, high-frequency measurements, an operation log, and video recordings. The heterogeneity of the dataset not only comes from the different data types but also from the different sampling rates (Low sampling vs. High sampling rates) and data availabilities (available during all operation vs. fixed length) and the way of data acquisition (Continuous, Triggered, On-demand), as introduced in Section 2.3.1. Table 6.5 summarizes the properties of the recorded data types.

Operation log

The operation log contained all manual changes made to the process, which were not logged by the SCADA system. It included operating conditions, start and end times of high-frequency measurements, the valve openings and observed flows. This data is useful for the synchronization of the different data types and the labelling of the dataset.

Process data

Process data were collected from the SCADA system sampled at 1 Hz from all of the connected sensors. The process variables are listed in Table 6.6 with their corresponding tags and units.

Table 6.5: Heterogeneous data recorded through the experiment

Data type	Sampling rate	Data acquisition	Availability
Operation log	-	On-demand	During all operation
Process variables	1 Hz	Continuous	During all operation
Alarm, event, change logs	-	Triggered	During all operation
Doppler ultrasonic sensor	10 kHz	On-demand	60 s windows
High frequency pressure sensors	5 kHz	On-demand	60 s windows
Videos	-	On-demand	30-60 s windows

Table 6.6: Process Variables

Tag	Process variable description	Unit
FT305/302	Input air flow rate	$Sm^3\ h^{-1}$
FT305-T	Input air temperature	°C
PT312	Air delivery pressure	bar(g)
FT102/104	Input water flow rate	$kg\ s^{-1}$
FT102-T	Input water temperature	°C
FT102-D	Input water density	$kg\ m^{-3}$
PT417	Pressure in the mixing zone	bar(g)
PT408	Pressure at the riser top	bar(g)
PT403	Pressure in the 2-phase separator	bar(g)
FT404	2-phase separator output air flow rate	$m^3\ h^{-1}$
FT406	2-phase separator output water flow rate	$kg\ s^{-1}$
PT501	Pressure in the 3-phase separator	bar(g)
PIC501	Air outlet valve 3-phase separator	(%)
LI502	Water level 3-phase separator	(%)
LI503	Water coalescer level	(%)
LVC502	Water coalescer outlet valve	(%)
LI101	Water tank level	m

Alarms, events, and changes data

Alarm, event and change data was sporadically logged during the whole operation from the SCADA system. Each log contained the time stamp, the corresponding sensor tag, the status and some additional information about the alarm/event/change type.

High-frequency Doppler ultrasonic flow data

The Doppler ultrasonic flow data was recorded at a 10 kHz sampling rate for 60 seconds for all of the tested scenarios in steady-state. Data acquisition was performed separately from the SCADA system in LabView. The ultrasonic data can be manually synchronized with the process data based on the operation log. A Continuous Wave Doppler Ultrasound non-invasive, clamp-on sensor was used for the measurement, which works on the basis of the Doppler Effect, namely that the frequency of an ultrasonic wave reflected from the scatterers of a moving medium is shifted in proportion to the velocity of the medium (Lynnworth, 2013). The sensor provided the Doppler frequency shift in the form of an output voltage signal.

High frequency pressure data

High-frequency pressure data was recorded at a 5 kHz sampling rate for 60-second windows during steady-state operation for all tested scenarios once the flow stabilized. Data were recorded in units of bar(g), they can be manually synchronized with the process data based on the operation log, as data acquisition was conducted separately from the SCADA system in LabView.

Videos

Videos were recorded on-demand for a period of 30-60 seconds during different tested scenarios to facilitate data labelling. For observing different operating conditions and air leakage the videos were recorded from the riser top. For diverted flow videos were recorded from the riser bottom.

6.2.5. Summary

The multiphase flow facility case study contains data recorded from heterogeneous sources from an experimental facility with and without induced faults. The induced faults were tested for various severities during which the process was operated at various operating conditions. The recorded data is heterogeneous as it was sampled differently, collected with various data acquisition system and also differs in terms of availability. The dataset contains high-frequency measurements from several sensors installed at various locations around the process, hence, feature extraction is relevant and feature and sensor selection methods may be tested and validated on this dataset. The dataset is suitable for testing and developing heterogeneous data fusion methods for plant level condition monitoring. Furthermore, the issue of changing operating conditions provides an additional condition monitoring challenge, similarly, as for the induction motor case study.

6.3. Summary of case studies

In this chapter, two case studies have been described both containing heterogeneous data recorded under various health states and operating conditions. The induction motor case study contains sensor measurements from different sensor types under changing external noise, changing loading conditions and induced faults. The multiphase flow facility case study contains disparate data from differently sampled sensor measurements, alarms and videos all recorded under changing operating conditions and induced faults. These case studies are suitable to validate data fusion and diagnostic methods. In the following part of the thesis, the previously described feature selection and data fusion methods are tested on these case studies to confirm their applicability for both plant level and component level monitoring applications.

7. Applications

In this chapter, the previously proposed feature selection methods (Section 4) and Bayesian data fusion methods (Section 5) are applied to the component-level monitoring case study of induction motors (Section 6.1) and to the plant-level monitoring case study of the multiphase flow facility (Section 6.2). Each method is shown why it is applicable for a certain monitoring problem, how it can be implemented on a typical real-life dataset and what are its advantages and limitations. This chapter builds on the results of the following publications: Stief et al. (2017, 2018c,b,a, 2019c,b,a). All of the results in this chapter were obtained after repeating the analyses described in the publications. In some cases the results have been improved and extended compared to published results. All analyses were conducted in Matlab®.

7.1. Feature and sensor selection for component level monitoring

In case of component level monitoring, a set of sensors provide measurements of various physical properties of the component. Feature extraction and signal processing in condition monitoring require domain knowledge about a system and its possible fault cases. Both are used to extract relevant information and also reduce the size of the dataset. Feature selection methods can evaluate feature relevancy and further refine the dataset to find the most sensitive features for various fault patterns and retain only the informative features.

In this section, the previously introduced feature selection method (Section 4.4), ReliefF is applied to evaluate the features extracted from an induction motor case study (Stief et al., 2018a). A standard feature set was extracted from the various sensors. The dataset contains data from eight different health states of an induction motor. Feature relevancy is calculated for each health state. The selected features are fed into a Bayesian binary classifier to calculate the most likely health state. The method provides insight into the relevance of features by sensor type and also by signal processing type. The newly proposed extension of ReliefF (Section 4.5.1) is also validated to reduce the correlation between the features. The evaluation of similarity among the selected features can help identify similar faults. The results emphasize the importance of domain knowledge in the proper design of features. Furthermore, by considering experimental data obtained for multiple loading and noise conditions, the feature selection method indicates features which are best suited for diagnosing specific faults, regardless of external conditions. Such information can support the creation of robust monitoring systems.

7.1.1. Feature extraction

As described previously in Section 6.1, the induction motor dataset contains 30-second steady state recordings from each sensor at a 51.2 kHz sampling frequency. Such measurements are available for

Table 7.1: Extracted feature from the motor dataset

Time and frequency domain features			
1	RMS	16	Amplitude at 50 Hz-rotation speed
2	Skewness	17	Amplitude at 50 Hz+rotation speed
3	Kurtosis	18	Ratio2X1X
4	Maximum Peak	19	Ratio3X1X
5	Crest factor	20	Envelope at 1X
6	Peak-to-peak	21	Envelope at 2X
7	Spectrum Area	22	Envelope at 3X
8	Frequency Center	23	Envelope at 50 Hz
9	Amplitude at 50 Hz	24	Envelope at 100 Hz
10	Amplitude at 100 Hz	25	Envelope at 50 Hz-2s
11	Amplitude at 1X	26	Envelope at 50 Hz+2s
12	Amplitude at 2X	27	Envelope at 50 Hz-rotation speed
13	Amplitude at 3X	28	Envelope at 50 Hz+rotation speed
14	Amplitude at 50 Hz-2s	29	Envelope of Ratio2X1X
15	Amplitude at 50 Hz+2s	30	Envelope of Ratio3X1X

each fault under several environmental and operating conditions leading to 58 recordings in total. The 30-second recordings are split into 0.5-second observation windows and a pre-defined feature set is extracted from all of the 16 signals from the five signal types for each observation window. In this way, 60 observations are obtained for each 30-second recording of the extracted features, in total 3480 observations for the 58 recordings. For each observation, the following time domain features were extracted: Root Mean Square (RMS), Skewness, Kurtosis, Maximum Peak, Peak-to-Peak, and Crest Factor. For each observation the following frequency domain features were extracted (both from the amplitude spectrum and from the envelope spectrum): Spectrum Area, Frequency Center, the amplitude of the components at the first two harmonics of the supply frequency (50, 100 Hz), the first three harmonics of the rotation speed (1X, 2X, 3X), the amplitude ratios (2X/1X, 3X/1X), and the amplitude at the side-bands of the supply frequency (50 Hz $\pm$ 2 $\times$ slip, 50 Hz $\pm$ rotation speed). For each of the 16 sensors, 30 features were extracted (Table 7.1), resulting in a total number of 480 features for each observation. These time and frequency domain features are standard for condition monitoring of induction motors (Jaramillo et al., 2017; Jardine et al., 2006; Nandi et al., 2005).

7.1.2. Implementation of methods

Once the features were extracted, the dataset was randomly split into a training set, containing 70% of the data and a test set, containing 30% of the data. The training data were relabelled for each fault case in such a way to fit a binary-class classification problem. The observations from one particular fault case and observations from the rest of the fault cases were labelled as "*Others*". The ReliefF algorithm was used to rank the features in the training set, as described in Section 4.4 by Equation 4.1

$$W(x) = W(x) - \frac{\text{diff}(x, R_i, H_j)}{m} + \frac{\text{diff}(x, R_i, M_j)}{m} \qquad \text{(4.1 revisited)}$$

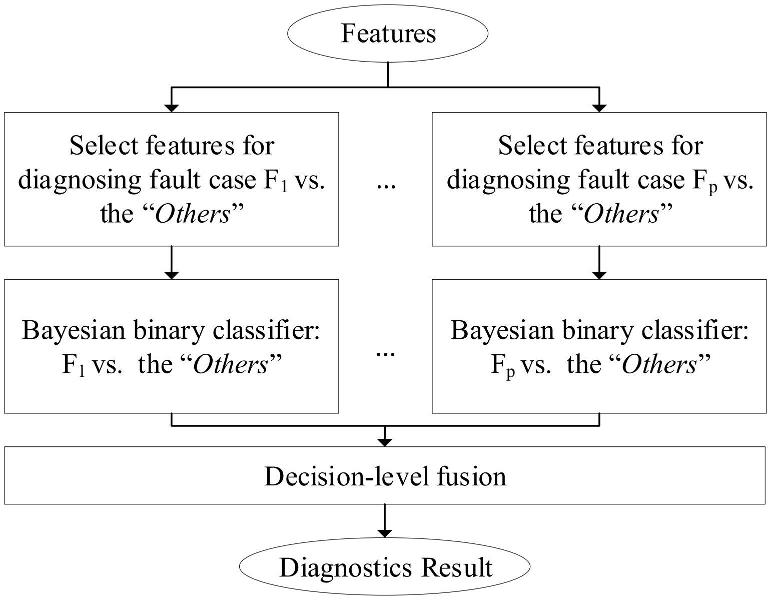


Figure 7.1: Structure of the Bayesian binary classifier

where R_i is the chosen observation, H_j are the nearest hits, M_j are the nearest misses, m is the number of iterations defined by the user and x denotes a feature. It was conducted for each fault case, altogether eight times to obtain the feature ranking for each fault case. Feature selection was conducted with the use of τ, as described by Equation 4.4,

$$\tau \leq \frac{1}{\sqrt{\upsilon \cdot m}} \tag{4.4 revisited}$$

where υ is the probability of accepting an irrelevant feature as relevant. In the analysis described here a value of 0.02 was chosen for τ. The ranking provided by ReliefF was recalculated using the newly proposed implementation of ReliefF with correlation removal, as described in Section 4.5.1 by Equation 4.6:

$$W_{\text{new}}(x_j) = W_{\text{old}}(x_j) \cdot \frac{\sum_{i=1}^{j-1} |(1 - \text{corr}(x_i, x_j))|}{j-1} \tag{4.6 revisited}$$

For the evaluation of the selected features, a Naive Bayes classifier was chosen to classify the observations into F_p fault classes. The feature-level fusion algorithm is formulated as described in Section 5.5.1 using thresholds. From the healthy training data, feature distributions are calculated for each feature, using Kernel Density Estimation (KDE). Thresholds are set symmetrically on the lower and upper end at 2.5% and 97.5% of the cumulative density functions. For any given observation, the probability that it belongs to fault category F_i given that a feature y_j crossed its associated threshold is calculated in the following way according to Equation 5.26:

$$P(F_i|\mathbf{y}) = \frac{P(\mathbf{y}|F_i)P(F_i)}{\sum_{j=1}^{m} P(y_j|F_i)} \tag{5.26 revisited}$$

The final prediction of the classifier is the fault class which has the highest posterior probability (index of the Maximum a posteriori class).

A set of binary NB classifiers have been used, as even in the case that there are more than two fault classes, there is an advantage associated with using binary classifiers: there is no need to use the same feature set for diagnosing the different fault classes. In the case of p faults, the fault patterns and the most relevant features can significantly vary amongst the different faults. After feature selection has been applied for each fault class using the training set, the first a features are selected to be subsequently used as inputs to the NB classifiers based on the relevancy index. Each classifier calculates the probability of a set of features belonging to a particular fault class, then the results of the p classifiers are fused on the decision-level to obtain the multi-class prediction using the following set of rules:

1. If only one classifier predicted a class label different than "*Others*", this class label is considered the final fault class.
2. If two or more classifiers predicted a class label different than "*Others*" then:
   - If there is a class label with a higher probability than other class labels, this class label is considered the final fault class.
   - If the two highest probabilities are equal, then the observation is classified as "*not known*".
3. If no classifier predicted a class label different than "*Others*", then the observation is classified as "*not known*".

The structure of the p binary classifiers is shown in Figure 7.1 for diagnosing p fault categories.

7.1.3. Results

For the purposes of comparison, the Bayesian binary classifier was tested with and without feature selection. The overall performance was measured by the percentage of correct classifications in the test set. In case of no feature selection and using all the 480 features the rate of correct classification was 96.06%. After ranking the features with ReliefF and selecting the features with higher weights than $\tau = 0.02$, the rate of correct classification was 96.99%, which is an improvement. Based on the ranking of ReliefF, the features were ranked again after correlation removal. The rate of correct classification with correlation removal was 97.45%, which represented a further improvement in the classification accuracy.

Figure 7.2 shows the 60 most relevant features by fault category after ReliefF. The table is color-coded according to sensor type and it provides information on which specific feature is extracted from the sensor in accordance with the features listed in Table 7.1.

The ranked features show patterns of correlation. In Figure 7.2, it may be observed that for example, the top eight most relevant features for distinguishing the healthy case F_0 are all from the Microflown sensor. These are the RMS, Maximum Peak, Peak-to-peak and Spectrum Area features from two Microflown signals. Table 7.2 shows the correlation matrix between these eight features ($x_1 - x_8$). The top three ($x_1 - x_3$) features, which are derived from the same Microflown signal are 100% correlated. Features $x_5 - x_8$, which are derived from the another Microflown signal are also 100% correlated. Even though these features may provide good class separation for diagnosing F_0, they introduce redundancy and correlation into the analysis that makes the naive conditional feature independence assumption of the naive Bayes classifier corrupted. This may affect the performance of the fault diagnosis due to overfitting, as described in Section 5.2.3.

Order of relevance of features

	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
1	1	21	10	9	9	22	8	8
2	4	22	8	1	1	10	7	9
3	6	4	1	4	4	4	8	11
4	7	10	4	6	6	10	1	16
5	1	3	6	10	10	6	11	16
6	4	10	22	7	8	8	8	11
7	6	21	4	10	1	10	16	17
8	7	6	1	8	4	1	11	17
9	10	29	21	10	6	4	6	12
10	22	8	10	21	4	6	8	12
11	8	10	11	8	8	11	8	8
12	21	8	9	1	5	1	4	12
13	10	1	1	4	10	21	20	17
14	2	9	7	6	5	16	7	7
15	1	1	4	21	11	15	7	9
16	5	1	6	3	7	20	16	5
17	7	4	1	27	3	12	8	6
18	21	6	8	21	7	15	7	17
19	4	4	7	22	20	14	7	17
20	6	6	4	11	16	2	7	28
21	1	4	6	8	21	7	4	9
22	7	8	16	7	8	12	8	11
23	11	7	7	8	8	14	8	9
24	10	17	7	10	1	13	6	17
25	20	8	8	16	10	15	4	28
26	4	6	8	22	10	16	3	15
27	8	1	7	1	22	12	16	16
28	10	11	7	7	4	14	5	22
29	16	7	7	10	5	9	8	17
30	7	10	7	20	6	1	6	18

Order of relevance of features

	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
31	10	2	6	4	2	7	15	9
32	7	9	20	21	10	17	21	28
33	8	1	7	10	5	8	7	12
34	7	7	21	7	7	7	14	27
35	20	5	10	7	1	10	26	17
36	7	7	11	5	4	4	4	10
37	9	16	27	1	6	6	6	20
38	8	22	10	4	7	7	19	1
39	1	7	2	6	7	1	5	8
40	20	20	22	7	7	7	25	18
41	6	1	22	7	11	4	28	4
42	7	21	16	11	7	6	10	6
43	11	7	7	5	7	5	13	3
44	21	9	5	7	10	13	5	13
45	10	7	10	7	7	16	10	28
46	4	7	10	7	7	7	1	10
47	3	7	27	10	5	7	19	10
48	16	10	4	2	17	7	18	2
49	6	27	3	21	21	7	9	9
50	7	11	8	20	8	8	1	1
51	22	10	4	3	8	8	4	5
52	7	1	21	8	16	7	4	14
53	27	9	9	3	9	11	6	7
54	22	7	8	9	6	17	16	9
55	3	5	4	16	22	8	11	3
56	27	18	7	6	8	13	5	17
57	16	8	16	22	8	7	6	13
58	9	9	16	16	8	17	1	8
59	4	8	11	3	16	11	23	11
60	27	4	6	11	21	7	7	1

Current
Voltage
Vibration
Microflown
Microphone

Figure 7.2: The 60 most relevant features by fault category

The newly proposed correlation removal method is applied to remove redundancy and correlation from the features. Figure 7.3 shows the 60 most relevant features by fault category with correlation removal. The table is color-coded according to sensor types.

7.1.4. Discussion

If considering the top 10 relevant features from the ranking provided by ReliefF for all fault cases in Figure 7.2, it may be observed that no features derived from voltage or microphone signals appear. Among the top 60 relevant features, the features derived from microphone and voltage signals appear the least frequently, indicating their limited importance. Features derived from current and vibration are the most dominant, while Microflown signals are of moderate importance. The voltage signal does not

Table 7.2: Correlation between features the first 8 features, which are the most relevant for diagnosing F_0 according to the ranking in Figure7.2

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	1.00	1.00	1.00	0.93	0.86	0.86	0.86	0.85
x_2	1.00	1.00	1.00	0.95	0.87	0.87	0.87	0.86
x_3	1.00	1.00	1.00	0.95	0.87	0.87	0.87	0.86
x_4	0.93	0.95	0.95	1.00	0.92	0.92	0.92	0.92
x_5	0.86	0.87	0.87	0.92	1.00	1.00	1.00	1.00
x_6	0.86	0.87	0.87	0.92	1.00	1.00	1.00	1.00
x_7	0.86	0.87	0.87	0.92	1.00	1.00	1.00	1.00
x_8	0.85	0.86	0.86	0.92	1.00	1.00	1.00	1.00

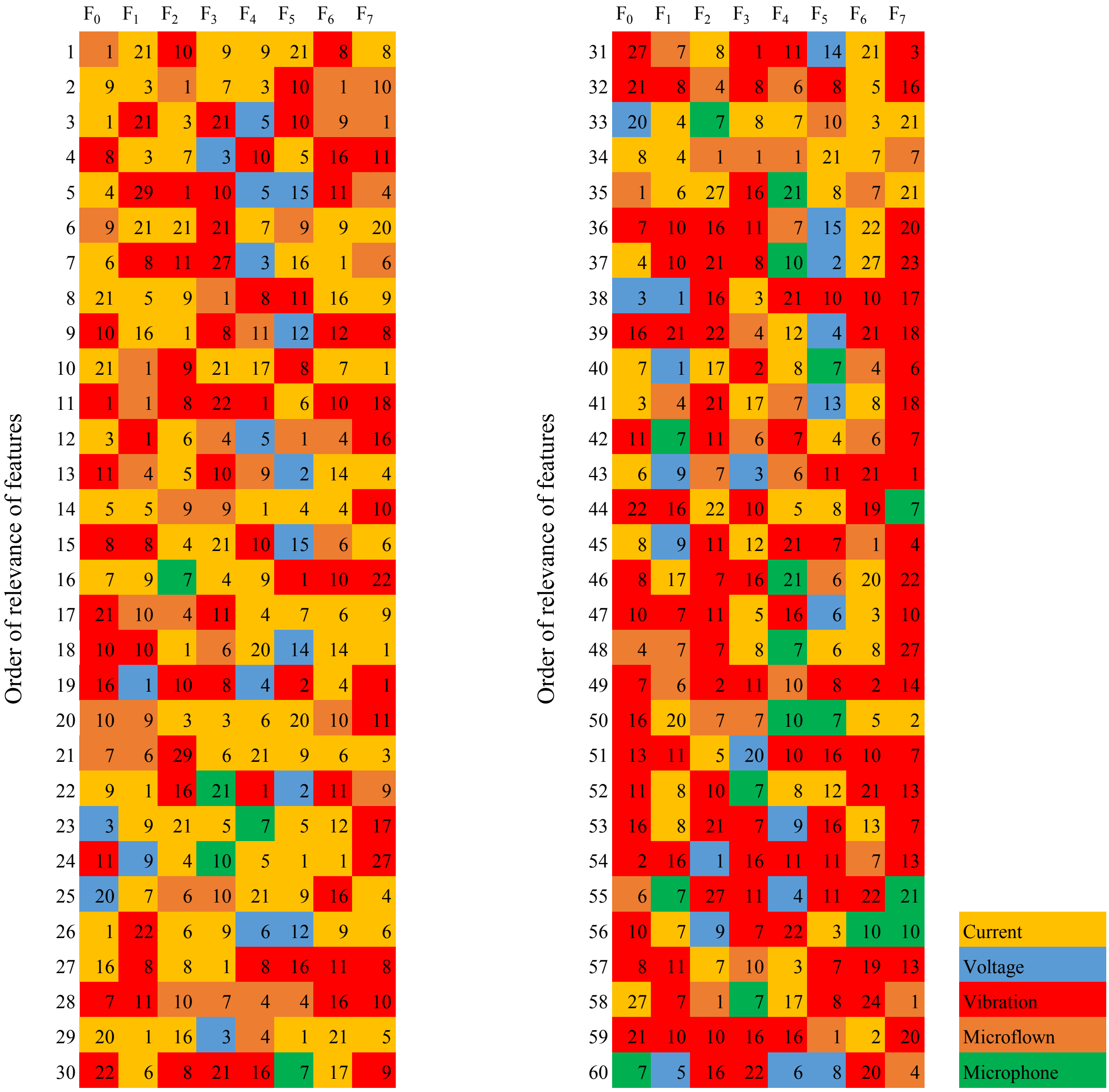


Figure 7.3: The 60 most relevant features by fault category using ReliefF with correlation removal

appear among the relevant features for F_2, F_6 and F_7, in these cases the voltage signal can be excluded from the diagnostics.

The results emphasize the importance of domain knowledge in induction motor condition monitoring, as many of the features identified as most relevant by the proposed algorithm align with those recommended in existing available literature related to diagnosing rotor, stator and bearing faults. For example, the most widely used approach for bearing fault diagnosis found in existing literature is the frequency domain analysis of measured vibrations (Tandon and Choudhury, 1999; Nandi et al., 2005), which is also reflected in the result. It may be observed in Figure 7.2 that in the case of bearing fault detection (F_7), frequency domain features extracted from the vibration signals are identified as the most relevant. This aligns with existing domain knowledge. The frequency analysis of currents, or motor current signature analysis, is a proven method for diagnosing electrical and magnetic field faults in induction motors (Li and Mechefske, 2006; Mehrjou et al., 2011; Nandi et al., 2005; Thomson, 2001). In the case of stator faults (F_2, F_3, F_4, and F_5) and rotor fault (F_6), it may be observed in Figure 7.2 that features based on frequency domain analysis of measured currents are the most relevant among the top 30 features.

The proposed method also highlights those features which hold less information about particular faults. For example, features based on Envelope at 50 Hz, Envelope at 50 Hz-2s, Envelope at 50 Hz+2s and Envelope of Ratio2X1X appear only once, while Envelope at 100 Hz, Envelope at 150 Hz and Envelope of Ratio3X1X do not appear at all. For the faults investigated in this section, features based on these metrics do not hold any significant value. For other types of fault not considered in this investigation, these indicators may have more value.

Although the microphone and Microflown acoustic signals are prone to background noise, their Spectrum Area feature appears for most of the fault categories as a relevant indicator. The Microflown signals also provide the top eight relevant features for the healthy motor. Whilst it may be observed that time domain and frequency domain features are relevant for all fault categories, in general, frequency domain features appear to be slightly more dominant. This is likely due to the fact that features based on frequency domain analysis are typically more discriminatory than time-domain features. Frequency analysis is used to extract components at specific frequencies associated with the dynamic signatures excited by a fault. By targeting specific frequencies of interest, the analysis is better able to discriminate between different fault modes and noise. This further illustrates the value of domain knowledge in selecting and constructing features. In this instance, applying domain expertise in the proper selection of particular frequencies of interest to be analyzed in the frequency domain. More advanced signal processing approaches, better suited for extracting specific aspects of fault signatures, may yield features with even greater relevance to monitoring specific fault modes. The finding that as a first consideration, domain knowledge should be applied in a feature selection problem in order to create the most informative features possible, agrees with classical literature in the field of feature selection Guyon and Elisseeff (2003).

It should also be noted that the analysis was performed on experimental data recorded both for different loading conditions and for different levels of environmental noise. The results, therefore, relate to features which are best suited for diagnosing particular faults, regardless of loading or noise conditions.

Another way to interpret and utilize the results of feature selection is to consider how many features provided the most relevant information for multiple fault categories. Table 7.3 gives a comparison of how many features appeared in the top 60 relevant features for any two fault categories. Fault F_0-F_4, which

Table 7.3: The similarity of the top 60 relevant features by fault category

	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
F_0		46	50	48	46	38	26	22
F_1	46		48	48	47	37	26	24
F_2	50	48		50	48	38	33	26
F_3	48	48	50		50	37	26	26
F_4	46	47	48	50		37	28	25
F_5	38	37	38	37	37		24	16
F_6	26	26	33	26	28	24		13
F_7	22	24	26	26	25	16	13	

include data recorded from the healthy motor and from four severities of stator fault have 46-50 shared features out of the top 60 relevant features. This results may be somewhat expected given the similarity of these fault types. F_6 (rotor fault: two broken rotor bars) and F_7 (bearing fault: outer raceway defect) are the most different faults. This is reflected in Table 3 as these two fault modes share only 13 common features among the top 60 relevant features for their respective fault categories.

It may be observed from the ranking presented in Figure 7.2, that many features with similar rankings are from the same sensor type. For example, RMS, Maximum Peak and Peak-to-peak features from the Microflown signal appear after each other in case of fault cases F_0-F_5. Similarly, in the case of F_6 the 10 most relevant features are all from the vibration sensors. This phenomenon is due to the correlation between the features.

When observing the ranking obtained by correlation removal in Figure 7.3, this phenomena no longer dominates the results. The current and vibration signals are the most relevant for the analysis. However, features from the voltage signal in case of stator faults F_3-F_5 also appear among the 10 most important features. Microphone signals are still the least relevant for the analysis. The relevance of Microflown features is reduced after correlation removal.

The standard implementation of ReliefF may help the user of the condition monitoring system to observe the relevance of each feature. However, ReliefF with correlation removal is able to produce uncorrelated feature sets, which may achieve less biased and more accurate fault detection and diagnosis results.

Finally, it has to be noted that these results were achieved by randomly splitting the observations from each fault category and loading condition into a training set and test set. In the case of using new data recorded from tests with different environmental or loading conditions, there is no guarantee that the selected features will still be the most relevant.

7.1.5. Summary

This section has investigated new applications of the ReliefF feature selection approach for condition monitoring of induction motors. The multi-class classification problem with several fault categories was decomposed into a two-class classification problems in order to use the best subset of features for diagnosing different bearing, stator and rotor faults. The method proved to be suitable for identifying which signal processing techniques and sensors are the most relevant for monitoring a certain fault class. The similarities between relevant features for fault categories can help in identifying similar faults.

By considering multiple loading and noise conditions, the obtained results indicate features which are best suited for diagnosing specific faults, regardless of loading or noise conditions. This information may help in designing more robust monitoring systems. The results obtained also reinforce the importance of domain knowledge in the selection of appropriate sensors and signal processing approaches which are able to discriminate both different faults from one another and from ambient noise.

The newly proposed method proved to be useful for removing correlation between features and the results showed that, without correlation removal, ReliefF itself would include many highly correlated features in the selection, which may bias the classifier. ReliefF with correlation removal also achieved improved fault diagnosis accuracy compared to the standard implementation of ReliefF.

The limitation of the described method for condition monitoring applications is that in the case of using data with different environmental or loading conditions than used in the training set, there is no guarantee that the selected features are still the most relevant or that the relevancy thresholds still hold. Therefore, it is advised to use it as a preliminary feature evaluation method in order to have a better understanding of the features and the faults.

7.2. Feature and sensor selection for plant level monitoring

Plant level monitoring systems may collect a set of measurements from distributed sensors. The most suitable sensors and subsets of features might differ for various monitoring problems. In this section, this issue is investigated on the multiphase flow facility case study described in Section 6.2 using the high-frequency pressure and ultrasonic measurements, which provide abundant information for flow regime monitoring and fault detection (Stief et al., 2019b). Power Spectral Density and the Discrete Wavelet Transform are used for feature extraction. ReliefF, which is a K-nearest neighbors-based feature selection filter introduced in Chapter 4, is used to rank the features for the various monitoring problems, such as fault detection and fault severity diagnosis. The ReliefF-based hybrid approach for feature selection proposed in Section 4.5.2 is used to select the features used for monitoring. The analysed dataset contains data from different operating conditions and induced faults with different severities. The optimal subset of features is explored for fault detection and fault severity diagnosis by applying a Naive Bayes classifier for feature-level fusion using Kernel Density estimate, as described in Section 5.5.3.

In this section, the effects of sensor failures are also investigated using a multi-class Naive Bayes classifier without feature selection to further confirm the relevance of feature selection for plant-level condition monitoring applications. Ensuring that fault detection and diagnosis (FDD) based on sensor fusion is robust against sensor failures is a known challenge (Heng et al., 2009; Jardine et al., 2006). The system architecture is crucial for successful monitoring (Esteban et al., 2005). Monitoring systems, which rely on many sensors, are more vulnerable to sensor failures than those which use fewer sensors. However, the observability of fault modes puts a constraint on the minimal number of sensors necessary for the successful diagnosis. A possible solution is to decompose the system into several sub-systems for different monitoring problems. Hence, if a sensor which is not necessary for all of the sub-systems fails, only part of the monitoring solution will potentially be compromised. As a result, modular and scalable approaches are preferable, where each monitoring subsystem uses as few sensors as possible, in order to ensure the overall monitoring solution is robust and easy to maintain.

7.2.1. Feature extraction

As periodic phenomena are common in two-phase flows, frequency domain and time-frequency domain methods provide a good means of analysing oscillation periods in high-frequency flow and pressure data (Shang et al., 2004). The most common feature extraction methods described in the literature for two-phase flow monitoring are based on Power Spectral Density Estimate (PSD) and the Discrete Wavelet Transform (DWT) (Xie et al., 2004).

Power Spectral Density

Santoso (2012) used Power Spectral Density (PSD) features, such as average power and variance of power in different frequency bands, from differential pressure data of horizontal gas-liquid flow, proving that these features were suitable for identifying the different flow patterns with high accuracy. Sun and Zhang (2007) both extracted spectral variance from vortex flow meter signals and calculated the average power in certain frequency bands in order to successfully identify flow patterns. Abbagoni and Yeung (2016) described that the above mentioned PSD features are also suitable for flow regime detection when using an ultrasonic Doppler sensor. The mean spectral power and spectrum variance can be calculated in the following way:

$$\bar{f} = \frac{\sum_i f_i P_x(f_i)}{\sum_i P_x(f_i)} \tag{7.1}$$

$$\sigma_f^2 = \frac{\sum_i (f_i - \bar{f})^2 P_x(f_i)}{\sum_i P_x(f_i)} \tag{7.2}$$

where $\bar{f}$ is the mean spectral power and P_x is the PSD function. The features extracted from the ultrasonic and pressure signals are summarized in Table 7.4 and Table 7.5. The Welch method is used to calculate the PSD spectrum (Welch (1967)).

Figure 7.4 shows the PSD spectrum for the ultrasonic signal for a developing leakage case. It can be observed that for severe leakage (V1=20, 25°) the pattern is visibly different and for V1=15°the spectrum in the lower frequency ranges is distinguishable from the rest, however, for milder leakage, the signals show similar characteristics as in the case of the normal condition.

Table 7.4: PSD features from the ultrasonic signal

Feature type	Frequency range (Hz)	Feature name
Average power	0-120	US^{B1}
	120-240	US^{B2}
	240-360	US^{B3}
	360-480	US^{B4}
	480-600	US^{B5}
Mean spectral power	0-2500	US^{msp}
Variance of spectral power	0-2500	US^{vsp}

Table 7.5: PSD features from the pressure signals. In the notation index x refers to the number of pressure sensor.

Feature type	Frequency range (Hz)	Feature name
Average power	0-240	Px^{B1}
	240-1000	Px^{B2}
	1000-1500	Px^{B3}
	1500-2500	Px^{B4}
Mean spectral power	0-2500	Px^{msp}
Variance of spectral power	0-2500	Px^{vsp}

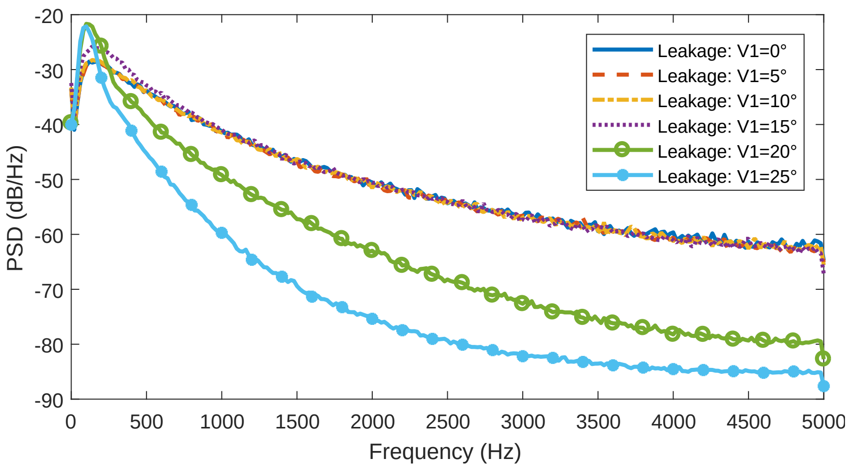


Figure 7.4: The PSD spectrum of the ultrasonic signal for a developing leakage(Operating condition B)

Discrete Wavelet Transform

Wavelet analysis is a powerful tool for analysing complex non-linear signals such as pressure and flow signals present in two-phase flows (Shang et al. (2004)). It has been successfully applied in two-phase flow monitoring to determine the flow regime and bubble sizes (Seleghim Jr and Milioli (2001)). The Discrete Wavelet Transform (DWT) convolves the original signal with a low-pass filter and then subsequently with a high pass filter. The low-pass filter outputs the approximation coefficients, while the high-pass filter outputs the detail coefficients after down-sampling. The transformation can be repeated maximum $\log_2 N$ times if the signal has N data points, producing approximations and detail levels in different frequency bands. To extract representative features from wavelet coefficients, standard statistical features can be applied, such as mean, variance, minimum value and maximum value (Abbagoni and Yeung (2016)). The extracted DWT signals are shown in Table 7.6 and Table 7.7. The db2 Daubechies wavelet (Malik and Verma (2012)) was used to compute the wavelet coefficients.

Table 7.6: DWT features for the Ultrasonic signal. Feature types var, mean, min and max are used in the superscript later on.

Details wavelet coefficients	Frequency range (Hz)	Feature name	Features
Level 1	2500-5000	US^{L1}	var, mean, min, max
Level 2	1250-2500	US^{L2}	
Level 3	312-625	US^{L3}	
Level 4	156-312	US^{L4}	
Level 5	78-156	US^{L5}	
Level 6	39-78	US^{L6}	
Level 7	19-39	US^{L7}	

Table 7.7: DWT features for the Pressure signals. Index x refers to the number of pressure sensor in the notation. Feature types var, mean, min and max are used in the superscript later on.

Details wavelet coefficients	Frequency range (Hz)	Feature name	Features
Level 1	1250-2500	Px^{L1}	var, mean, min, max
Level 2	312-625	Px^{L2}	
Level 3	156-312	Px^{L3}	
Level 4	78-156	Px^{L4}	
Level 5	39-78	Px^{L5}	
Level 6	19-39	Px^{L6}	
Level 7	9-19	Px^{L7}	

7.2.2. Implementation of the method

The dataset used for this analysis is described in Section 6.2. Only the high-frequency pressure and ultrasonic flow measurements were considered. The analysis used 49 recordings from operating condition A and B for normal conditions and for the three induced faults, as shown in Table 6.3. The 49 recordings from the nine high frequency pressure and one ultrasonic flow sensors were labelled according to fault class (*Normal*, *Blockage*, *Leakage*, *Diverted* flow). Each 60-second recording was divided into 1-second windows and features were extracted from each measurement window. Once the 341 features were extracted the data were randomly divided into 60% training set, 20% validation set, and 20 % test set. The observations were relabelled to fit binary-class classification problems: for each fault case the data was labelled in such a way that all the other conditions were labelled as *Others*, for example, *Normal* versus *Others*. Once the data was prepared, the ReliefF algorithm ranked the features in the training set by relevance for each fault class. ReliefF also ranked the features for the multi-class classification problem with all the fault labels.

Feature selection from the ranked feature set was conducted with the hybrid filter-wrapper approach, as described in Section 4.5.2. The classifier selected for the evaluation of the features was the Bayesian feature-level fusion with KDE described in Section 5.5.3. KDE was used to obtain the posterior probabilities in the Naive Bayes classifier framework for each observation belonging to a fault case. The classifier was trained on the training set and tested on the validation set. Initially, only the most relevant features were included in the model. The training and validation steps were subsequently repeated multiple times, with each successive model incorporating one additional feature, ordered in terms of ranking, until all of the features were included in the analysis. The performance of the classifier was determined by its classification accuracy. The feature set, which achieved the highest classification accuracy was selected for Bayesian feature-level fusion model, which was retrained using the training set with the selected features. The selected features from the test set were used as inputs to the Bayesian feature-level fusion model, which obtained the final fault diagnosis results.

The faulty measurements were also labelled according to fault severity (*Mild*, *Moderate*, *Severe*). For each fault severity class within a fault class, ReliefF ranked the features by relevance to obtain the best class separation for severity. The validation set was used similarly as before to select the features with the hybrid approach. The test set was used to obtain the final fault severity diagnosis results.

7.2.3. Results

This section presents the results of the implemented feature selection and fault diagnosis framework for fault diagnosis and for fault severity diagnosis. For both cases, the ranking from ReliefF is provided in a table format to give an insight into which features, which sensors and which signal processing methods proved to be the most relevant. Then, the feature selection results of the hybrid filter-wrapper approach are shown for the validation set, followed by the number of selected features and fault diagnosis accuracies.

Feature ranking for fault diagnosis

The ReliefF algorithm provided feature ranking for the binary-class classification problems of the induced faults and the normal case (*Normal*, *Blockage*, *Leakage*, *Diverted* flow). ReliefF was also applied for the multi-class fault diagnosis for comparison (*All*). The 20 most relevant features for each case are presented in Table 7.8. P5 pressure sensor provided the most relevant features for all three fault classes. Its high ranking, along with the other P6, P7 and P8 sensors on the riser was somewhat expected, as the flow regime in this section was highly dependent on the induced faults, especially in the case of leakage. The ultrasonic sensor was present only once among the 20 most relevant features, which indicated that the pressure sensors were most sensitive to the induced faults. P1, P2, P3, and P4 were also less relevant. They were located along the horizontal pipeline, which was less affected by the induced faults than the riser. This might have been due to the location of the ultrasonic sensor at the very top of the riser being less sensitive for pressure and flow fluctuations as the other sensors located along the riser. Based on the feature ranking results in Table 7.8, the DWT features were more relevant than the PSD features. The variance of the DWT features was more relevant than the minimum, maximum or mean value of the wavelet coefficients.

The results of the hybrid feature selection with Bayesian feature-level fusion are presented in Figure 7.5. This analysis used the validation set for the relabelled binary classification and for the multi-class classification. The number of the features included in the analysis are shown on the x-axis, while the achieved classification accuracy is shown on the y-axis. For all cases, the best classification accuracy was not for the case when all features were included. This result indicates that feature selection can lead to an improvement in diagnostic accuracy. It may be observed that for *Blockage* and *Diverted* flow the inclusion of more features did not significantly decrease the accuracy. For the rest of the cases, using

Table 7.8: Feature ranking for fault diagnosis. The notation of the features are used as indicated in Tables 7.4, 7.5, 7.6, 7.7.

no.	All	Normal	Blockage	Leakage	Diverted
1	$P5^{L2}_{var}$	$P5^{B2}$	$P5^{L2}_{var}$	$P5^{B3}$	$P5^{B2}$
2	$P5^{L1}_{var}$	$P5^{B3}$	$P5^{L1}_{var}$	$P5^{B2}$	$P5^{L1}_{var}$
3	$P5^{B3}$	$P5^{L2}_{var}$	$P5^{L6}_{var}$	$P5^{L3}_{var}$	$P5^{L2}_{var}$
4	$P5^{B2}$	$P6^{B2}$	$P5^{B3}$	$P5^{L2}_{var}$	$P5^{B3}$
5	$P5^{L6}_{var}$	$P5^{L1}_{var}$	$P5^{L4}_{var}$	$P5^{L1}_{var}$	$P5^{L4}_{var}$
6	$P5^{L4}_{var}$	$P8^{B2}$	$P6^{L1}_{var}$	$P5^{L4}_{var}$	$P5^{L6}_{var}$
7	$P5^{L3}_{var}$	$P6^{L2}_{var}$	$P5^{L3}_{var}$	$P6^{B3}$	$P6^{L1}_{var}$
8	$P6^{L1}_{var}$	$P8^{L1}_{var}$	$P8^{L4}_{var}$	$P6^{L1}_{var}$	$P6^{L2}_{var}$
9	$P6^{L2}_{var}$	$P6^{L1}_{var}$	$P6^{L2}_{var}$	$P5^{L6}_{var}$	$P5^{L5}_{var}$
10	$P8^{L4}_{var}$	$P8^{L3}_{var}$	$P5^{B2}$	$P8^{L1}_{var}$	$P8^{L1}_{var}$
11	$P8^{L1}_{var}$	$P7^{B2}$	$P5^{L6}_{max}$	$P6^{L2}_{var}$	$P6^{B4}$
12	$P5^{L5}_{var}$	$P8^{L4}_{var}$	$P8^{L3}_{var}$	$P8^{B2}$	$P5^{L3}_{var}$
13	$P8^{L3}_{var}$	$P6^{L3}_{var}$	$P5^{L5}_{var}$	$P8^{L4}_{var}$	$P5^{B4}$
14	$P6^{B3}$	$P7^{L3}_{var}$	$P7^{L3}_{var}$	$P8^{L3}_{var}$	$P7^{L1}_{var}$
15	$P8^{B2}$	$P5^{L6}_{var}$	$P6^{L3}_{var}$	$P8^{L2}_{var}$	$P8^{L4}_{var}$
16	$P7^{L3}_{var}$	$P3^{B2}$	$P8^{L1}_{var}$	$P7^{L1}_{var}$	$P8^{L3}_{var}$
17	$P6^{B4}$	$P7^{L1}_{var}$	$P8^{B2}$	$P8^{B4}$	$P8^{B2}$
18	$P6^{L3}_{var}$	$P8^{L2}_{var}$	$P5^{L6}_{min}$	$P8^{msp}$	$P6^{B3}$
19	$P7^{L1}_{var}$	US^{B1}	$P3^{L3}_{var}$	$P7^{L4}_{var}$	$P8^{B4}$
20	$P5^{L6}_{max}$	$P5^{L3}_{var}$	$P1^{L3}_{var}$	$P6^{msp}$	$P7^{L3}_{var}$
	DWT		PSD		

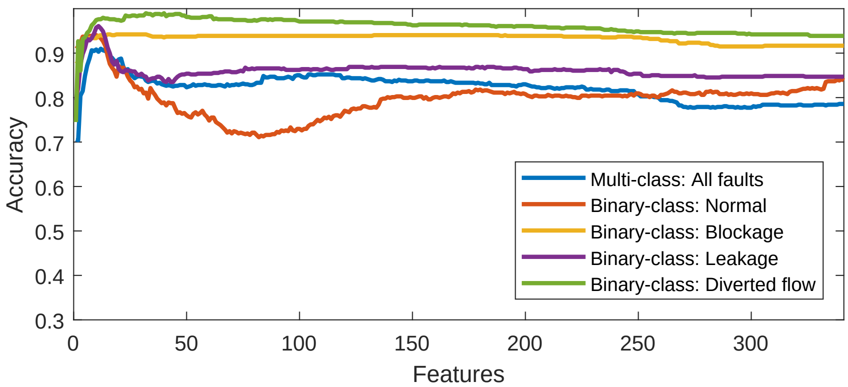


Figure 7.5: Feature selection with a hybrid approach using Bayesian feature-level fusion with KDE to evaluate the performance of the ranked features for fault diagnosis

more features caused a decrease in the diagnosis accuracy. This behaviour was most visible for detecting the *Normal* conditions from the rest of the fault cases. When using around 10 features, the accuracy remained above 90%. When more features were added, the accuracy decreased to 71%.

Table 7.9 summarizes the number of features selected with the hybrid approach and the classification accuracy achieved with the selected features on the test set for fault diagnosis. The multi-class fault

Table 7.9: Fault diagnosis: selected number of features and fault severity diagnosis accuracies

	All	Normal	Blockage	Leakage	Diverted
Number of selected features	12	10	15	11	32
Accuracy(%)	91.16	92.52	93.71	95.92	98.3

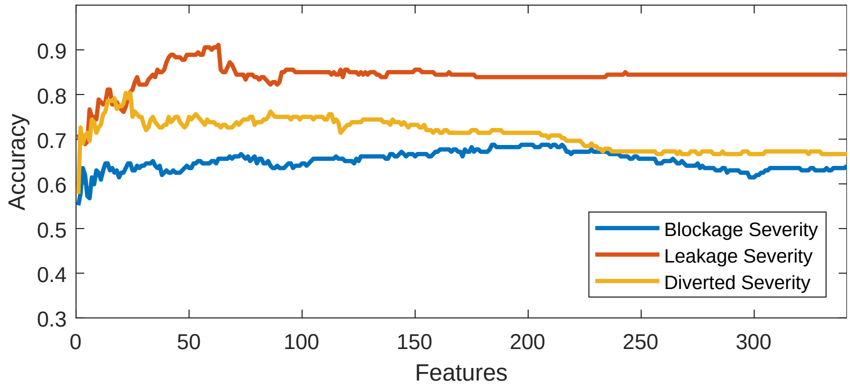


Figure 7.6: Feature selection with a hybrid approach using Bayesian feature-level fusion with KDE to evaluate the performance of the ranked features for diagnosing fault severities

diagnosis accuracy was 91.16% when only the 12 most relevant features according to the ranking in Table 7.8 were used. This result could have been achieved by only using features from P5, P6, and P8, similarly as for *Normal* and *Leakage*. In the case of *Blockage*, a feature from P7 was also added to the selected feature set. The most, 32, features were selected for the *Diverted* flow case. In terms of accuracy, all classifiers obtained at least 91% accuracy, Diverted flow diagnosis achieving the highest 98.3% accuracy.

Feature ranking for fault severity diagnosis

The ranked features for diagnosing fault severities are presented in Table 7.10. Those sensors which did not provide relevant features for fault detection, proved to be useful for fault severity detection, as P1, P2, P3, P4, and the ultrasonic sensor appeared among the most important features. The ratio of DWT and PSD features are almost equal in Table 7.10. For the *Blockage* case, the sensors on the horizontal pipeline provided the most relevant features, which is due to the fact that those features are closer to the location of valve *U39*. The features relevant for diagnosing the severity of leakage are mostly the mean of spectral power and variance of spectral power from the PSD features. All of the ten most relevant features for *Leakage* severity detection were from sensors located on the riser, which is according to expectations: severe leakage caused slugging in the riser with significant pressure fluctuations.

The results of the hybrid feature selection with Bayesian feature-level fusion for fault severity diagnosis of the three fault classes are presented in Figure 7.6, confirming that fault severity detection is more difficult than fault detection. The classification accuracies were not as high as for the fault diagnosis case. *Leakage* severity was the easiest to diagnose achieving over 93% accuracy. This was again due to the fact that severe *Leakage* caused significant pressure fluctuations in the riser. *Blockage* and *Diverted* flow had less obvious flow patterns. The valves were also highly non-linear, which may have resulted in the few first valve positions being mislabelled as *Mild*, when in fact *Normal* conditions persisted.

Table 7.11 shows the number of features selected with the hybrid approach and the classification accuracy achieved with the selected features on the test set for fault severity diagnosis. Even though the *Blockage* severity diagnoses required more than half of the original feature set, with 184 selected features originating from all sensors, it achieved the least fault severity diagnosis accuracy with 66.15%. While *Diverted* flow detection required only 22 features not selecting features from P8, P9, and the ultrasonic sensor. This is due to the fact that *Diverted* flow severity does not influence the top of the riser as where these sensors are located.

7.2.4. Discussion

In this section, the effects of a possible sensor fault on the accuracy of a monitoring result are investigated and discussed. Consider the FDD model constructed using data from sensor S_1, S_2, ..., S_N, as shown in Figure 7.7. The FDD model fuses data in order to provide a monitoring result. In this investigation, the multi-class classifier, results for which were given in the previous section, was considered as the FDD model. Without feature selection, the multi-class classifier achieved an accuracy of 80.10%. To test the robustness of the FDD model without feature selection, a sensor failure was simulated by setting all values in the signal output from a given sensor equal to zero. Such a simulation was conducted for all of the sensors. The achieved fault detection accuracies are summarised in Table 7.12.

Table 7.10: Feature ranking for fault severity diagnosis. The notation of the features are used as indicated in Tables 7.4, 7.5, 7.6, 7.7.

no.	Blockage	Leakage	Diverted
1	$P5^{B2}$	$P9^{msp}$	$P5^{B4}$
2	$P4^{L1}{}_{var}$	$P8^{msp}$	$P5^{L3}{}_{var}$
3	$P1^{L6}{}_{var}$	$P8^{vsp}$	$P5^{B2}$
4	$P1^{L7}{}_{min}$	$P6^{msp}$	$P2^{L5}{}_{var}$
5	$P1^{L7}{}_{max}$	$P5^{B3}$	$P6^{B2}$
6	$P1^{L7}{}_{var}$	$P8^{L1}{}_{var}$	$P5^{L5}{}_{var}$
7	$P4^{L3}{}_{var}$	$P7^{msp}$	$P6^{L3}{}_{var}$
8	$P1^{L6}{}_{min}$	$P7^{vsp}$	$P1^{L7}{}_{min}$
9	$P4^{B4}$	$P6^{vsp}$	$P1^{L6}{}_{var}$
10	$P2^{L7}{}_{var}$	US^{B1}	$P5^{L4}{}_{var}$
11	$P5^{B4}$	$P3^{msp}$	$P1^{B2}$
12	$P3^{L7}{}_{var}$	$P2^{msp}$	$P1^{vsp}$
13	$P4^{L2}{}_{var}$	$P5^{B2}$	$P7^{B2}$
14	$P1^{L5}{}_{var}$	$US^{L6}{}_{var}$	$P2^{msp}$
15	$P4^{L7}{}_{var}$	$P9^{vsp}$	$P4^{vsp}$
16	$P4^{B3}$	$P3^{vsp}$	$P3^{vsp}$
17	$P1^{L6}{}_{max}$	$P1^{msp}$	$P2^{B3}$
18	$P5^{B3}$	$P8^{L4}{}_{var}$	$P3^{B2}$
19	$P2^{L7}{}_{max}$	$P1^{vsp}$	$P1^{L6}{}_{min}$
20	$P4^{B2}$	$P8^{L3}{}_{var}$	$P5^{L5}{}_{max}$
DWT		PSD	

Table 7.11: Fault severity diagnosis: selected number of features and fault severity diagnosis accuracies

	Blockage	Leakage	Diverted
Number of selected features	184	63	22
Accuracy(%)	66.15	93.33	79.76

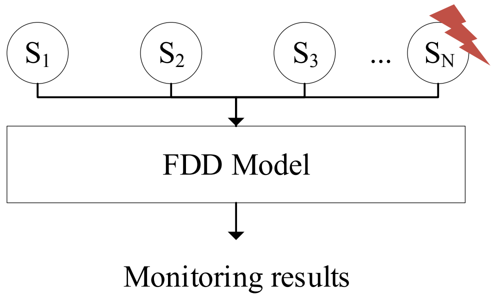


Figure 7.7: FDD model fusing N sensors

Table 7.12: Fault detection results on the test set and the number of selected features with the hybrid method, all good 80.10%

Failed sensor	US	P1	P2	P3	P4	P5	P6	P7	P8	P9
Accuracy (%)	31.63	13.27	35.37	23.81	30.27	13.44	25.85	22.45	33.84	35.37

The classification accuracy for fault diagnosis significantly drops in case of any sensor fault. Furthermore, the accuracy drops dramatically even for those sensors which are not relevant according to the results obtained from ReliefF in Table 7.8. If P5 has a sensor fault, which has provided the most relevant features for fault diagnosis, the accuracy drops to 13.44%. However, the accuracy is even lower if P1 has a sensor fault, which does not seem to provide relevant features for the multi-class case. This observation shows the importance of selecting the optimal number of features from the ranked feature set. The multi-class classifier is able to obtain an accuracy of 91.16% by using only the first 12 features extracted from sensors P5, P6, and P8, as shown in Table 7.9. Hence, it is possible to achieve high fault diagnosis accuracies, even if there is a sensor fault for example in sensor P1, as with feature selection this sensor would simply not be incorporated in the analysis.

7.2.5. Summary

An investigation of the diagnostic properties of sensors and features in the multiphase flow facility case study has been described in this section. ReliefF, which is a supervised feature selection method, was used to evaluate which features were best suited for fault detection and fault severity diagnosis from feature data obtained from several high-frequency pressure sensors and an ultrasonic sensor. Feature selection can not only help in identifying relevant features but also in indicating the relevant sensors for monitoring. The best subset of features differed for the different monitoring problems. This observation was confirmed by applying a Naive Bayes classifier with KDE on the selected subset of features compared to the full set of features for different monitoring problems. Feature selection was conducted with the newly proposed hybrid filter-wrapper approach. The results confirm that the proposed hybrid filter-wrapper approach is able to provide an appropriate selection from the ranked features. The selected feature set is significantly reduced in terms of dimensions compared to the original feature set. Furthermore, high classifications results can be obtained with the reduced feature set. It was also shown that using a multi-class fault tree classifier without feature selection makes the system less robust against sensor faults. It has been highlighted that while using irrelevant features does not necessarily result in performance degradation in the case of a well-parametrized fault classifier, sensor failures can have a significant influence on monitoring performance, even in case of failure of seemingly irrelevant sensors.

7.3. Two-stage Bayesian multi-sensory data fusion for diagnostics

Early diagnosis of faults in industrial machinery is essential to avoid serious and costly failures. Each condition monitoring approach has its own strengths and weaknesses. There is not a single technique that can diagnose all types of faults. As a result, it can be a greater challenge to find the root cause of a problem when only a single feature or sensor is used for monitoring. There is also a greater risk of missed- or false-alarms. For this reason, condition monitoring systems that fuse multiple signal and feature types can be more accurate and robust at correctly identifying faults.

In this section, the two-stage Bayesian framework proposed in Section 5.4 is applied to a component-level condition monitoring problem using the induction motor case study (Stief et al., 2017). The fusion considers acoustic, electric and vibration signals from healthy and faulty induction motors operating under various loading conditions. Features are extracted from the raw sensor signals as described in Section 7.1.1. Load-dependent thresholds are set for each feature. As the dataset is not only labelled according to health state but also according to operating condition, load models can be built in a supervised manner (see Section 5.7.2). Features extracted from each type of signal are fused independently using the Bayesian feature-level fusion with thresholds (see Section 5.5.1) in order to obtain initial diagnoses of the health state of the system. The decision-level fusion takes the feature-level diagnosis results from each of sensor type and fuses them in order to obtain an overall diagnosis of the system (see Section 5.6).

7.3.1. Implementation of the method

The induction motor dataset described in Section 6.1 has been used for this analysis. The dataset contained 58 recordings of 30-second long steady-state measurements from microphone, Microflown, voltage, current and vibration sensors for each fault case under several environmental and loading conditions. Features were extracted from 0.5-second non-overlapping observation windows using the features described in Section 7.1 in Table 7.1.

The training, validation and test sets were set in a way to simulate data available for real-life condition monitoring systems. The training set contained all features from the five healthy recordings without background noise and one from each fault case without background noise at nominal loads, 12 recordings in total. The validation set contained all features from 27 recordings and the test set contained all features from 19 recordings, both containing recordings with and without background noise and loading conditions differing from the nominal load.

Once the features were extracted for each sensor type, the next step involved the determination of alarm thresholds for each feature using the training set. As many of the feature values are load dependent, improved results can be achieved if the thresholds are also load-dependent. Data were recorded from the healthy motor operating under five different loads with no background noise. These five healthy datasets were included in the training set and were used to calculate five thresholds points for each feature. Fitting a first-order polynomial on the feature values from the training set allowed to build a load model as shown in Section 5.7.2 in Figure 5.5.

The thresholds were calculated based on Kernel Density Estimation, as described in Section 5.5.1. The KDEs were constructed from healthy data for each feature and a 95 % confidence interval was determined using the training set. The end of each confidence interval was set as a threshold so that any feature value that exceeded the threshold would trigger an alarm. The likelihood functions for the feature-level fusion were formulated using the training set, which contained recordings from each fault case with nominal loading and no background noise. The elements of the local likelihood functions represent the probability of a feature exceeding the threshold for a given fault case in the training sets. The prior probabilities were set to be equal for each fault case.

The confusion matrices for the decision-level fusion were calculated as described in Section 5.6 using the data from the validation set. The test set was used to determine how well the trained framework was able to diagnose the fault cases of the induction motors.

7.3.2. Results

The two-stage Bayesian framework was applied to induction motor dataset and the results obtained were compared both after the feature-level fusion stage for each sensor type separately shown in Table 7.13 – Table 7.17, as well as after the decision-level fusion shown in Table 7.18. The results are also shown for the feature-level fusion of all the signals shown in Table 7.19 to have a basis of comparison between the decision-level and feature-level fusion.

For clarity the fault cases are listed again:

- F_0: Healthy motor
- F_1: Stator fault: Phase one bypassed in the first phase
- F_2: Stator fault: Phase one bypassed in half of the first phase
- F_3: Stator fault: Phase-phase short-circuit
- F_4: Stator fault: Phase-phase short-circuit with offset point
- F_5: Stator fault: Break of half of the phase one
- F_6: Rotor fault: Two broken rotor bars
- F_7: Bearing fault: Outer raceway defect

The results of applying the proposed diagnostic method are presented in confusion matrices in order to show the false and missed alarms by faults and also to indicate the misclassification rates between faults. In each table, the rows represent the actual health condition F_i of the motors under consideration, while columns represent the diagnosed condition F_j. The diagonal elements of the table represent proper diagnosis, where $F_i = F_j$. For reference, the correct classifications in the diagonal of the matrices are highlighted.

Vibration signals

The Bayesian feature-level fusion using the vibration signals achieved 100% and 94% accuracy when diagnosing the rotor fault F_6 and the bearing fault F_7 respectively. However, the results were only 79% accurate at recognizing healthy motor F_0, as in 21% of the investigated cases the vibration signals falsely indicated stator faults. This may have been due to the fact that the stator faults were less severe and less easy to diagnose than the rotor and bearing faults. Stator fault F_1 was detected with an 98% accuracy. Stator fault F_3 was detected with only 39% accuracy, as the algorithm misclassified it as F_1 in 50% of the cases. The monitoring system did not indicate a fault in 34% of the cases when diagnosing stator faults F_4, resulting in a high missed alarm rate. The F1-score, which is used to provide an overall accuracy measure calculated by Equation 2.2, is 94.62% for the vibration signals,

$$\text{F1} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \qquad \text{(2.2 revisited)}$$

where the false negatives (FN) are the missed alarms, the false positives (FP) are the false alarms and the true positives (TP) are the correctly diagnosed faults F_1-F_7.

Table 7.13: Feature-level fusion results in the form of a confusion matrix when using only vibration signals.

		Diagnosed condition							
		F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
Actual condition	F_0	0.79	0.03	0.04	0.00	0.00	0.14	0.00	0.00
	F_1	0.00	0.98	0.03	0.00	0.00	0.00	0.00	0.00
	F_2	0.05	0.03	0.83	0.00	0.02	0.00	0.00	0.07
	F_3	0.00	0.50	0.03	0.39	0.08	0.00	0.00	0.00
	F_4	0.34	0.07	0.05	0.00	0.43	0.00	0.00	0.11
	F_5	0.00	0.05	0.18	0.00	0.07	0.70	0.00	0.00
	F_6	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
	F_7	0.00	0.00	0.00	0.00	0.01	0.00	0.05	0.94

Current signals

The Bayesian feature-level fusion using the current signals achieved 98% and 97% accuracy for stator faults F_2 and F_4. Bearing fault F_7 was diagnosed with a 100% accuracy. However, the false alarm rate increased. In 62% of the healthy cases, the algorithm falsely predicted stator fault F_2. Furthermore, the motor with two broken rotor bars, which may be considered a severe fault, remained undetected in 20% of the cases. Hence, the current signal on its own did not provide sufficiently accurate diagnostics results. The F1-score was 92.08% for the current signals.

Microflown signals

The Bayesian feature-level fusion using the Microflown signals achieved 100% accuracy for the rotor fault F_6. However, bearing fault F_7 was misclassified as a rotor fault in 36% of the cases. Stator faults F_1 and F_2 were diagnosed with 93% correct classification rate in both cases. The missed alarm rate was 0%. However, there was a 41% false alarm rate in case of the Microflown signals. There was a significant

Table 7.14: Feature-level fusion results in the form of a confusion matrix when using only current signals.

Actual condition	Diagnosed condition F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
F_0	0.36	0.01	0.62	0.00	0.00	0.00	0.00	0.00
F_1	0.00	0.54	0.14	0.00	0.00	0.32	0.00	0.00
F_2	0.00	0.03	0.98	0.00	0.00	0.00	0.00	0.00
F_3	0.00	0.00	0.00	0.29	0.69	0.00	0.02	0.00
F_4	0.03	0.00	0.00	0.00	0.97	0.00	0.01	0.00
F_5	0.00	0.01	0.48	0.00	0.00	0.51	0.00	0.00
F_6	0.20	0.00	0.00	0.00	0.05	0.02	0.73	0.00
F_7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Table 7.15: Feature-level fusion results in the form of a confusion matrix when using only Microflown signals.

Actual condition	Diagnosed condition F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
F_0	0.59	0.01	0.39	0.00	0.00	0.00	0.01	0.00
F_1	0.00	0.93	0.03	0.00	0.00	0.00	0.00	0.04
F_2	0.00	0.00	0.93	0.00	0.00	0.00	0.01	0.07
F_3	0.00	0.00	0.33	0.50	0.00	0.00	0.07	0.10
F_4	0.00	0.00	0.36	0.01	0.48	0.01	0.06	0.08
F_5	0.00	0.00	0.43	0.00	0.00	0.51	0.03	0.03
F_6	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
F_7	0.00	0.00	0.07	0.00	0.00	0.00	0.36	0.57

misclassification phenomena between the stator faults, most commonly they were misclassified as F_2, F_6 and F_7. The F1-score is 95.97 %, which is the highest among all the signals after the feature-level fusion.

Microphone signals

The Bayesian feature-level fusion using the microphone signals also obtained achieved 100% and 97% accuracy when diagnosing the rotor fault F_6 and the bearing fault F_7 respectively. However, they were only 79 % accurate at diagnosing the healthy motor F_0, as in 21 % of the investigated cases the vibration signals falsely indicated stator faults. There were also missed alarms and misclassification present among stator faults F_2, F_3, F_4 and F_5. The F1-score is 94.21% for the microphone signals.

Voltage signals

The Bayesian feature-level fusion using the voltage signals were also achieved 100% and 97% accuracy when diagnosing the rotor fault F_6 and the bearing fault F_7 respectively. However, a significant misclassification rate can be observed for the stator faults. These faults are miss-classified as rotor fault F_6 and bearing fault F_7. A 43% false alarm rate can be observed for stator fault F_2 and there is also a 31% missed alarm rate. The F1-score is 91.06% for the voltage signals, which is the lowest among all

Table 7.16: Feature-level fusion results in the form of a confusion matrix when using only microphone signals.

		Diagnosed condition							
		F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
Actual condition	F_0	0.79	0.02	0.18	0.00	0.01	0.00	0.00	0.00
	F_1	0.00	0.47	0.48	0.02	0.02	0.00	0.00	0.02
	F_2	0.07	0.00	0.93	0.00	0.00	0.00	0.00	0.00
	F_3	0.03	0.17	0.47	0.19	0.13	0.02	0.00	0.00
	F_4	0.05	0.14	0.45	0.33	0.03	0.00	0.00	0.00
	F_5	0.13	0.00	0.38	0.02	0.08	0.41	0.00	0.00
	F_6	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
	F_7	0.00	0.00	0.02	0.00	0.00	0.00	0.01	0.97

Table 7.17: Feature-level fusion results in the form of a confusion matrix when using only voltage signals.

		Diagnosed condition							
		F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
Actual condition	F_0	0.69	0.00	0.25	0.00	0.00	0.00	0.02	0.04
	F_1	0.00	0.51	0.00	0.00	0.00	0.00	0.21	0.28
	F_2	0.43	0.03	0.48	0.00	0.00	0.00	0.00	0.08
	F_3	0.00	0.00	0.00	0.26	0.23	0.01	0.23	0.27
	F_4	0.00	0.00	0.00	0.00	0.50	0.00	0.02	0.48
	F_5	0.05	0.00	0.21	0.00	0.01	0.30	0.08	0.36
	F_6	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
	F_7	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.97

the results achieved by the feature-level fusion of the individual signals.

Results of the two-stage Bayesian fusion

From the previous individual feature-level fusion results, it can be observed that the sensor types reacted differently for various fault modes. Most of them were able to achieve relatively accurate results in case of the rotor fault F_6 and the bearing fault F_7. However, when diagnosing the less serious stator faults, the diagnosis accuracy varied over a wide scale. High missed alarm rates also appeared among the results of feature-level classifiers and only the fusion results using the Microflown signals were without false alarms. The feature-level fusion results have shown that each signal contained relevant information regarding the health state of the induction motor.

The results of the two-stage Bayesian fusion framework are presented in Table 7.18. Rotor fault F_6 and bearing fault F_7 were accurately diagnosed with no missed alarms, although in the case of F_7 there was a 3% misclassification rate. The stator faults were more accurately diagnosed compared to the feature-level results of the individual signals. F_1, F_2 and F_3 were diagnosed with an above 90% accuracy. F_5 still had a low diagnostic accuracy of 54%, which showed that this fault was the most difficult to diagnose with the available signals. The two-stage Bayesian fusion misclassified F_5 as another stator fault in 40% of the cases and there appeared also a 5% missed alarm. In case of other stator faults, such

Table 7.18: Results of the two-stage Bayesian fusion.

		Diagnosed condition							
		F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
Actual condition	F_0	0.97	0.00	0.02	0.00	0.00	0.01	0.00	0.00
	F_1	0.00	0.97	0.02	0.02	0.00	0.00	0.00	0.00
	F_2	0.05	0.03	0.90	0.00	0.00	0.02	0.00	0.00
	F_3	0.02	0.00	0.00	0.81	0.18	0.00	0.00	0.00
	F_4	0.01	0.00	0.00	0.00	0.99	0.00	0.00	0.00
	F_5	0.05	0.03	0.37	0.01	0.00	0.54	0.00	0.00
	F_6	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
	F_7	0.00	0.00	0.00	0.01	0.01	0.00	0.01	0.97

as F_2, F_3 and F_4, a missed alarm rate below 5% was also present. The false alarm rate was 3% caused by misclassification as stator faults. The F1-score is 98.7% for the fused results, which is higher than the F1-scores achieved for any of the individual signals.

Results of feature-level fusion of all signals

For comparison, the results are also given for the case when all of the extracted feature from the raw signals are fused together on the feature-level. The results are presented in Table 7.19. When observing the correct classification rates for F_6, F_7 and the false alarm rates, the results are comparable with the results of the two-stage fusion. However, when looking at the stator faults, the diagnosis accuracies are much lower. Another difference compared to the results of the two-stage Bayesian fusion is the missed alarm rates for the stator faults, which are an order of magnitude higher for the results of feature-level fusion of all signals. The F1-score is 82.30%, which is lower than the results achieved by any of the individual signal types.

Table 7.19: Results of feature-level fusion of all signals.

		Diagnosed condition							
		F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
Actual condition	F_0	0.98	0.00	0.02	0.00	0.00	0.00	0.00	0.00
	F_1	0.47	0.50	0.03	0.00	0.00	0.00	0.00	0.00
	F_2	0.33	0.00	0.67	0.00	0.00	0.00	0.00	0.00
	F_3	0.50	0.00	0.00	0.50	0.00	0.00	0.00	0.00
	F_4	0.37	0.00	0.00	0.03	0.47	0.00	0.13	0.00
	F_5	0.08	0.00	0.41	0.00	0.00	0.51	0.00	0.00
	F_6	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
	F_7	0.04	0.00	0.00	0.00	0.00	0.00	0.02	0.94

7.3.3. Discussion

On the basis of the results obtained, it is possible to evaluate which signal types are best suited for the monitoring of induction motors. When comparing the F1-scores of the feature-level fusion results of the individual signal types, classifiers using the Microflown and vibration signals achieved the highest F1-scores, while the voltage signals achieved the lowest. It is also possible to conclude which faults are easy to diagnose with the available data and which ones are more difficult to distinguish. Clearly, the rotor and bearing faults were the easiest to diagnose, as all signal types were able to diagnose at least one of them with 100% accuracy. The stator faults were the least easy to diagnose. In many cases, the feature-level results and even the fused results contained misclassified observations of stator faults between one another or they were classified as healthy causing false alarms.

Effects of the order of polynomials used for the load model

There are various other ways in which the accuracy of the method could potentially be improved. The load dependency model could be extended in the future for more general cases when there is not only one parameter which defines the operating condition. The order of polynomials used for the load model could be further studied. Depending on the relationship between the features and the thresholds, the order of polynomial may be selected. In the above-presented case, a first-order relationship was assumed for simplicity. In this section, it is explored how the results differ for the fusion when considering second or third order polynomial approximations.

When considering a second order polynomial, the two-stage fusion has a reduced F1-score with 86.36 %. The detection accuracies for stator faults significantly drops and an increased ratio of missed alarms appear. This confirms that assuming a second order relationship between the features and the load-dependent thresholds is not valid in case of the motor dataset.

Using a third order polynomial approximation for calculating the load-dependent thresholds results in the same F1-score of 98.70 % as for the first order polynomials. The missed alarm rate drops to only 1% which is lower than for the first order polynomial. However, the detection accuracy of the stator fault also decreases. Hence, the first order polynomial approximation for calculating the load-dependent thresholds may be assumed to be an acceptable in the case of the induction motor case study.

Table 7.20: Fused results using second order polynomials for calculating the load dependent thresholds.

		Diagnosed condition							
		F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
Actual condition	F_0	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	F_1	0.27	0.23	0.48	0.00	0.00	0.02	0.00	0.00
	F_2	0.55	0.00	0.43	0.00	0.00	0.02	0.00	0.00
	F_3	0.14	0.01	0.02	0.83	0.00	0.00	0.00	0.00
	F_4	0.07	0.00	0.48	0.01	0.45	0.00	0.00	0.00
	F_5	0.32	0.00	0.02	0.18	0.00	0.48	0.00	0.01
	F_6	0.00	0.00	0.00	0.00	0.02	0.00	0.98	0.00
	F_7	0.00	0.00	0.00	0.11	0.03	0.00	0.00	0.87

Table 7.21: Fused results using third order polynomials for calculating the load dependent thresholds.

Actual condition	Diagnosed condition F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
F_0	0.99	0.00	0.01	0.00	0.00	0.00	0.00	0.00
F_1	0.00	0.96	0.02	0.00	0.00	0.03	0.00	0.00
F_2	0.07	0.00	0.93	0.00	0.00	0.00	0.00	0.01
F_3	0.00	0.01	0.00	0.39	0.60	0.00	0.00	0.00
F_4	0.00	0.00	0.00	0.33	0.67	0.00	0.00	0.00
F_5	0.06	0.00	0.33	0.00	0.00	0.61	0.00	0.00
F_6	0.01	0.00	0.00	0.00	0.00	0.00	0.99	0.00
F_7	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.98

Effects of the confidence interval

Another parameter of the two-stage framework is the width of the confidence interval. In the presented implementation, this interval was set using the 2σ rule with a 95% confidence interval. The threshold may be further optimized to the monitoring problem. Below an example is given about how the threshold setting may affect the results of the diagnostics. Considering a higher threshold with a 97% confidence interval the diagnosis accuracy of stator faults becomes higher, while the false and missed alarm rates increase. The F1-score is 96.98%, which is less than 1% lower than for the results achieved with the 95% confidence interval.

Table 7.22: Fused results with thresholds set based on a 97% confidence interval.

Actual condition	Diagnosed condition F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
F_0	0.95	0.00	0.03	0.00	0.00	0.02	0.00	0.00
F_1	0.00	0.95	0.05	0.00	0.00	0.00	0.00	0.00
F_2	0.04	0.00	0.91	0.00	0.01	0.03	0.00	0.01
F_3	0.04	0.01	0.06	0.78	0.11	0.01	0.00	0.00
F_4	0.23	0.00	0.00	0.08	0.69	0.00	0.00	0.00
F_5	0.02	0.01	0.18	0.01	0.00	0.78	0.00	0.00
F_6	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
F_7	0.00	0.00	0.01	0.00	0.02	0.00	0.00	0.97

Presentation of the fused result

The results of the feature- and decision-level fusion are in the form of posterior probabilities. Even though the current implementation of the two-stage Bayesian framework uses the maximum a posteriori probabilities as the final fault diagnostic results, all the posterior probabilities are available to provide insight into the decision support system. The end user may be interested to know what was the level of uncertainty when determining the health state for a given an observation. In Figure 7.8 an example is given how the posterior probabilities may look like after the decision-level fusion. In this example, the observation under analysis is from fault class F_2 with no background noise and with a load lower than the

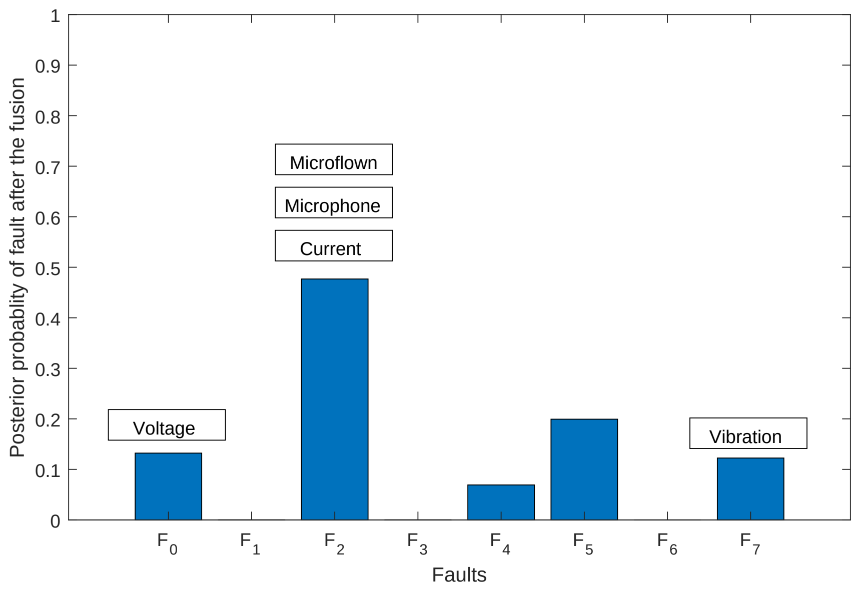


Figure 7.8: Posterior probabilities for an observation from fault class F_2 after the fusion, the feature-level fusion predictions are indicated for each sensor type

nominal value. The feature-level fusion predictions are indicated for each sensor type. The Microflown, microphone and current signals were able to correctly diagnose F_2. However, the vibration and current signals were mistaken in the diagnosis. The decision-level fusion was able to account for their error and provided the final result for this observation as F_2 according to the MAP test. Similarly, it is possible to obtain the results of the feature-level fusion in such a form, which contributes to the transparency of the method.

7.3.4. Summary

In this section, it was shown that the fusion of acoustic, electric and vibration signals utilizing the newly-proposed two-stage Bayesian framework can significantly improve the accuracy of diagnosing faults in induction motors. The method was validated on the induction motor case study.

Features were extracted from each sensor type and fused independently on the feature-level in order to obtain initial diagnoses of the health state of the system. After this stage, the number of false alarms ranged from 5 % of the total number of test cases for the microphone signals, up to 72 % of the total number of test cases when considering voltage signals.

At the decision-level fusion stage, the diagnoses obtained from the feature-level fusion for each sensor type were fused in order to obtain an overall diagnosis of the system. Applying the decision-level stage allowed the number of false alarms to be reduced to 3 %. Furthermore, the two-stage Bayesian framework achieved higher diagnostic accuracies than the feature-level fusion stage was able to achieve on the signals individually or fusing all signals together. The framework was proven to increase the accuracy of diagnosis with missed alarms only being observed in the case of stator fault and always being below 5%.

The results obtained indicated that the proposed method can increase the reliability and efficiency of fault detection compared to a single stage feature-level fusion approach. The method also provides a structured approach for comparing the performance of different signal types in diagnosing various induction motor fault modes.

A limitation of this approach is that, in its present form, the method is only suitable for steady-state signals and does not take into account the severity of a feature exceeding the threshold. Furthermore, as a data-driven method, the accuracy of the likelihood functions used in the approach will be dependent on the availability of large quantities of comparable measurement data. Furthermore, it was assumed that all health states are equally likely, whereas in practice this is unlikely to be the case. Improvement might be obtained by calculating a priori probabilities on the basis of historical fleet data described in Section 2.2.5.

7.4. A PCA - two-stage Bayesian sensor fusion approach for diagnosing electrical and mechanical faults in induction motors

In this section, the two-stage Bayesian framework proposed in Section 5.4 is further investigated for sensor fusion and diagnostics applications at the component-level for the induction motor case study (Stief et al., 2019c).

In the previously described application of the two-stage Bayesian framework in Section 7.3, it was shown that by fusing independent diagnoses of different sensor types at the decision-level, the false and missed alarm rates of the fault diagnostics framework could be significantly reduced. Simple linear models of expected threshold values relative to the operating condition were applied to account for the operating condition dependency of features. Through the addition of Principal Component Analysis (PCA) and the Gaussian Naive Bayes (GNB) classifier, this implementation of the two-stage Bayesian framework does not require monitoring thresholds to be defined, as the posterior fault class probabilities can be directly calculated.

The features extracted in Section 7.1.1 are used as the inputs of the Bayesian framework. The extracted features were observed to be highly correlated in Section 7.1. Many correlated features mean that the naive conditional feature independence assumption is corrupted, which may cause over-fitting and effect the fault detection algorithm towards certain diagnoses, as described in Section 5.2.3. A PCA step is included to remove the correlations that are present in the extracted features and reduce the influence of the operating and environmental conditions. At the feature-level, principal components of the features grouped by sensor type are fused with a GNB classifier, as described in Section 5.5.2. At the decision-level, the results of the feature-level fusion are fused in order to create a final diagnosis.

The generality of the algorithm is investigated by omitting data recorded at selected operating and environmental conditions from the training set and subsequently testing the trained model using the omitted data. It is shown that the proposed method is able to accurately diagnose faults even for operating and environmental conditions not present in the training set.

7.4.1. Implementation of the method

The structure of the PCA - Two-stage Bayesian sensor fusion approach is shown in Figure 7.9. Firstly, the features **x** are divided by sensor type D. Arithmetic means μ_{x_i,D_j} and standard deviations σ_{x_i,D_j} are calculated for each x_i feature and D_j sensor type. Then, the features are normalized. The PCA step is included to remove correlations between the features. The PCA transformation is conducted on the features by sensor type, as described in Section 5.7.2. PCA calculates the UL_{D_j} scores and P_{D_j} loadings for each sensor type using the training set, which is used for transforming new observations in a similar way to the principal component space.

The Bayesian feature-level fusion takes the principal components from each sensor type and provides the results of the diagnostics. The number of principal components considered for each sensor type are calculated using the validation set in a way that the performance of the algorithm is maximized whilst the false and missed alarm rates are reduced. The feature-level diagnosis accuracy of each signal type is used as an optimization parameter, as described in Section 5.7.2.

The Gaussian Naive Bayes (GNB) classifier was selected for the implementation of the feature-level fusion. GNB allows the posterior fault class probabilities to be directly calculated without considering monitoring thresholds. To calculate the conditional probabilities of the GNB classifier according to Equation 5.19,

$$P(x_j|c_i(\mu_{i,j}, \sigma_{i,j}) = \frac{1}{\sigma_{i,j}\sqrt{2\pi}} e^{\frac{-(x_j - \mu_{i,j})^2}{2\sigma_{i,j}^2}} \quad \text{(5.19 revisited)}$$

the μ_{x_i,D_j,c_k} means and σ_{x_i,D_j,c_k} standard deviations of the principal components are calculated for each c_k fault class in the labelled training set.

The decision-level fusion of the feature-level results is conducted by sensor type, as described in Section 5.6. The confusion matrices are obtained for each sensor type D_k using the validation set by

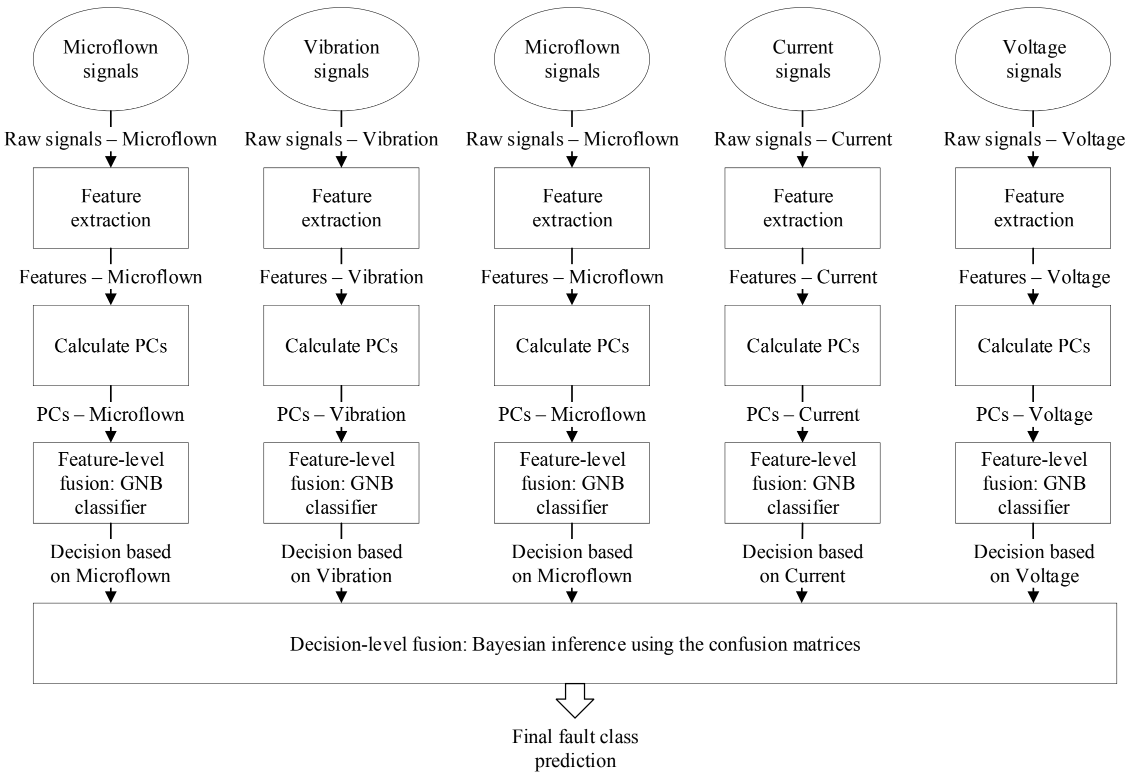


Figure 7.9: The flow diagram of the PCA - Two-stage Bayesian sensor fusion framework applied to the induction motor case study (Stief et al., 2019c)

Equation 5.38:

$$G_{D_k} = \begin{pmatrix} P(F_1|F_1) & P(F_1|F_2) & \cdots & P(F_1|F_M) \\ P(F_2|F_1) & P(F_2|F_2) & \cdots & P(F_2|F_M) \\ \vdots & \vdots & P(F_i|F_i) & \vdots \\ P(F_M|F_1) & P(F_M|F_2) & \cdots & P(F_M|F_M) \end{pmatrix} \qquad \text{(5.38 revisited)}$$

The predicted fault class label c_{pred} for a new observation is determined by Equation 5.40:

The 480 features of the 3480 observations were grouped by signal types into five groups namely vibration features, current features, Microflown features, microphone features, and voltage features. The data was then split into a training set, a validation set, and a test set, in the same way for the 5 signal types. The division is described in the next section. The training sets were used to train the feature-level fusion stage, the validation sets were used to calculate the confusion matrices for the decision-level fusion. Finally, the test sets were used to test the performance of the algorithm.

7.4.2. Results

In order to illustrate the performance of the described algorithm with respect to different loading and environmental noise conditions, the experimental data was divided into different training, validation, and test sets. In Test case A a random split was applied. In Test case B and C eight entire datasets (one from each fault case) were included in the test set with no datasets from experiments conducted at this loading condition being considered in the training or validation sets. In Test Case B the lowest load datasets with no background noise are the test set. In Test Case D the highest load datasets with background noise are the test set. The aim of testing different divisions for testing, validation, and training is to observe the performance of the algorithm under different operating conditions, particularly under loading conditions that were not considered during model training.

Test case A: Random split

Test case A was used to evaluate the overall performance of the algorithm. The total 3480 observations were randomly split into a training set, validation set and test set with a respective ratio of 60-20-20%. The random split was applied 100 times and the averaged results are shown in Table 7.23. The columns represent the conditions diagnosed by the algorithm while the rows represent the actual fault conditions of the motors. The healthy motor was correctly diagnosed in 94% of the cases with a 6% false alarm rate in case of F_2 stator fault. Missed alarms are present for F_2, however, it is only 2%. F_2 is the least severe fault among the 7 seeded faults, which explains this behaviour. The successful detection rate is above 98% for all fault cases, with 100% success rate for F_1, F_5, F_6 and F_7. Among the stator faults the following scenario can be observed: F_3 and F_4 are sometimes misdiagnosed as one another, as they are the variations of the same fault: F_3 is phase-phase short-circuit, while F_4 is phase-phase short-circuit with an offset point. To give an overall measure of the test accuracy, the F1-score is calculated to be 99.32% by Equation 2.2.

Table 7.23: Results of the two-stage Bayesian fusion for Test Case A - Random split.

		Diagnosed condition							
		F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
Actual condition	F_0	0.94	0.00	0.06	0.00	0.00	0.00	0.00	0.00
	F_1	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
	F_2	0.02	0.00	0.98	0.00	0.00	0.00	0.00	0.00
	F_3	0.00	0.00	0.00	0.98	0.02	0.00	0.00	0.00
	F_4	0.00	0.00	0.00	0.02	0.98	0.00	0.00	0.00
	F_5	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
	F_6	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
	F_7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Test Case B: Lowest load, no noise

In Test Case B, the test set was formed of data taken from the lowest loading conditions, with no datasets from experiments conducted at this loading condition being considered in the training or validation sets. The aim was to test the performance of the algorithm under load conditions which are lower than those contained within the training and validation sets. The results are shown in Table 7.24. The accuracy of the algorithm was 100% when diagnosing the healthy condition (F_0); there were no false alarms. When diagnosing broken rotor bars and bearing faults (F_6 and F_7) the algorithm performed with 100% accuracy.

However the performance for the stator faults needs further analysis: whilst fault F_1 and F_3 are diagnosed with a success rate of 97% and 100%, faults F_2, F_4 and F_5 were identified less reliably. The algorithm was able to diagnose the F_2 stator fault in only 57% of the cases. In 43% of the cases, the algorithm misdiagnosed F_2, either as healthy or as the other similar stator faults F_1 and F_5. This was because F_2, as the least severe fault, was the most difficult to diagnose. The algorithm was also unable to distinguish between fault modes F_4 and F_5, in 20% and 13% of the cases respectively. F_5 was also mistakenly diagnosed as other stator faults phase one bypassed in 10% of the cases. This result indicates that in the case of loading conditions lower than those seen in the training datasets the algorithm can

Table 7.24: Results of the two-stage Bayesian fusion for Test Case B - Lowest load, no noise.

		Diagnosed condition							
		F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
Actual condition	F_0	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	F_1	0.02	0.97	0.00	0.00	0.00	0.01	0.00	0.00
	F_2	0.22	0.07	0.57	0.00	0.00	0.14	0.00	0.00
	F_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
	F_4	0.00	0.00	0.00	0.00	0.80	0.20	0.00	0.00
	F_5	0.02	0.07	0.03	0.00	0.13	0.75	0.00	0.00
	F_6	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
	F_7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

accurately determine the type of fault, however, it is unable to accurately ascertain the severity of the fault.

Test case C: Highest load with noise

Test case C used datasets recorded for the highest loading conditions with background noise as the test set, with no data from this loading condition being considered in the training. This test case investigates the performance of the algorithm for loading conditions exceeding those considered in the training set and for unique environmental conditions, specifically when the background noise is at increased levels. The results are shown in Table 7.25. The correct diagnosis of the healthy motor was 100%, as well as the diagnosis for F_1, F_4, F_5, F_6 and F_7. In the case of stator fault F_2, there is a 2% missed alarm rate. In case of stator fault F_3, the algorithm misdiagnoses F_3 as F_4 in 8% of the cases. These phenomena are similar to the ones observed in Test case A: the stator faults are less severe and less easy to diagnose. Due to fault similarities the algorithm can sometimes misdiagnose stator fault severities or confuse them with the healthy motor. The F1-score is 99.88%, which is even higher than the random split test case.

Table 7.25: Results of the two-stage Bayesian fusion for Test Case C - Highest load with noise

		Diagnosed condition							
		F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
Actual condition	F_0	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	F_1	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
	F_2	0.02	0.00	0.98	0.00	0.00	0.00	0.00	0.00
	F_3	0.00	0.00	0.00	0.92	0.08	0.00	0.00	0.00
	F_4	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
	F_5	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
	F_6	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
	F_7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Principal Components

The number of principal components is shown in Table 7.26 for each signal type together with the variance explained to compliment the results in the above-presented test cases. In case of the random split in Test Case A the variance explained by the chosen principal components is always above 90%. In the case of Test Case B and C, the number of chosen principal components are less than for Test Case A. This is due to the specific loading and noise conditions chosen for the test sets. The first few principal components have been analyzed for all signal types to determine if there is any feature which dominates the principal component coefficients in the loading matrix. It was found that there was no single feature which would stand out for any signal type, therefore the importance of PCA for correlation reduction is further confirmed.

Figure 7.10 presents the histograms and underlying Gaussian distributions of the first principal component of the vibration signal by fault conditions. The distributions for each fault types have distinct mean and variance values and are not significantly different from Gaussian distributions. It can be observed that F_6 and F_7 are the most distinguishable from F_0, while the other stator faults have overlapped with F_0. It should be noted that F_0 shows evidence of multi-modal behaviour. This is due to the additional

Table 7.26: Number of principal components and variance explained

		Vibration	Current	Microflown	Microphone	Voltage
Test A	PC	30	18	20	30	28
	σ expl.	92%	95%	90%	95%	91%
Test B	PC	22	12	14	25	15
	σ expl.	89%	91%	86%	94%	82%
Test C	PC	23	16	14	27	29
	σ expl.	88%	94%	84%	94%	92%

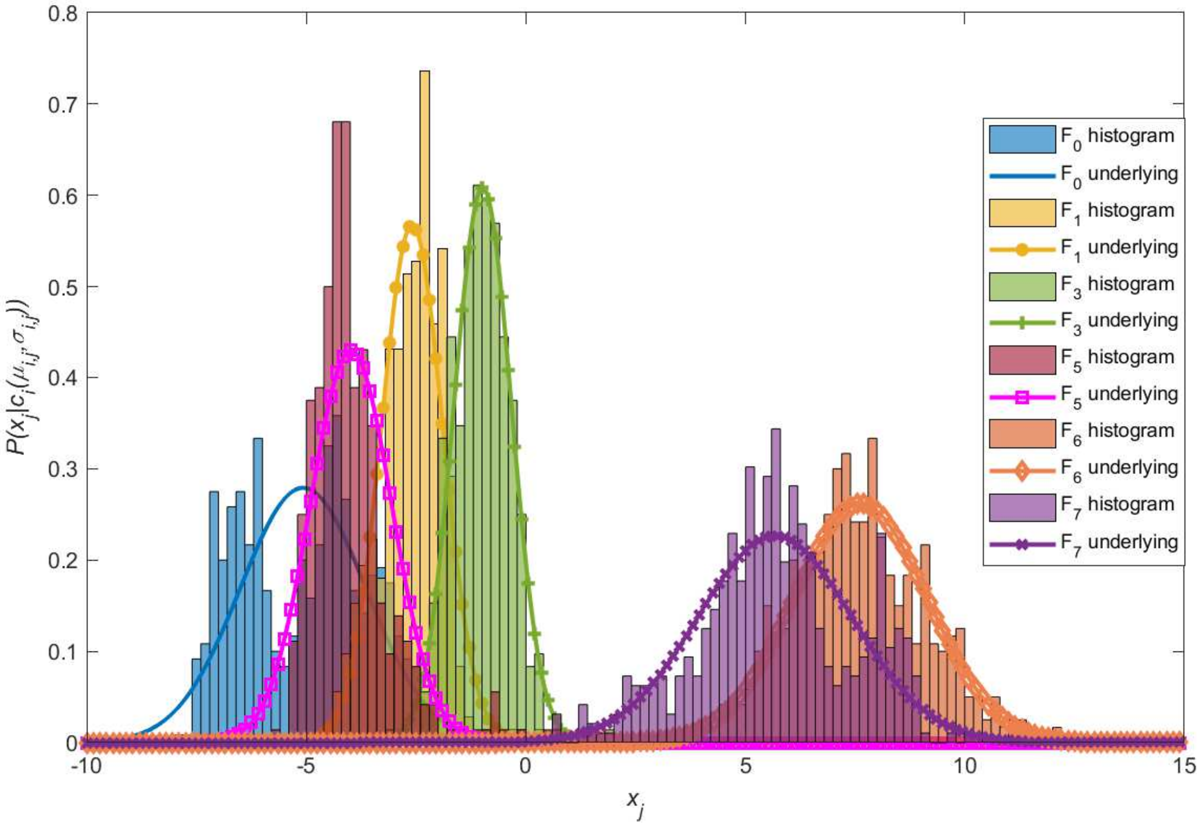


Figure 7.10: The histograms and underlying normal distributions of the first principal component of the vibration signal by fault conditions (Stief et al., 2019c)

background noise incorporated to investigate the influence of different environmental conditions on the accuracy of diagnosis. However, as shown for Test Case A, B, and C, this noise did not significantly influence the resulting likelihood calculations.

Single stage feature-level data fusion

A comparison of the performance of the two-stage approach relative to a more standard single-stage approach, where sensors are not separated according to type, but instead all fused in a single stage, was performed. The total 3480 observations were randomly split according to the conventional 70-30% partition to a training set and test set. The random split was applied 100 times to a single stage approach and the averaged results are shown in Table 7.27. The results show that the performance of the single-stage algorithm significantly drops compared to the results of the two-stage method shown in Table 7.23.

Table 7.27: Results of the single stage feature-level fusion

Actual condition	Diagnosed condition F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
F_0	0.09	0.10	0.32	0.05	0.11	0.32	0.01	0.00
F_1	0.02	0.94	0.01	0.01	0.00	0.01	0.01	0.00
F_2	0.03	0.01	0.90	0.02	0.01	0.03	0.00	0.00
F_3	0.02	0.00	0.00	0.93	0.05	0.00	0.00	0.00
F_4	0.02	0.00	0.01	0.16	0.77	0.03	0.01	0.00
F_5	0.01	0.01	0.01	0.01	0.01	0.94	0.01	0.00
F_6	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
F_7	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.98

The most significant difference appears in the reduced successful detection of the healthy motor, with the single stage approach yielding false alarms in 91% of test cases. The F1-score is 92%.

Comparison of results with Support Vector Machine

To provide a quantitative comparison with another classifier, the proposed PCA - Two-Stage Bayesian method is compared with the well-known Support Vector Machine (SVM). Test Case A, B, and C are repeated using *fitcecoc*, which is the default Matlab®implementation of SVM for multiclass problems. The F1-scores are compared. Similarly to the investigation described in the previous section, the SVM was applied in a single stage. A 70-30% data split was applied and repeated 100 times resulting in a 96.96% F1-score for Test Case A. This result is 0.64% better than the proposed method. For Test Case B the F1-score for SVM was 96.15%, which is 1.84% below what was achieved with the newly proposed method. For Test Case C the F1-score for SVM was 97.8%, which is 2.08% below what was achieved with the newly proposed method. Whilst the performance of the two approaches are comparable, an advantage of PCA - Two-stage Bayesian method lies in its transparency. The final decision of the newly proposed method may be traced back to the individual signal types, while in the case of SVM the final result is the outcome of a "black-box". PCA - Two-stage Bayesian method has also the advantage of modularity over SVM. In case of a newly added or a failed sensor, only the affected feature-level fusion module has to be trained or removed, while in case of SVM the whole monitoring system has to be retrained. Furthermore, the method also provided improved performance in case of environmental and loading conditions not contained in the training set, as shown in Test Case B and C.

Signal types separately versus two-stage fusion

Table 7.28 shows the performance of only considering a single-stage fusion of features from a single signal type, for the random split case described in Section 7.4.2. For comparison, the equivalent performance from the two-stage approach, which fuses the data from all sensors types in the global fusion stage, is also given. Results are given in terms of the proportion of correct diagnoses, which are equivalent to the values on the diagonal of the previously presented results in Tables 7.23, 7.24, and 7.25. It is evident that the two-stage data fusion of multiple signal types outperforms the equivalent results when only considering a single signal type. This is due to the fact that different sensor-types have different strengths and weaknesses. For example, it may be observed that the analysis based only on vibration

Table 7.28: Proportion of correct diagnosis for each fault types when considering each signal type individually and after two-stage fusion

	Single-Stage Fusion by Signal Type					Two-stage
	Vibration	Current	Microflown	Microphone	Voltage	
F_0	0.75	0.88	0.91	0.73	0.66	0.94
F_1	0.92	0.98	0.90	0.90	0.83	1.00
F_2	0.87	0.62	0.84	0.82	0.85	0.98
F_3	0.86	0.82	0.91	0.90	0.86	0.98
F_4	0.87	0.72	0.87	0.88	0.88	0.98
F_5	0.90	0.92	0.94	0.92	0.90	1.00
F_6	1.00	0.89	0.98	0.97	0.99	1.00
F_7	1.00	0.96	0.91	0.99	1.00	1.00

signals accurately diagnosed the mechanical bearing fault F_7 in 100% of test cases, but was only able to diagnose an electrical stator fault such as F_1 in 92% of cases. In contrast, when only current signals were considered, stator fault F_1 was diagnosed correctly in 98% of cases, but bearing fault F_7 was only diagnosed correctly in 96% of cases. When the two signals are fused, the conditional probabilities in the global confusion matrix effectively gives greater weight to vibration signals and less weight to current signals when diagnosing mechanical faults and vice-versa in the case of diagnosing electrical faults. This leverages the strengths of each sensor type for fault monitoring and minimizes the impact of the weaknesses.

7.4.3. Discussion

In this section the results and the structure of the algorithm are discussed further, highlighting the observed strengths and weaknesses of the algorithm.

Implementation and constraints

The training of the method takes place offline using historical datasets containing healthy and faulty data. Once the model is trained, the diagnosis can be performed either online or offline. By applying a sliding window of the same size as used for training, the new sensor measurements can be fed into the two-stage Bayesian classifier online after the feature extraction and PCA steps have been performed. The width of the window could be different based on the nature of the monitored system, the extracted features and the data available. The computational complexity of the classifier is proportional to the number of principal components retained and the number of fault modes monitored. The computational complexity of the feature extraction and PCA step depends on the number of features extracted and the size of the sliding window. For a better representation of the original feature space non-linear multivariate methods, like kernel-PCA (Choi et al., 2005) could be explored in the future instead of the currently used linear PCA. It should also be noted that the features used as inputs to the method may also be refined according to state of the art signal processing and feature extraction methods so that they may better discriminate between different health states. Thus, the accuracy and reliability of the approach would likely be improved further.

Algorithm validation

Three different algorithm validation test cases were presented by splitting the data into different training sets, test sets, and validation sets. It has been shown that for small data sets the simple split-sample estimates can be biased and cross-validation is more suitable for the prediction assessment of the classifiers (Molinaro et al., 2005). In the case of a two-stage method, this result is still valid theoretically, however it is infeasible due to the increase in the number of computational steps associated with the addition of further fusion stages. Specifically, relative to a simple single stage fusion, when implementing cross-validation on a two-stage approach, the method becomes N^2 more computationally expensive, where N is the number of the observations, as both the local and the global stage have to be trained using separate training sets. In this section, a pragmatic split-sample method was considered. It is also foreseen that such an approach would be applicable for applications of the method with larger volumes of data sets available. In the future, increases in computing power might also allow the cross-validation approach to be feasibly applied.

Naive Bayes classifier using kernel density estimate

The GNB classifier is a parametric method which assumes a normal distribution of the observed variables. The more the distribution of the observation variables differs from the normal distribution, the less accurate the method is. One possible way to eliminate this Gaussian assumption is to use a naive Bayes classifier with kernel density estimate (KDE), where the probability density function of the features are estimated using a non-parametric kernel distribution. Such an approach can be used when there is no prior knowledge regarding the distribution of the data, no assumptions are made or a parametric distribution cannot describe the data. Tests conducted using such naive Bayes classifier with KDE, with the same random split as described in Test Case A, yielded comparable results to the Gaussian Naive Bayes classifier. The naive Bayes classifier with KDE resulted in correct classification rates in the $\pm 2\%$ range compared to the results in Table 7.23, while the F1-score is 99.64% which is 0.32% better compared to the results in Table 7.23. However, when applying KDE, the computation time was two magnitudes greater for the local stage than for the case of the Gaussian Naive Bayes classifier. It took 4.277s for the original method to train the local stage and obtain the confusion matrices for the vibration signals while the same computation took 351.78s with KDE. The processing hardware was an Intel®Core™ i5-4300U, 1.9 GHz.

Two-stage data fusion without PCA

Whilst not the primary focus of this section, it is worth noting that an investigation into the importance of incorporating the PCA step into the algorithm was also performed. It was observed that when the PCA step was omitted from the algorithm, all test cases, including fault cases, were subsequently diagnosed as being healthy (F_0). This was due to the load dependency of the features. This observation indicates that a PCA step, or similar, ensures that the algorithm is robust against changing loading and environmental conditions.

Advantages of the Method

The preceding sections provide quantifiable comparisons of the performance of the algorithm when including the novel steps of applying a Gaussian Naive Bayes classifier and splitting the approach into two stages, relative to the cases when the steps are omitted. Due to the multitude of ways of properly designing and tuning various algorithms, it is infeasible to perform similarly rigorous quantitative comparisons to benchmark the method relative to other data-driven fault detection methods. However, qualitative comparisons, which can guide design decisions at an early stage of the analytics development process, can be made. The main advantages of the proposed method are its transparency and modularity. In contrast to many other data-driven fault diagnostics methods such as Support Vector Machines or Neural Networks, the decision making process of the algorithm is easily back traceable from the global predictions to the inputs of the local stage to identify how the different sensors reacted to a fault. Such transparency is important for cases where the algorithm will be used to support maintenance decisions. Whilst in this section only maximum a posteriori probabilities were considered, in practice the Bayesian sensor fusion approach allows the results to be presented in the form of likelihoods, as shown in Figure 7.8. This additional insight can support maintenance decisions. The modularity of the approach, achieved by splitting the data fusion into two stages, also offers further advantages when considering practical implementation. In case of removing sensors from the systems, there is no need to retrain the whole model, as the removed sensor type can easily be omitted from the decision level fusion, which is not possible for other fault diagnostics methods which only consider feature-level data fusion. Similarly, additional sensor types may be readily incorporated into the analysis with limited requirements for retraining.

7.4.4. Summary

In this section, the performance of a proposed PCA-two stage Bayesian sensor fusion method has been evaluated under various test scenarios at the component-level monitoring scale using the induction motor case study. The framework was shown to be able to diagnose stator faults, broken rotor bar faults and bearing faults in induction motors, with low false and missed alarm rates. The framework also proved its ability to diagnose faults under different loading and environmental conditions. In addition to discussing the several advantages of the presented method, the limitations of the method were also highlighted. For example, it was shown that the method is capable of correctly distinguishing different types of fault, however, to consistently distinguish between different fault severities, adequate training sets are required at comparable loading conditions.

7.5. Process and alarm data integration under the two-stage Bayesian framework for fault diagnostics

In Section 2.2 it was discussed that industrial scale process plants may be monitored by various data acquisition systems generating disparate data in large quantities, which may be in the form of process and alarm data, amongst others. Process data containing various sensor measurements are commonly used for supervisory control and monitoring. However, they are also often used for fault detection and diagnostics. Data-driven monitoring methods using process data have been developed for early incipient fault detection, for fault classification and for fault prediction (Yang et al., 2013; Bauer and Thornhill, 2008; Thornhill, 2005). Alarms also play an important role in maintaining safe, reliable and efficient operations in industrial plants. An alarm can be set based on a threshold related to a process variable to determine the instantaneous health state of the process, often as part of a protection system. Alarms may also be used to indicate sensor and communication failures. They are also used for process monitoring in alarm management systems and for pattern mining in multiple alarm flood sequences (Lai and Chen, 2017). The results of pattern mining algorithms can provide valuable insights about the health state of the process, can support an operator pinpoint the root cause of a problem and can help in the identification of bad alarm configurations (Lai et al., 2017). Other applications use alarm data to extract fault templates from a set of alarms to identify faults (Charbonnier et al., 2016).

Despite the fact that both process signals and alarms may be used to perform similar, although not identical, functions in process monitoring, they are rarely used in combination. One of the main reasons for this is that process data is numerical and sampled continually, while alarm data is either in the form of a historical log or is in binary format and appears at discrete times. The fusion of such mixed data is not a trivial problem. There are a limited number of works reported in the literature which have attempted to deal with high dimensional mixed type variables especially in statistical process control (SPC). A density-based statistical process control approach is proposed in (Ning and Tsung, 2012) to use high-dimensional mixed type data for process monitoring based on the local outlier factor scheme. Another work (Tuerhong and Kim, 2014) proposed a Gower distance-based multivariate control chart for a mixture of continuous and categorical variables. Whilst both of these works considered data which contained both categorical and continuous numerical process variables, alarms were not considered.

Alarm and process data contain complementary information for plant-level monitoring regarding the health state of the process, as described in Section 2.2. Therefore, their fusion can potentially improve the results of fault diagnostics. As alarms are binary and process measurements are numerical their fusion on the data- or feature-level is problematic, however decision-level fusion is a promising direction towards using alarm and process data together.

In this section, the two-stage Bayesian framework proposed in Section 5.4 is applied for the fusion of alarm and process data from the multiphase flow facility case study (Section 6.2) on the decision-level for fault diagnostics, targeting industrial processes. It is shown that the Bayesian framework can also be applied to plant-level monitoring problems, resulting in a framework for fusing heterogeneous alarm and process data (Stief et al., 2018b) for the detection of induced faults under operating condition A and B.

- Operating condition A: 120 Sm^3/h air flow rate, 0.1 kg/s water flow rate

- Operating condition B: 150 Sm^3/h air flow rate, 0.5 kg/s water flow rate

7.5.1. Implementation of the method

The two-stage Bayesian framework is extended to process condition monitoring problems, resulting in a framework for fusing heterogeneous alarm and process data, as shown in Figure 7.11. Instead of the raw process data, the principal components of the process variables are used as the inputs of a NB classifier at the feature-level fusion stage. This step reduces the correlation between the process variables and reduces the dimension of the data. The alarm history is transformed into binary alarm features, which are inputs to another NB classifier at the feature-level fusion. The decision-level stage of the method fuses the diagnostics results of the alarm and process data and provides the final classification result.

Alarms

The following terminology is defined to differentiate between the varieties of information that may be contained within the alarm logs. Alarm data are usually available in the form of historical alarm logs. Alarm type is defined as a possible alarm connected to a sensor measurement. The status of an alarm

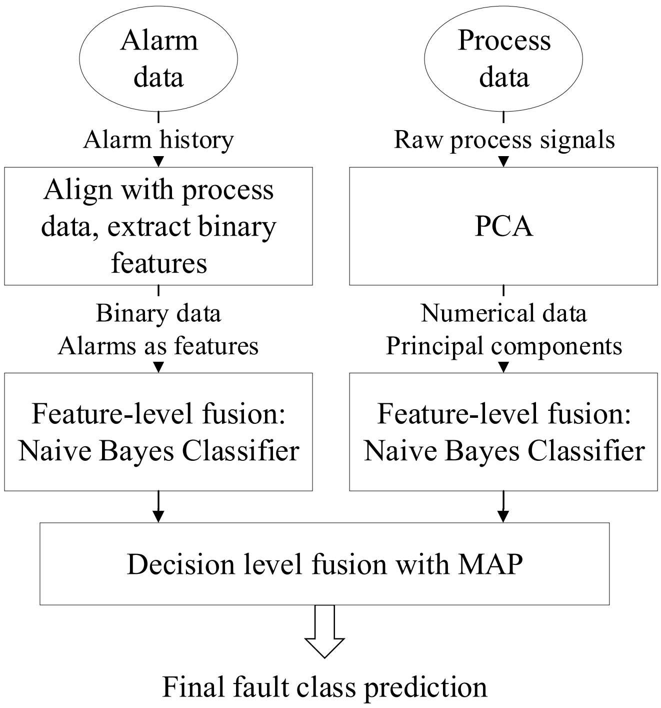


Figure 7.11: Flow diagram of the two-stage Bayesian framework for fusing alarms and process data

type is considered to indicate whether or not a specific alarm type is active at a given time. An alarm event represents the instance when the status of an alarm type transitions from inactive, or no alarm, to active. In order to process alarm data with the NB classifier, the alarm logs must be pre-processed, aligned with the associated process data and converted to a binary form. An alarm log contains the timestamp, the corresponding sensor tag, and status information, whether they are active or inactive. Based on this information they can be transformed into a binary form aligned with the process measurements by rounding the time of the alarm event down to the nearest second. The binary alarm value of alarm $A_{k,i}$ for sensor i and timestamp k can be formulated as follows:

$$A_{k,i} = \begin{cases} 0, & \begin{array}{l}\text{if no active alarm for timestamp } k \text{ and no active alarm} \\ \text{for timestamp } k-1 \text{ or inactive alarm for timestamp } k\end{array} \\ 1, & \begin{array}{l}\text{if no active alarm for timestamp } k \text{ and active alarm value} \\ \text{for timestamp } k-1\text{, or active alarm for timestamp } k.\end{array} \end{cases}$$

Each alarm type is considered as a separate feature. Once the alarm data is brought to a binary form, where $\mathbf{a} = \{a_1, a_2, ..., a_m\}$ are the possible alarm features, the local likelihood functions $P(a_k|F_i)$ can be calculated for each fault case F_i. The sum of the alarm values are divided by n, which is the total number of observations for a fault case F_i,

$$P(a_k|F_i) = \frac{\sum_{j=1}^{n} a_{k,j}}{n} \tag{7.3}$$

Unless there is prior knowledge available about the distribution of faults, $P(F_i)$ is assumed to have a uniform distribution. The probability that an observation is classified as fault F_i, given that the alarm features have values **a** can be expressed in a similar way, as the Bayesian feature-level fusion was described by Equation 5.26 in Section 5.5.1.

$$P(F_i|\mathbf{a}) = \frac{P(\mathbf{a}|F_i)P(F_i)}{\sum_{j=1}^{m} P(a_j|F_j)} \tag{7.4}$$

Process data

The process data contains measurements from different process sensors. These sensor measurements can be treated as features. In order to reduce both the correlation between sensor measurements and the dimensionality of the dataset, PCA is applied to the raw process data as described in Section 5.7.2. The NB classifier takes the principal components as process features. The Bayesian feature-level fusion of process data is conducted with the calculation of thresholds, as described in Section 5.5.1. KDE determines the cumulative density functions, based on which, fault indicative thresholds are set for the principal components (PCs). If a particular PC exceeds its associated threshold, it is considered to be indicating a potential fault. The likelihood functions for each fault case are constructed on the basis of the probability that each PC would cross its respective threshold given that a particular fault category is present. The thresholds are determined by applying KDE to the data, where no known fault is present. Thresholds are set symmetrically on the lower and upper end at 2.5% and 97.5% of the cumulative density functions. The probability that a fault is F_i, given that a PC y_k has crossed its threshold is calculated using Equation 5.25, while the probability that an observation is classified as fault F_i is calculated in the same manner as in the case of alarm data, using Equation 5.26.

Decision level fusion

The decision-level fusion of the Bayesian framework is conducted according to Equation 5.40. After training the feature-level classifiers, a validation step is used to determine the confusion matrices for the alarm and process data (Equation 5.38). By doing so the algorithm is able to learn the probability of a particular data type classifying each type of fault correctly. These confusion matrices are then used to calculate the final prediction of the algorithm for the test sets by supposing independence between the prediction performance of alarm data and process data. If the alarm data predicted F_a and process data predicted F_p, then the probability that the final fault prediction is F_i is calculated as

$$P(F_i|F_a, F_p) = \frac{P(F_a|F_i)}{F_a} \cdot \frac{P(F_p|F_i)}{F_p} \cdot P(F_i) \tag{7.5}$$

The final prediction of the algorithm is the index of the Maximum a posteriori (MAP) fault class.

Training, validation, testing

A crucial point when implementing the method is to select the training, validation and test sets in an appropriate way. In practice, the amount of data under normal process operation is usually much greater than the amount of data which is available during periods of faulty operation. To obtain more precise threshold values, healthy data may be incorporated under various operating conditions in the analysis. Hence, the thresholds were trained using a training set that contained data from various normal operating conditions. The datasets for each fault case were split randomly, 60% of the data was selected to the training set, and 20-20% of the data were put into to the validation set and test set (as described in Section 5.7.1). The division of the data was applied separately for operating condition A and B.

In addition to the proper selection of the training set, validation set, and test set, the performance of the algorithm is also dependent on the prior distribution of faults, on the thresholds and on the number of principal components selected. All of these choices can highly influence the correct prediction rate of the algorithm. In the multiphase flow facility case study there was no prior information available about the typical distribution of the faults in the facility. Therefore, the priors of the fault cases were set uniformly for both the alarm and process data. The thresholds were set using 2σ rule. However, for other datasets, the optimal threshold value can vary, which may be solved by including an optimization step to evaluate the performance of the validation set at different threshold values and choose the threshold which gives the best performance. A way to measure the performance of the algorithm is to summarise the correct prediction rate of the confusion matrix of the process data. A similar approach can be applied to find the optimal number of principal components. The number of principal components was determined by taking the first $N = 9$, such that 99.9% of the total variance of the dataset was retained in the first N principal components.

7.5.2. Results

As previously noted, the dataset was randomly split to a training set, validation set, and test set. This random split was applied 100 times with the results from all 100 data divisions being averaged to generate a final result. Averaged results are shown below for the two operating conditions A and B. The process and alarm results of the classifiers are compared with the fused results.

Results are presented in the form of confusion matrices, where the columns represent the conditions diagnosed by the algorithm and the rows represent the actual conditions. The diagonal contains the correct classification values. For clarity the fault cases are listed again: F_0: Normal process operation, F_1:

Table 7.29: Alarm data, Operating condition A

		Diagnosed condition			
		F_0	F_1	F_2	F_3
Actual condition	F_0	0.41	0.59	0.00	0.00
	F_1	0.00	0.95	0.03	0.02
	F_2	0.06	0.21	0.73	0.00
	F_3	0.00	0.72	0.25	0.03

Table 7.30: Process data, Operating condition A

		Diagnosed condition			
		F_0	F_1	F_2	F_3
Actual condition	F_0	0.30	0.16	0.25	0.29
	F_1	0.03	0.97	0.00	0.00
	F_2	0.11	0.00	0.89	0.00
	F_3	0.03	0.00	0.00	0.97

Table 7.31: Fused data, Operating condition A

		Diagnosed condition			
		F_0	F_1	F_2	F_3
Actual condition	F_0	0.54	0.16	0.00	0.29
	F_1	0.03	0.97	0.01	0.00
	F_2	0.06	0.00	0.94	0.00
	F_3	0.02	0.00	0.01	0.97

Air blockage, F_2: Air leakage, F_3: Diverted flow. The F1-score was used to compare the performances of the feature-level fusion results and the decision-level fusion results for both operating conditions.

7.5.3. Discussion

Table 7.29 shows the results obtained after applying feature-level data fusion on the alarm data for operating condition A. The F1-score was 84.03%. This observation shows that in general, for operating condition A, the alarm data did not contain enough information to adequately classify the faults. This is primarily due to the limited number of alarms triggered during the case study. However, the algorithm correctly classified the air blockage fault F_1 in 95% of the cases. The false alarm rate was only 2%, but the 59% missed alarm rate was significantly higher. The misclassification between faults was also high, especially in the case of F_3 (diverted flow), where the misclassification rate was 97%.

Table 7.30 shows the results obtained after feature-level data fusion on the process data for operating condition A. The F1-score was 86.68%. The algorithm was correct in 97% of the cases for F_1 and F_3. The results also indicate a relatively high classification rate for F_2 with 89%. The false alarm rate was around 6%, but the missed alarm rate was even higher than for the alarms with a 70% rate.

Table 7.31 shows the results obtained after performing the decision-level data fusion, combining the results of both alarm and process data. The F1-score was 91.14% for the fused data for operating condition A. This is a 4.46% and 7.11% performance improvement compared to the F1-scores of the process and alarm data respectively. The false alarm rate was around 4%, while the missed alarm rate was 36%. The misclassification rate between faults was negligible. The correct classification rates were at least as good as the individual results obtained from the alarm and process data individually, and in many cases were improved.

Table 7.32 shows the results obtained after applying feature-level data fusion on the alarm data for operating condition B. The F1-score was 80.18%. These results are similar to the alarm data for operat-

Table 7.32: Alarm data, Operating condition B

		Diagnosed condition			
		F_0	F_1	F_2	F_3
Actual condition	F_0	0.20	0.02	0.18	0.59
	F_1	0.00	0.53	0.07	0.40
	F_2	0.07	0.18	0.21	0.54
	F_3	0.00	0.00	0.00	1.00

Table 7.33: Process data, Operating condition B

		Diagnosed condition			
		F_0	F_1	F_2	F_3
Actual condition	F_0	0.75	0.12	0.06	0.07
	F_1	0.04	0.96	0.00	0.00
	F_2	0.07	0.02	0.70	0.21
	F_3	0.05	0.00	0.07	0.88

Table 7.34: Fused data, Operating condition B

		Diagnosed condition			
		F_0	F_1	F_2	F_3
Actual condition	F_0	0.85	0.02	0.08	0.05
	F_1	0.03	0.97	0.00	0.00
	F_2	0.07	0.02	0.88	0.03
	F_3	0.05	0.00	0.07	0.88

ing condition A, again indicating that the alarm data did not contain enough information to adequately classify the faults. The false alarm rate was around 2% and the missed alarm rate is 80%. The results indicate a 100% correct classification rate for F_3 (diverted flow), while there is a high misclassification rate of fault for F_1 and F_2.

Table 7.33 shows the results obtained after applying feature-level data fusion on the process data for operating condition B. The F1-score was 92.53%. The algorithm was correct in 96% of the cases for F_1 and had a relatively high classification rate of 88% for F_3. The false alarm rate was around 5% and the missed alarm rate was 25%. The fusion with the process data performed better compared for operating condition A with regards to lower false and missed alarm rates, however, the misclassification between faults increased especially for F_2 with 23%.

The F1-score was 94.79% after performing the decision-level data fusion for operating condition B as shown in Table 7.34. This is a 2.26% improvement compared to the F1-score of the process data and a 14.61% improvement compared to the F1-score of the alarm data. The false alarm rate was around 4%, while the missed alarm rate was 15%. The misclassification rate between faults was also decreased relative to the results of alarm and process data. The correct classification rates were better than or as good as the results of the alarm and process data, except for the case of F_3 in the results for the alarm data. Although the alarm data was 100% accurate in classifying F_3, the algorithm also classified F_0, F_1 and F_2 in almost half of the cases as F_3, which reduced the confidence in the alarm data for F_3 at the global fusion stage.

The fused results improved the performance of the algorithm under both operating conditions. The alarm data was not reliable for fault classification for neither of the operating conditions, while the process data gave a better insight. The fused results show that even in the case of insufficient alarm data the method is able to improve the performance of the fault classification when compared to using only the process data.

The advantage of the method that besides alarm and process data, other types of data may be included in the framework. If one data type is not available, the framework would still be able to decide about the health state of the process based on the available data types. Hence, the modularity and the scalability of the framework can be exploited for process monitoring applications, where some data types are not available continuously. The framework also showed its modularity with regards to the aspect that pre-processing and feature-level fusion may differ by data types.

In the current form of the framework, only steady state diagnostics are possible. In case of alarm floods, the pattern of the alarm sequence would not be taken into account, as the framework decides about the health state only on the basis of the current observation. This aspect could be further investigated in the future and the framework could be extended for accounting for process dynamics and alarm floods.

7.5.4. Summary

In this section, a two-stage Bayesian framework for fusing alarm and process data for fault detection and classification for process monitoring purposes was applied. Alarm and process data are fused independently on the feature-level using NB classifiers. The decision-level stage of the method fuses the feature-level classification results of the alarm and process data, providing a final classification result. Fusing both process and alarm data together on the decision-level was shown to improve the performance of the algorithm at correctly classifying faults relative to considering each data type independently. Under two different operating conditions, it was shown that fusing both process and alarm data together was able to improve the classification performance of the algorithm. The results showed that even in the case of only a few alarm events occurring, the proposed method for process and alarm fusion is able to improve the performance of the fault classification relative to the case of only considering process data.

7.6. Bayesian feature-level fusion with IKDE

In this section, the IKDE method proposed in Section 5.5.4 is applied and compared with the KDE with respect to their computational times and storage requirements for Bayesian feature-level fusion and fault diagnostics (Stief et al., 2019a). The comparison considers the induction motor case study described in Section 6.1. In online condition monitoring and protection applications, the computational times can be of high relevance, especially when the sensors are providing measurements at high sampling rates. The two methods are compared with respect to their storage requirements, training and testing times.

7.6.1. Implementation of the method

The processing hardware was an Intel® Core™ i5-4300U, 1.9 GHz. 120 features were extracted from the vibration signals containing in total 3480 observations for 8 different health conditions, as described in Section 7.1.1. The dataset was randomly split into 70 % training set containing 2436 observations and 30 % test set containing 1044 observations. Two Naive Bayes classifiers were trained on the training set, one using KDE for calculating the likelihood functions as described in Section 5.5.3 and one using IKDE for calculating the likelihood functions as described in Section 5.5.4. A KDE function was implemented for each feature within each fault class, resulting in 960 KDE functions being saved after training. The two methods were tested on the test set and the elapsed training and testing times were measured and saved for both methods.

7.6.2. Results

The obtained results for KDE and IKDE are presented in Table 7.35. There is an order of magnitude difference between the training times of the two algorithms. IKDE is one order of magnitude slower than KDE. When it comes to testing an observation, KDE is an order of magnitude slower than IKDE. The results clearly underline the superiority of IKDE when testing an observation.

Table 7.35: Results of comparison between KDE and IKDE.

Base of comparison	KDE	IKDE
Training time [s]	3.1	43.1
Testing time [s]	660.2	52.9
Testing time per observation [s]	0.63	0.05
Storage after training (KB)	4081	3048

7.6.3. Discussion

The superiority of IKDE over KDE with regards to computational time comes from the fact that the interpolation order N is much lower than the number of samples in training set n. For the presented test a standard approach was applied using *chebfun* (i.e. conversion of interpolating polynomial to the finite series in orthogonal Chebyshev basis, and determining the interpolation order, for which series coefficients are indistinguishable under machine precision).

Because IKDE is in the early stages of development there are multiple areas for further consideration, in particular,

– IKDE may be further explored for condition monitoring applications when the online processing time is the bottleneck for implementing more complex diagnostics algorithms. The method may be suitable for other applications where data streams have to be processed very fast.

– The currently proposed adaptive interval expansion strategy may be further developed to account for probability distributions with longer tails, which may suffer from scaling issues.

– If many outliers are present in the training dataset, the probability distribution will be multimodal with numerous regions of flatness with zero or constant values in the distribution. This will result in an increased number of Chebyshev nodes, higher computational times and increased storage space. Additionally, there is a potential source for numerical errors, as the regions of flatness correspond to multiple real zeros of the derivative of the interpolating polynomial. Future research will focus on exploring solutions for outlier removal or detecting regions of flatness in order to split interpolation intervals.

– All of the calculations were done in Matlab®using the open source software package *Chebfun*. Step 1,2 and 4 of the training of IKDE shown in Figure 5.2 may be run in parallel, therefore parallel computing could further improve the training times of IKDE.

– The advantages of reduced data storage and computational time offered by IKDE would be even more pronounces when used with larger training datasets. IKDE only stores two scaling factors and three vectors containing coefficients for the Barycentric formula (and practically only two) compared to KDE, which stores all of the training set and the bandwidth value for any given calculation of a posterior probability. The number of operations needed for the calculation a posterior probability does not depend on the size of the training set, only on the number of Chebyshev nodes.

– A drawback of IKDE is the difficulty in incorporating new data. While KDE can be modified in order to incorporate new samples, in the case of IKDE, a new training procedure has to be performed.

7.6.4. Summary

In this section, the newly proposed Interpolated Kernel Density Estimator was applied to approximate the kernel density estimator for feature-level fusion in the Bayesian framework. IKDE was tested and compared to the original implementation of KDE on the induction motor case study. Compared to KDE, IKDE was able to reduce the testing time by one order of magnitude, however, the time required for training was one order of magnitude higher. IKDE may be a computationally efficient solution for applications, where the testing time per observation and storage resource are limited and the training time is not critical.

7.7. Summary of applications

In this chapter, the newly developed feature selection methods (Section 4) and Bayesian data fusion methods (Section 5) have been tested and validated on a component-level monitoring case study of induction motors and on a plant-level monitoring case study of a multiphase flow facility.

The proposed feature selection methods have been applied to both of the case studies. The results confirmed that ReliefF with the newly developed correlation removal proposed in Section 4.5.1 is applicable for condition monitoring problems where reduced and uncorrelated feature sets may further improve the feature design. The new hybrid filter-wrapper approach proposed in Section 4.5.2 was able to provide an appropriate selection from the ranked features. Feature selection was able to significantly reduce the number of necessary features for the analysis, while achieving high classifications results. The relevance of feature selection for condition monitoring was further highlighted by observing how the relevant features differed for various fault detection and fault severity diagnosis problems. Hence, feature design based on domain knowledge can be efficiently complimented with ReliefF-based feature selection, which is able to pinpoint the most relevant features by signal processing type, sensor type and sensor location for various monitoring problems.

All of the previously introduced Bayesian feature-level fusion methods described in Section 5.5 have been used on either one of the case studies. The obtained results showed that they are capable of achieving relatively high classifications performance even on the feature-level fusion stage, however, decision-level fusion was able to improve their performance in all of the observed scenarios. Furthermore, this chapter has also highlighted some of their weaknesses, such as the computation demand of KDE-based NB classifiers, which has been efficiently solved with the new IKDE approach proposed in Section 5.5.4.

The proposed two-stage Bayesian framework has been applied to both of the case studies. The results indicated that the proposed method can increase the reliability and efficiency of fault detection and diagnosis compared to a single stage feature-level fusion approach. Furthermore, the results confirmed that heterogeneous data fusion using the two-stage Bayesian framework is both applicable for a component-level monitoring case when various sensor data are fused and also applicable to a plan-level monitoring case, where process and alarm data are fused. The fusion achieved lower false and missed alarm rates compared to feature-level fusion applied to individual sensor or data types. The newly proposed framework also provided a structured approach for comparing the performance of different signal and data types in diagnosing various faults.

8. Conclusion

The goal of this thesis was to investigate how combining disparate data may be used to support condition monitoring of assets from the component level to the plant level. After a review of the field of condition monitoring in Chapter 2, it was confirmed that there are many potential sources of data that may be used for condition monitoring purposes. However, the data available is disparate, originating from various sources, therefore new data fusion methods are needed to tackle the issue of heterogeneity. Chapter 3 gave a comprehensive review of data fusion methods, their types, levels, and applications in condition monitoring. The abundance of condition monitoring data from heterogeneous sources holds several advantages for condition monitoring applications, such as improved accuracy, effective distinction between faults and improved robustness of the CM system. Chapter 4 explored methods for find relevant features and sensors which hold informative data for condition monitoring problems through feature design and feature selection. The application of feature selection methods to condition monitoring problems were also discussed, as well as their application with correlated heterogeneous data. Two methods have been proposed to solve challenges often encountered during feature selection. Chapter 5 proposed a two-stage Bayesian framework for diagnostics, which is highly relevant for condition monitoring applications with heterogeneous data. All of the methods proposed in the thesis were tested and validated using experimental data from two case studies, which were described in Chapter 6. The application results of the methods and a discussion on their applicability for component and plant-level monitoring were provided in Chapter 7. The proposed methods have been shown to improve the results of diagnostics, whilst making the monitoring systems robust, modular and scalable.

This chapter gives a summary of the achievements and contributions of the thesis, compares them with the initial research goals and indicates future research directions. The main contributions of the thesis are as follows:

- Identification of algorithm requirements with reference to the data types typically available in an industrial setting (Page 27 and Page 39). After the review of data sources in condition monitoring systems and data fusion techniques, a set of requirements were determined that condition monitoring systems have to fulfil to adapt to the modern industrial environment where data is available from disparate sources.

- Formulation of a novel two-stage Bayesian data fusion framework based on these requirements. The proposed method is modular, scalable and transparent, so the end user may trace back the result to the root cause. The framework achieves robustness and deals with uncertainty in measurement quality using the Bayesian formulation on the feature-level and on the decision-level. The proposed method is applicable for both component and plant-level monitoring problems, which highlights its generality and transferability (Page 64).

- Formulation of feature-level fusion methods within the two-stage Bayesian data fusion framework with regards to the properties of the available data (Page 65). The selection of the feature-level fusion method may depend on what prior knowledge available about the distributions of data and on the level of dependency with respect to changing operating conditions.

- Formulation of a decision-level fusion method within the two-stage Bayesian data fusion framework (Page 72). The decision-level fusion method provides transparency for the end user, who may trace back the final diagnostic results to the root cause of the decision.

- Application of the two-stage Bayesian data fusion framework for sensor fusion for induction motor fault diagnostics, where the improved performance of the two-stage fusion relative to single stage feature-level fusion or no fusion fusion has been demonstrated (Page 108 and Page 117).

- Application of the two-stage Bayesian data fusion framework for heterogeneous alarm and process data fusion for process condition diagnostics of a multiphase flow facility, where the two different data types were collected using different data acquisition methods. The superiority of the two-stage fusion in performance has been shown compared to using only one data type (Page 126).

- Identification of a feature selection method which is applicable for condition monitoring and data fusion applications: ReliefF as an efficient feature selection filter with the flaw of not taking into account the correlation of features (Page 46).

- Novel formulation of ReliefF for producing relevant and less correlated feature sets with a new Pearson's linear correlation coefficient based re-ranking approach (Page 48).

- Novel formulation of ReliefF for feature selection with a hybrid filter-wrapper approach, in order to provide an automated solution for feature selection, achieving optimal feature sets (Page 51).

- Application of a feature selection approach for induction motor fault diagnostics problem, showing that feature selection can contribute to finding the features which are best suited for diagnosing specific faults (Page 91). Such information can support the creation of robust monitoring systems. Furthermore, the application of the newly proposed ReliefF with correlation removal proved to be efficient for removing the correlation between features. ReliefF with correlation removal also achieved improved fault diagnosis accuracy compared to the standard implementation of ReliefF.

- Application of a feature selection approach for process condition diagnostics of a multiphase flow facility with the conclusion that feature selection can not only help in identifying relevant features but also in indicating the relevant sensors for monitoring. It has been highlighted that whilst using irrelevant features does not necessarily result in performance degradation in the case of a well-parametrized fault classifier, sensor failures can have a significant influence on monitoring performance, even in case of failure of seemingly irrelevant sensors (Page 91). The results obtained with the newly proposed hybrid filter-wrapper approach confirm that the approach is able to provide an appropriate selection from the ranked features. The selected feature set are significantly reduced in terms of dimensions compared to original feature set and accurate classifications results may be obtained with them.

- The novel formulation of Interpolated Kernel Density Estimate, which makes the powerful and non-parametric KDE method suitable for online FDD problems where reduced computational costs are beneficial. IKDE may be used for feature-level fusion in the two-stage data fusion framework. IKDE is able to achieve at least one order of magnitude reduction in the elapsed CPU computing time compared to KDE (Page 99).

- Application of the newly proposed Interpolated Kernel Density Estimate for feature-level fusion within a Naive Bayes classifier for fault diagnosis of induction motors. IKDE required an order of magnitude less testing time for each observation, which makes it suitable for application in online condition monitoring systems (Page 133).

- Creation of a novel heterogeneous dataset from a multiphase flow facility, which is suitable for developing and validating algorithms for fault detection and diagnosis and heterogeneous data fusion concepts. The dataset is publicly available (Page 83). The work was done in collaboration with Ruomu Tan.

The research conducted in this thesis also identified a set of new research directions which may be further investigated based on the limitations of the newly proposed methods.

Firstly, the limitation of ReliefF for condition monitoring applications is that in the case of using data with environmental or operating conditions not present in the training set, there is no guarantee that the selected features are still the most relevant. Future research may focus on investigating how to account for various environmental or operating conditions during the feature ranking and selection process. As shown in Section 7.2, feature selection with ReliefF is not only suitable for selecting the most relevant features. In the case of systems with many sensors, ReliefF may also provide information which type of sensor is the most informative or which sensor location along the monitored asset is the most optimal. Hence, feature selection may be applied in a framework for designing data acquisition systems with optimal sensor selection and placement. Furthermore, ReliefF with correlation removal may be combined with the hybrid filter-wrapper feature selection approach to produce relevant and optimal feature sets. The combined feature selection method may be further improved to include a required level of robustness in the condition monitoring system as a requirement of the selection. The number of features included in the analysis may be a trade-off between redundancy, observability, and robustness, which may be implemented by the formulation of the feature selection as a multi-objective optimization problem.

Secondly, a limitation of the two-stage Bayesian data fusion framework is that, in its present form, it is only suitable for steady-state signals. As a data-driven method, the accuracy of the likelihood functions used in the approach will be dependent on the availability of large quantities of comparable measurement data. The framework is not able to classify faults according to faults severities, unless the training data are explicitly labelled according to severities. The two-stage Bayesian framework may be further developed to account for this issue. Further improvements in the framework may be achieved by better specifying the prior probabilities, both on the feature- and on the decision-level. Prior probabilities may be obtained on the basis of historical fleet data as described in Section 2.2.5 which would represent the inclusion of an additional source of information in the analysis. The two-stage Bayesian data fusion framework may be further developed to take into account the phenomenon of alarm floods to better diagnose faults when fusing alarm and process data. Future research may also focus on the integration of feature-selection in the two-stage Bayesian data fusion framework to exploit the strengths of a modular fault detection approach which utilizes an optimized number of sensors and features for each monitoring sub-problem. The two-stage Bayesian data fusion framework may also be extended for prognostics by dynamically projecting the posterior probabilities of faults into the future based on the historical condition monitoring data.

Thirdly, as IKDE is in the early stages of development there are multiple areas for further consideration, in particular further improving the interval expansion strategy to account for probability distributions with longer tails and developing outlier removal methods or detecting regions of flatness in order to split interpolation intervals. A drawback of IKDE is the difficulty in incorporating new data, as a new training procedure has to be performed. This is an interesting challenge that may be solved in the future. IKDE may also be further explored for online condition monitoring applications with larger datasets.

To conclude, the research conducted in this thesis proposed new methods for performing feature selection from high dimensional datasets, developed data fusion methods for fault classification using multivariate statistics and Bayesian reasoning and developed methods for fusing disparate data types with respect to data typically available in an industrial setting. The methods were tested on datasets relevant both for component-level and plant-level condition monitoring. The results confirmed that the methods improved the diagnostics performance, while creating a robust, modular and scalable monitoring frameworks.

A. Appendix

A.1. Publications

The work in the thesis has led to three published journal papers and four published conference papers. A further conference paper has been submitted for consideration. Furthermore, the work in the thesis has resulted in collaborations with other researchers of similar fields resulting in one submitted journal paper and a published conference paper. These publications are summarized below in chronological order.

1. **Anna Stief**, James R. Ottewill, Michal Orkisz, and Jerzy Baranowski. Two stage data fusion of acoustic, electric and vibration signals for diagnosing faults in induction motors. *Elektronika ir Elektrotechnika*, 23(6):19–24, 2017. DOI: 10.5755/j01.eie.23.6.19690

2. **Anna Stief**, James R. Ottewill, and Jerzy Baranowski. ReliefF-based feature ranking and feature selection for monitoring induction motors. In *2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR)*. IEEE, 2018. DOI: 10.1109/MMAR.2018.8486097

3. **Anna Stief**, James R. Ottewill, Ruomu Tan, and Yi Cao. Process and alarm data integration under a two-stage Bayesian framework for fault diagnostics. *IFAC-PapersOnLine*, 51(24):1220–1226, 2018. DOI: 10.1016/j.ifacol.2018.09.696

4. **Anna Stief**, Ruomu Tan, Yi Cao, and James R. Ottewill. Analytics of heterogeneous process data: Multiphase flow facility case study. *IFAC-PapersOnLine*, 51(18):363–368, 2018. DOI: 10.1016/j.ifacol.2018.09.327

5. **Anna Stief**, James R. Ottewill, Jerzy Baranowski, and Michal Orkisz. A PCA - two stage Bayesian sensor fusion approach for diagnosing electrical and mechanical faults in induction motors. *IEEE Transactions on Industrial Electronics*, 2019. DOI: 10.1109/TIE.2019.2891453

6. **Anna Stief**, James R. Ottewill, and Jerzy Baranowski. Investigation of the diagnostic properties of sensors and features in a multiphase flow facility case study. *IFAC-PapersOnLine*, 2019.

7. Matthieu Lucke, Xueyu Mei, **Anna Stief**, Moncef Chioua, and Nina F. Thornhill. Variable selection for fault detection and identification based on mutual information of multi-valued alarm series. *IFAC-PapersOnLine*, 2019

8. **Anna Stief**, Ruomu Tan, Yi Cao, James R. Ottewill, Jerzy Baranowski, and Nina F. Thornhill. A heterogeneous benchmark dataset for data analytics: Multiphase flow facility case study. *Journal of Process Control*, 79: 41–55, 2019. DOI: 10.1016/j.jprocont.2019.04.009

9. **Anna Stief**, James R. Ottewill, Jerzy Baranowski. Fault diagnosis using interpolated kernel density estimate. *58th Conference on Decision and Control*, 2019. *[submitted]*

10. Matthieu Lucke, **Anna Stief**, Moncef Chioua, James R. Ottewill, and Nina F. Thornhill. Bayesian fault detection and identification combining process measurements and alarm series. *Control Engineering Practice*, 2019. *[submitted]*

A.2. The PRONTO project

Optimal and sustainable operation of assets over the typical 30-50 years lifetime of a process plant requires novel approaches for managing information and resources. Enhancing the operation and optimization of process plants lies in the efficient management of the large installed asset base in existing plants, in the design of optimal maintenance strategies and lastly in the development of new state-of-the-art process plants. As process plants generate a large amount of heterogeneous data available from different sub-systems, this data will be the baseline for the development of novel technologies (PRONTO Consortium, 2015).

PRONTO is a 3-year European Industrial Doctorate program funded by the Marie Skłodowska-Curie Actions scheme under EU Horizon 2020. PRONTO focuses on process network optimization for the efficient and sustainable operation of Europe's process industries, taking machinery condition and process performance into account. The consortium partners include companies with high reputations for innovation such as ABB, Equinor, INEOS, BASF, AST, and leading universities such as Imperial College London, Carnegie Mellon University, Cranfield University, AGH University of Science and Technology, Norwegian University of Science and Technology, University of Valladolid and Technical University of Dortmund. The PRONTO project offers the early stage researchers training under the European Industrial Doctorate in joint supervision of the doctoral training with a strong emphasis on industrially-relevant Ph.D. projects leading to practical demonstrations.

The research topics in the PRONTO project are grouped into three areas. (1) Data analytics for assessment of the condition and performance of networks of process industry production equipment. (2) Optimization of resource use in process networks, taking account of real-time information about the condition and performance of the process equipment. (3) New concepts for process operation identified as having a high potential for impact by industrial partners.

This work contributed to the PRONTO project under the first topic of data analytics. My assigned research topic as ESR-C was "Combining data from disparate sources for condition monitoring purposes". My research objective was to create new methods to combine quantitative and qualitative data recorded online, offline, and periodically in an automated way for equipment condition monitoring to enable more reliable and robust condition assessment by incorporating data from a greater number of diverse sources (PRONTO Consortium, 2015).

Bibliography

ABB. ABB Ability Smart Sensor for mounted bearings. `https://new.abb.com/mechanical-power-transmission/mounted-bearings/smart-sensor-for-mounted-bearings`, 2018. Accessed: 2019-02-04.

ABB. Abb Ability^{TM} System 800xA. `https://new.abb.com/control-systems/system-800xa`, 2019a. Accessed: 2019-05-07.

ABB. Introducing ABB Ability Symphony Plus. `https://new.abb.com/power-generation/systems/power-plant-automation/abb-ability-symphony-plus/what-is-abb-ability-symphony-plus`, 2019b. Accessed: 2019-03-08.

B. M. Abbagoni and H. Yeung. Non-invasive classification of gas–liquid two-phase horizontal flow regimes using an ultrasonic Doppler sensor and a neural network. *Measurement Science and Technology*, 27(8):084002, 2016.

B. Al-Najjar and I. Alsyouf. Enhancing a company's profitability and competitiveness using integrated vibration-based maintenance: A case study. *European Journal of Operational Research*, 157(3):643–657, 2004.

P. Albrecht, J. Appiarius, R. McCoy, E. Owen, and D. Sharma. Assessment of the reliability of motors in utility applications–Updated. *IEEE Transactions on Energy conversion*, (1):39–46, 1986.

H. Alkhadafe, A. Al-Habaibeh, and A. Lotfi. Condition monitoring of helical gears using automated selection of features and sensors. *Measurement*, 93:164–177, 2016.

J. C. Ang, A. Mirzal, H. Haron, and H. N. A. Hamed. Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 13(5):971–989, 2016.

F. Anifowose, A. Khoukhi, and A. Abdulraheem. Investigating the effect of training–testing data stratification on the performance of soft computing techniques: an experimental study. *Journal of Experimental & Theoretical Artificial Intelligence*, 29(3):517–535, 2017.

F. Arellano-Espitia, J. J. Saucedo-Dorantes, R. A. Osornio-Rios, M. Delgado-Prieto, J. A. Cariño-Corrales, and R. J. Romero-Troncoso. Statistical data fusion as diagnosis scheme applied to a kinematic chain. In *2018 IEEE International Conference on Industrial Technology (ICIT)*, pages 2111–2118. IEEE, 2018.

D. Asfani, M. Purnomo, and D. Sawitri. Naïve Bayes classifier for temporary short circuit fault detection in stator winding. In *2013 9th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives (SDEMPED)*, pages 288–294. IEEE, 2013.

M. Askarian, G. Escudero, M. Graells, R. Zarghami, F. Jalali-Farahani, and N. Mostoufi. Fault diagnosis of chemical processes with incomplete observations: A comparative study. *Computers & Chemical Engineering*, 84:104–116, 2016.

S. Bagavathiappan, B. B. Lahiri, T. Saravanan, J. Philip, and T. Jayakumar. Infrared thermography for condition monitoring–A review. *Infrared Physics & Technology*, 60:35–55, 2013.

J. Baranowski, P. Bania, I. Prasad, and T. Cong. Bayesian fault detection and isolation using Field Kalman Filter. *EURASIP Journal on Advances in Signal Processing*, 2017(1):79, 2017.

J. A. Barnett. Computational methods for a mathematical theory of evidence. In *Classic Works of the Dempster-Shafer Theory of Belief Functions*, pages 197–216. Springer, 2008.

V. Barnett and T. Lewis. *Outliers in statistical data*. Wiley, 1974.

W. Batko, M. Kłaczyński, J. Wierzbicki, T. Wszołek, and J. Frączek. Vibroacoustic analysis of defect electric motors, 2011. Internal ABB Technical report.

M. Bauer and N. F. Thornhill. A practical method for identifying the propagation path of plant-wide disturbances. *Journal of Process Control*, 18(7-8):707–719, 2008.

T. Bayes, R. Price, and J. Canton. An essay towards solving a problem in the doctrine of chances. 1763.

E. Bechhoefer and M. Kingsley. A review of time synchronous average algorithms. In *Annual Conference of the Prognostics and Health Management Society, San Diego, CA, Sept*, pages 24–33, 2009.

M. E. H. Benbouzid. A review of induction motors signature analysis as a medium for faults detection. *IEEE Transactions on Industrial Electronics*, 47(5):984–993, 2000.

J.-P. Berrut and L. N. Trefethen. Barycentric Lagrange Interpolation. *SIAM Review*, 46(3):501–517, 2004.

T. Biagetti and E. Sciubba. Automatic diagnostics and prognostics of energy conversion processes via knowledge-based systems. *Energy*, 29(12-15):2553–2572, 2004.

A. Bodensohn, M. Haueis, R. Mäckel, M. Pulvermüller, and T. Schreiber. System monitoring for lifetime prediction in automotive industry. In *Advanced Microsystems for Automotive Applications 2005*, pages 149–158. Springer, 2005.

V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos. A review of feature selection methods on synthetic data. *Knowledge and Information Systems*, 34(3):483–519, 2013.

M. Bonfe, P. Castaldi, W. Geri, and S. Simani. Fault detection and isolation for on-board sensors of a general aviation aircraft. *International Journal of Adaptive Control and Signal Processing*, 20(8): 381–408, 2006.

M. Borunda, O. A. Jaramillo, A. Reyes, and P. H. Ibargüengoytia. Bayesian networks in renewable energy systems: A bibliographical survey. *Renewable and Sustainable Energy Reviews*, 62:32–45, 2016.

H. Boström, S. F. Andler, M. Brohede, R. Johansson, A. Karlsson, J. Van Laere, L. Niklasson, M. Nilsson, A. Persson, and T. Ziemke. On the definition of information fusion as a field of research. Institutionen för Kommunikation och Information, 2007.

A. W. Bowman and A. Azzalini. *Applied smoothing techniques for data analysis: the kernel approach with S–Plus illustrations*, volume 18. OUP Oxford, 1997.

S. Braun. *Discover signal processing: An interactive guide for engineers*. John Wiley & Sons, 2008.

K. Brun and M. G. Nored. Guideline for field testing of gas turbine and centrifugal compressor performance. *Gas Machinery Research Council/Southwest Research Institute*, pages 1–93, 2006.

BS EN 13306:2010. Maintenance — maintenance terminology. Standard, British Standard, 2010.

J. Cai, J. Luo, S. Wang, and S. Yang. Feature selection in machine learning: A new perspective. *Neurocomputing*, 300:70–79, 2018.

F. Caliskan and C. Hajiyev. A review of in-flight detection and identification of aircraft icing and reconfigurable control. *Progress in Aerospace Sciences*, 60:12–34, 2013.

Y. Cao, H. He, and H. Man. SOMKE: Kernel density estimation over data streams by sequences of self-organizing maps. *IEEE Transactions on Neural networks and Learning Systems*, 23(8):1254–1268, 2012.

E. P. Carden and P. Fanning. Vibration based condition monitoring: A review. *Structural Health Monitoring*, 3(4):355–377, 2004.

M. Cerrada, R. V. Sánchez, D. Cabrera, G. Zurita, and C. Li. Multi-stage feature selection by using genetic algorithms for fault diagnosis in gearboxes based on vibration signal. *Sensors*, 15(9):23903–23926, 2015.

G. Chandrashekar and F. Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

S. Charbonnier, N. Bouchair, and P. Gayet. Fault template extraction to assist operators during industrial alarm floods. *Engineering Applications of Artificial Intelligence*, 50:32–44, 2016.

C. Chen, B. Zhang, and G. Vachtsevanos. Prediction of machine health condition using neuro-fuzzy and Bayesian algorithms. *IEEE Transactions on Instrumentation and Measurement*, 61(2):297–306, 2012.

Q. Chen, P. Goulding, D. Sandoz, and R. Wynne. The application of kernel density estimates to condition monitoring for process industries. In *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No. 98CH36207)*, volume 6, pages 3312–3316. IEEE, 1998.

L. H. Chiang and R. D. Braatz. Process monitoring using causal map and multivariate statistics: fault detection and identification. *Chemometrics and Intelligent Laboratory Systems*, 65(2):159–178, 2003.

L. H. Chiang, E. L. Russell, and R. D. Braatz. *Fault detection and diagnosis in industrial systems*. Springer Science & Business Media, 2000.

S. Cho, S. Binsaeid, and S. Asfour. Design of multisensor fusion-based tool condition monitoring system in end milling. *The International Journal of Advanced Manufacturing Technology*, 46(5-8):681–694, 2010.

K. Choi, S. Singh, A. Kodali, K. R. Pattipati, J. W. Sheppard, S. M. Namburu, S. Chigusa, D. V. Prokhorov, and L. Qiao. Novel classifier fusion approaches for fault diagnosis in automotive systems. *IEEE Transactions on Instrumentation and Measurement*, 58(3):602–611, 2009.

S. W. Choi, C. Lee, J.-M. Lee, J. H. Park, and I.-B. Lee. Fault detection and identification of nonlinear processes based on kernel PCA. *Chemometrics and Intelligent Laboratory Systems*, 75(1):55–67, 2005.

B. R. Cobb and P. P. Shenoy. A comparison of bayesian and belief function reasoning. *Information Systems Frontiers*, 5(4):345–358, 2003.

D. R. Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232, 1958.

A. Czajkowski and K. Patan. Design of predictive fault tolerant control by the means of state space neural networks. In *Control and Automation (MED), 2016 24th Mediterranean Conference on*, pages 201–206. IEEE, 2016.

A. M. Da Silva, R. J. Povinelli, and N. A. Demerdash. Induction machine broken bar and stator short-circuit fault diagnostics based on three-phase stator current envelopes. *IEEE Transactions on Industrial Electronics*, 55(3):1310–1318, 2008.

P. R. N. da Silva, H. A. Gabbar, P. V. Junior, and C. T. da Costa Junior. A new methodology for multiple incipient fault diagnosis in transmission lines using QTA and Naive Bayes classifier. *International Journal of Electrical Power & Energy Systems*, 103:326–346, 2018.

B. V. Dasarathy. Sensor fusion potential exploitation-innovative architectures and illustrative applications. *Proceedings of the IEEE*, 85(1):24–38, 1997.

S. Datta and S. Sarkar. A review on different pipeline fault detection methods. *Journal of Loss Prevention in the Process Industries*, 41:97–106, 2016.

H. de Faria Jr, J. G. S. Costa, and J. L. M. Olivas. A review of monitoring methods for predictive maintenance of electric power transformers based on dissolved gas analysis. *Renewable and Sustainable Energy Reviews*, 46:201–209, 2015.

A. P. Dempster. A generalization of Bayesian inference. *Journal of the Royal Statistical Society: Series B (Methodological)*, 30(2):205–232, 1968.

Z. Deng, F.-L. Chung, and S. Wang. Robust Relief-feature weighting, margin maximization, and fuzzy optimization. *IEEE Transactions on Fuzzy Systems*, 18(4):726–744, 2010.

J. Deutsch and D. He. Using deep learning-based approach to predict remaining useful life of rotating components. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(1):11–20, 2018.

S. S. Dhami and B. Pabla. Gear fault classification using vibration and acoustic sensor fusion: A case study. In *2018 Condition Monitoring and Diagnosis (CMD)*, pages 1–6. IEEE, 2018.

Y. Ding, F. Wang, Z.-y. Wang, and W.-j. Zhang. Fault diagnosis for hydraulic servo system using compressed random subspace based ReliefF. *Complexity*, 2018, 2018.

W. E. Dixon, I. D. Walker, D. M. Dawson, and J. P. Hartranft. Fault detection for robot manipulators with parametric uncertainty: A prediction-error-based approach. *IEEE Transactions on Robotics and Automation*, 16(6):689–699, 2000.

P. Domingos and M. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine learning*, 29(2-3):103–130, 1997.

E. B. dos Santos, N. F. Ebecken, E. R. Hruschka Jr, A. Elkamel, and C. M. Madhuranthakam. Bayesian classifiers applied to the Tennessee Eastman process. *Risk Analysis*, 34(3):485–497, 2014.

D. Dou, B. Xue, M. He, and J. Jiang. An intelligent fault diagnosis method for rotating machinery based on genetic algorithm and classifier ensemble. In *Control And Decision Conference (CCDC), 2017 29th Chinese*, pages 4178–4181. IEEE, 2017.

T. A. Driscoll, N. Hale, and L. N. Trefethen. Chebfun guide. Pafnuty Publications, Oxford, 2014.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley & Sons, 2012.

B. J. Emran and H. Najjaran. A review of quadrotor: An underactuated mechanical system. *Annual Reviews in Control*, 46:165–180, 2018.

S. J. Engel, B. J. Gilmartin, K. Bongort, and A. Hess. Prognostics, the real issues involved with predicting life remaining. In *Aerospace Conference Proceedings, 2000 IEEE*, volume 6, pages 457–469. IEEE, 2000.

J. Esteban, A. Starr, R. Willetts, P. Hannah, and P. Bryanston-Cross. A review of data fusion models and architectures: towards engineering guidelines. *Neural Computing & Applications*, 14(4):273–281, 2005.

B. Everitt and A. Skrondal. *The Cambridge Dictionary of Statistic*. Cambridge University Press New York, 2010.

M. Farajzadeh-Zanjani, R. Razavi-Far, and M. Saif. Dimensionality reduction-based diagnosis of bearing defects in induction motors. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2539–2544. IEEE, 2017.

C. R. Farrar and K. Worden. An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851):303–315, 2006.

D. M. Frangopol, J. S. Kong, and E. S. Gharaibeh. Reliability-based life-cycle management of highway bridges. *Journal of Computing in Civil Engineering*, 15(1):27–34, 2001.

N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine learning*, 29(2-3): 131–163, 1997.

D. Galar, M. Palo, A. Van Horenbeek, and L. Pintelon. Integration of disparate data sources to perform maintenance prognosis and optimal decision making. *Insight - Non-destructive Testing and Condition Monitoring*, 54(8):440–445, 2012.

A. Gelman, H. S. Stern, J. B. Carlin, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 2013.

A. Geraci, F. Katki, L. McMonegal, B. Meyer, J. Lane, P. Wilson, J. Radatz, M. Yee, H. Porteous, and F. Springsteel. *IEEE standard computer dictionary: Compilation of IEEE standard computer glossaries*. IEEE Press, 1991.

K. Ghosh, Y. S. Ng, and R. Srinivasan. Evaluation of decision fusion strategies for effective collaboration among heterogeneous fault diagnostic methods. *Computers & Chemical Engineering*, 35(2):342–355, 2011.

K. Gillen, M. Celina, and R. Clough. Density measurements as a condition monitoring approach for following the aging of nuclear power plant cable materials. *Radiation Physics and Chemistry*, 56(4): 429–447, 1999.

C. Gini. Variabilità e mutabilità. *Reprinted in Memorie di metodologica statistica (Ed. Pizetti E, Salvemini, T). Rome: Libreria Eredi Virgilio Veschi*, 1912.

S. A. Glantz. *Primer of biostatistics*. 1976.

A. Glowacz. Acoustic based fault diagnosis of three-phase induction motor. *Applied Acoustics*, 137: 82–89, 2018.

A. Glowacz. Fault diagnosis of single-phase induction motor based on acoustic signals. *Mechanical Systems and Signal Processing*, 117:65–80, 2019.

S. Goldman. *Vibration spectrum analysis: a practical approach*. Industrial Press Inc., 1999.

D. Gorinevsky. Fault isolation in data-driven multivariate process monitoring. *IEEE Transactions on Control Systems Technology*, 23(5):1840–1852, 2015.

D. Goyal and B. Pabla. Condition based maintenance of machine tools—a review. *CIRP Journal of Manufacturing Science and Technology*, 10:24–35, 2015.

D. Goyal, B. Pabla, and S. Dhami. Condition monitoring parameters for fault diagnosis of fixed axis gearbox: A review. *Archives of Computational Methods in Engineering*, 24(3):543–556, 2017.

R. Gravina, P. Alinia, H. Ghasemzadeh, and G. Fortino. Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges. *Information Fusion*, 35:68–80, 2017.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(Mar):1157–1182, 2003.

I. Guyon and A. Elisseeff. An introduction to feature extraction. In *Feature Extraction*, pages 1–25. Springer, 2006.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422, 2002.

D. L. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1): 6–23, 1997.

M. A. Hall and L. A. Smith. Practical feature subset selection for machine learning. In *Computer science'98 proceedings of the 21st Australasian computer science conference ACSC*, volume 98, pages 181–191, 1998.

E. Hancer, B. Xue, and M. Zhang. Differential evolution for filter feature selection based on information theory and feature ranking. *Knowledge-Based Systems*, 140:103–119, 2018.

K. He, Z. Zhao, M. Jia, and C. Liu. Dynamic Bayesian network-based approach by integrating sensor deployment for machining process monitoring. *IEEE Access*, 2018.

C. Heinz and B. Seeger. Towards kernel density estimation over streaming data. In *COMAD*, pages 80–91, 2006.

C. Heinz and B. Seeger. Cluster kernels: Resource-aware kernel density estimators over streaming data. *IEEE Transactions on Knowledge and Data Engineering*, 20(7):880–893, 2008.

C. Hellier and M. Shakinovsky. *Handbook of nondestructive evaluation*, volume 10. McGraw-Hill Education, New York, 2001.

A. Heng, S. Zhang, A. C. Tan, and J. Mathew. Rotating machinery prognostics: State of the art, challenges and opportunities. *Mechanical Systems and Signal Processing*, 23(3):724–739, 2009.

T. Heyns, S. Godsill, J. P. De Villiers, and P. S. Heyns. Statistical gear health analysis which is robust to fluctuating loads and operating speeds. *Mechanical Systems and Signal Processing*, 27:651–666, 2012.

V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22 (2):85–126, 2004.

T. Höfling and R. Isermann. Fault detection based on adaptive parity equations and single-parameter tracking. *Control Engineering Practice*, 4(10):1361–1369, 1996.

L. Hou and N. W. Bergmann. Novel industrial wireless sensor networks for machine condition monitoring and fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, 61(10):2787–2798, 2012.

Y. Hu, P. Baraldi, F. Di Maio, and E. Zio. A systematic semi-supervised self-adaptable fault diagnostics approach in an evolving environment. *Mechanical Systems and Signal Processing*, 88:413–427, 2017.

S. H. Huang. Supervised feature selection: A tutorial. *Artificial Intelligence Research*, 4(2):22, 2015.

IEC 62682:2014. Management of alarms systems for the process industries. Standard, International Electrotechnical Commission, 2014.

IEEE. Report of large motor reliability survey of industrial and commercial installations, Part I. *IEEE Transactions on Industry Applications*, IA-21(4):853–864, 1985.

IEEE 493-2007. IEEE recommended practice for the design of reliable industrial and commercial power systems. Standard, 2007.

R. Isermann. On fuzzy logic applications for automatic control, supervision, and fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 28(2):221–235, 1998.

R. Isermann. *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Science & Business Media, 2006.

R. Isermann. *Fault-diagnosis applications: model-based condition monitoring: actuators, drives, machinery, plants, sensors, and fault-tolerant systems*. Springer Science & Business Media, 2011.

ISO 10816-1:1995. Mechanical vibration – evaluation of machine vibration by measurements on non-rotating parts – Part 1: General guidelines. Standard, International Organization for Standardization, 1995.

ISO 10816-3:2009. Mechanical vibration – evaluation of machine vibration by measurements on non-rotating parts – Part 3: Industrial machines with nominal power above 15 kw and nominal speeds between 120 r/min and 15 000 r/min when measured in situ. Standard, International Organization for Standardization, 2014.

ISO 17359:2018. Condition monitoring and diagnostics of machines – general guidelines. Standard, 2018.

F. Jansen, O. Shoham, and Y. Taitel. The elimination of severe slugging—experiments and modeling. *International Journal of Multiphase Flow*, 22(6):1055–1072, 1996.

O. Janssens, R. Schulz, V. Slavkovikj, K. Stockman, M. Loccufier, R. Van de Walle, and S. Van Hoecke. Thermal image based fault diagnosis for rotating machinery. *Infrared Physics & Technology*, 73: 78–87, 2015.

V. H. Jaramillo, J. R. Ottewill, and M. Orkisz. Condition monitoring and its link to energy savings. `http://www.energy-smartops.eu/single-post/2014/03/11/Condition-Monitoring-and-its-Link-to-Energy-Savings`, 2014. Accessed: 2019-01-10.

V. H. Jaramillo, J. R. Ottewill, R. Dudek, D. Lepiarczyk, and P. Pawlik. Condition monitoring of distributed systems using two-stage Bayesian inference data fusion. *Mechanical Systems and Signal Processing*, 87:91–110, 2017.

A. K. Jardine and A. H. Tsang. *Maintenance, replacement, and reliability: theory and applications*. CRC press, 2005.

A. K. Jardine, D. Lin, and D. Banjevic. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20(7):1483–1510, 2006.

F. V. Jensen et al. *An introduction to Bayesian networks*, volume 210. UCL press London, 1996.

J. Jiao, J. Yue, and D. Pei. Multi-sensor information fusion method for vibration fault diagnosis of rolling bearing. In *IOP Conference Series: Materials Science and Engineering*, volume 241, page 012034. IOP Publishing, 2017.

S. Joe Qin. Statistical process monitoring: basics and beyond. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 17(8-9):480–502, 2003.

I. Jolliffe. *Principal Component Analysis*, pages 1094–1096. Springer, 2011.

D. S. Jung, H. E. Kim, S. H. Kim, and E. S. Kang. Development of accelerated life test method of hydraulic pump. In *Key Engineering Materials*, volume 326, pages 1861–1864. Trans Tech Publ, 2006.

A. Kachko, L. V. van der Ham, L. F. Geers, A. Huizinga, A. Rieder, M. R. Abu-Zahra, T. J. Vlugt, and E. L. Goetheer. Real-time process monitoring of CO_2 capture by aqueous AMP-PZ using chemometrics: pilot plant demonstration. *Industrial & Engineering Chemistry Research*, 54(21):5769–5776, 2015.

S. A. Kalogirou. Artificial intelligence for the modeling and control of combustion processes: a review. *Progress in Energy and Combustion Science*, 29(6):515–566, 2003.

M. S. Kan, A. C. Tan, and J. Mathew. A review on prognostic techniques for non-stationary and non-linear rotating systems. *Mechanical Systems and Signal Processing*, 62:1–20, 2015.

E. Karakose, M. T. Gencoglu, M. Karakose, I. Aydin, and E. Akin. A new experimental approach using image processing-based tracking for an efficient fault diagnosis in pantograph–catenary systems. *IEEE Transactions on Industrial Informatics*, 13(2):635–643, 2017.

S. Karmakar, S. Chattopadhyay, M. Mitra, and S. Sengupta. *Induction motor fault diagnosis*. Springer, 2016.

K. Kawaguchi, M. Takekawa, T. Ohtani, H. Wada, et al. Optimal sensor placement for multi-phase flow rate estimation using pressure and temperature measurements. In *SPE Digital Energy Conference*. Society of Petroleum Engineers, 2013.

B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1):28–44, 2013.

M. Khazaee, H. Ahmadi, M. Omid, A. Moosavian, and M. Khazaee. Classifier fusion of vibration and acoustic signals for fault diagnosis and classification of planetary gears based on Dempster–Shafer evidence theory. *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, 228(1):21–32, 2014.

I. Khelf, L. Laouar, H. Bendjama, and A. M. Bouchelaghem. Combining RBF-PCA-ReliefF filter for a better diagnosis performance in rotating machines. In *Condition Monitoring of Machinery in Non-Stationary Operations*, pages 285–292. Springer, 2012.

A. M. Kibriya, E. Frank, B. Pfahringer, and G. Holmes. Multinomial naive Bayes for text categorization revisited. In *Australasian Joint Conference on Artificial Intelligence*, pages 488–499. Springer, 2004.

K. Kira and L. A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *AAAI*, volume 2, pages 129–134, 1992a.

K. Kira and L. A. Rendell. A practical approach to feature selection. In *Machine Learning Proceedings 1992*, pages 249–256. Elsevier, 1992b.

A. N. Kolmogorov. *Foundations of the theory of probability*. Chelsea Publishing Company, New York, 1956.

I. Kononenko. Estimating attributes: analysis and extensions of RELIEF. In *European Conference on Machine Learning (ECML 1994)*, pages 171–182. Springer, 1994.

I. Kononenko, E. Šimec, and M. Robnik-Šikonja. Overcoming the myopia of inductive learning algorithms with RELIEFF. *Applied Intelligence*, 7(1):39–55, 1997.

J. V. Kresta, J. F. Macgregor, and T. E. Marlin. Multivariate statistical monitoring of process operating performance. *The Canadian Journal of Chemical Engineering*, 69(1):35–47, 1991.

J. Kullaa. Detection, identification, and quantification of sensor fault in a sensor network. *Mechanical Systems and Signal Processing*, 40(1):208–221, 2013.

H. Kumar, T. Ranjit Kumar, M. Amarnath, and V. Sugumaran. Fault diagnosis of bearings through vibration signal using Bayes classifiers. *International Journal of Computer Aided Engineering and Technology*, 6(1):14–28, 2014a.

M. G. Kumar, K. Hemanth, N. Gangadhar, H. Kumar, and P. Krishna. Fault diagnosis of welded joints through vibration signals using Naïve Bayes algorithm. *Procedia Materials Science*, 5:1922–1928, 2014b.

J. Kurien and P. P. Nayak. Back to the future for consistency-based trajectory tracking. In *AAAI/IAAI*, pages 370–377, 2000.

S. Lai and T. Chen. A method for pattern mining in multiple alarm flood sequences. *Chemical Engineering Research and Design*, 117:831–839, 2017.

S. Lai, F. Yang, and T. Chen. Online pattern matching and prediction of incoming alarm floods. *Journal of Process Control*, 56:69–78, 2017.

M. Lampis and J. Andrews. Bayesian belief networks for system fault diagnostics. *Quality and Reliability Engineering International*, 25(4):409–426, 2009.

P. S. Laplace. Théorie analytique des probabilités. *Paris: Courcier Imprimeur*, 1812.

C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. de Schaetzen, R. Duque, H. Bersini, and A. Nowe. A survey on filter techniques for feature selection in gene expression microarray analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 9 (4):1106–1119, 2012.

A. J. Lee and U. M. Diwekar. Optimal sensor placement in integrated gasification combined cycle power systems. *Applied Energy*, 99:255–264, 2012.

J.-M. Lee, C. Yoo, S. W. Choi, P. A. Vanrolleghem, and I.-B. Lee. Nonlinear process monitoring using kernel principal component analysis. *Chemical Engineering Science*, 59(1):223–234, 2004a.

J.-M. Lee, C. Yoo, and I.-B. Lee. Statistical process monitoring with independent component analysis. *Journal of Process Control*, 14(5):467–485, 2004b.

J.-M. Lee, S. J. Qin, and I.-B. Lee. Fault detection of non-linear processes using kernel independent component analysis. *The Canadian Journal of Chemical Engineering*, 85(4):526–536, 2007.

J. Lever, M. Krzywinski, and N. Altman. Points of significance: model selection and overfitting. *Nature Methods*, 13:703—704, 2016.

D. D. Lewis. Naive (Bayes) at forty: The independence assumption in information retrieval. In *European conference on machine learning*, pages 4–15. Springer, 1998.

B. Li, M.-Y. Chow, Y. Tipsuwan, and J. C. Hung. Neural-network-based motor rolling bearing fault diagnosis. *IEEE Transactions on Industrial Electronics*, 47(5):1060–1069, 2000.

C. J. Li and H. Lee. Gear fatigue crack prognosis using embedded model, gear dynamic model and fracture mechanics. *Mechanical Systems and Signal Processing*, 19(4):836–846, 2005.

J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):94, 2018.

W. Li and C. K. Mechefske. Detection of induction motor faults: a comparison of stator current, vibration and acoustic methods. *Journal of Vibration and Control*, 12(2):165–188, 2006.

X. Li. A brief review: acoustic emission method for tool wear monitoring during turning. *International Journal of Machine Tools and Manufacture*, 42(2):157–165, 2002.

Y. Li and P. Nilkitsaranont. Gas turbine performance prognostic for condition-based maintenance. *Applied Energy*, 86(10):2152–2161, 2009.

X. Liang, M. J. Zuo, and Z. Feng. Dynamic modeling of gearbox faults: A review. *Mechanical Systems and Signal Processing*, 98:852–876, 2018.

H. Liao, W. Zhao, and H. Guo. Predicting remaining useful life of an individual unit using proportional hazards model and logistic regression model. In *Reliability and Maintainability Symposium, 2006. RAMS'06. Annual*, pages 127–132. IEEE, 2006.

M. Liggins II, D. L. Hall, and J. Llinas. *Handbook of multisensor data fusion: theory and practice*. CRC press, 2017.

R. J. Little and D. B. Rubin. *Statistical analysis with missing data*, volume 333. John Wiley & Sons, 2014.

B. Liu, C. Wan, and L. Wang. An efficient semi-unsupervised gene selection method via spectral biclustering. *IEEE Transactions on Nanobioscience*, 5(2):110–114, 2006.

J. Liu, Q. Li, W. Chen, Y. Yan, and X. Wang. A fast fault diagnosis method of the pemfc system based on extreme learning machine and Dempster-Shafer evidence theory. *IEEE Transactions on Transportation Electrification*, 2018a.

K. Liu, N. Z. Gebraeel, and J. Shi. A data-level fusion model for developing composite health indices for degradation modeling and prognostic analysis. *IEEE Transactions on Automation Science and Engineering*, 10(3):652–664, 2013.

R. Liu, B. Yang, E. Zio, and X. Chen. Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mechanical Systems and Signal Processing*, 108:33–47, 2018b.

W. Liu, B. Tang, J. Han, X. Lu, N. Hu, and Z. He. The structure healthy condition monitoring and fault diagnosis methods in wind turbines: A review. *Renewable and Sustainable Energy Reviews*, 44: 466–472, 2015.

T. Loutas, D. Roulias, E. Pauly, and V. Kostopoulos. The combined use of vibration, acoustic emission and oil debris on-line monitoring towards a more effective condition monitoring of rotating machinery. *Mechanical Systems and Signal Processing*, 25(4):1339–1352, 2011.

T. H. Loutas, D. Roulias, and G. Georgoulas. Remaining useful life estimation in rolling bearings utilizing data-driven probabilistic e-support vectors regression. *IEEE Transactions on Reliability*, 62(4): 821–832, 2013.

F. Lu, C. Jiang, J. Huang, Y. Wang, and C. You. A novel data hierarchical fusion method for gas turbine engine performance fault diagnosis. *Energies*, 9(10):828, 2016.

Z.-J. Lu, Q. Xiang, and L. Xu. An application case study on multi-sensor data fusion system for intelligent process monitoring. *Procedia CIRP*, 17:721–725, 2014.

M. Lucke, M. Chioua, C. Grimholt, M. Hollender, and N. F. Thornhill. Online alarm flood classification using alarm coactivations. *IFAC-PapersOnLine*, 51(18):345–350, 2018.

R. C. Luo and M. G. Kay. Multisensor integration and fusion in intelligent systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5):901–931, 1989.

L. C. Lynnworth. *Ultrasonic measurements for process control: theory, techniques, applications*. Academic press, 2013.

J. Ma and J. Jiang. Applications of fault detection and diagnosis methods in nuclear power plants: A review. *Progress in nuclear energy*, 53(3):255–266, 2011.

C. Madhusudana, S. Budati, N. Gangadhar, H. Kumar, and S. Narendranath. Fault diagnosis studies of face milling cutter using machine learning approach. *Journal of Low Frequency Noise, Vibration and Active Control*, 35(2):128–138, 2016.

R. Mahler. Random sets: Unification and computation for information fusion-a retrospective assessment. In *Proceedings of the Seventh International Conference on Information Fusion*, volume 1, pages 1–20. I, 2004.

S. Malik and V. Verma. Comparative analysis of DCT, Haar and Daubechies wavelet for image compression. *International Journal of Applied Engineering Research*, 7(11):2012, 2012.

F. P. G. Márquez, A. M. Tobias, J. M. P. Pérez, and M. Papaelias. Condition monitoring of wind turbines: Techniques and methods. *Renewable Energy*, 46:169–178, 2012.

I. Martin-Diaz, D. Morinigo-Sotelo, O. Duque-Perez, and R. J. Romero-Troncoso. An experimental comparative evaluation of machine learning techniques for motor fault diagnosis under various operating conditions. *IEEE Transactions on Industry Applications*, 54(3):2215–2224, 2018.

A. Mauricio, J. Qi, W. Smith, R. Randall, and K. Gryllias. Vibration based condition monitoring of planetary gearboxes operating under speed varying operating conditions based on cyclo-non-stationary analysis. In *International Conference on Rotor Dynamics*, pages 265–279. Springer, 2018.

P. McFadden. A revised model for the extraction of periodic waveforms by time domain averaging. *Mechanical Systems and Signal Processing*, 1(1):83–95, 1987.

M. R. Mehrjou, N. Mariun, M. H. Marhaban, and N. Misron. Rotor fault condition monitoring techniques for squirrel-cage induction machine—a review. *Mechanical Systems and Signal Processing*, 25(8): 2827–2848, 2011.

M. Mejía-Lavalle, E. Sucar, and G. Arroyo. Feature selection with a perceptron neural net. In *Proceedings of the international workshop on feature selection for data mining*, pages 131–135, 2006.

J. Miao and L. Niu. A survey on feature selection. *Procedia Computer Science*, 91:919–926, 2016.

P. Miller, R. E. Swanson, and C. E. Heckler. Contribution plots: a missing link in multivariate quality control. *Applied Mathematics and Computer Science*, 8(4):775–792, 1998.

T. M. Mitchell. *Machine learning*. McGraw-Hill Science/Engineering/Math, 1997.

A. M. Molinaro, R. Simon, and R. M. Pfeiffer. Prediction error estimation: a comparison of resampling methods. *Bioinformatics*, 21(15):3301–3307, 2005.

A. Mujahid and F. L. Dickert. Monitoring automotive oil degradation: analytical tools and onboard sensing technologies. *Analytical and Bioanalytical Chemistry*, 404(4):1197–1209, 2012.

E. F. Nakamura, A. A. Loureiro, and A. C. Frery. Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Computing Surveys (CSUR)*, 39(3):9, 2007.

S. Nandi, H. A. Toliyat, and X. Li. Condition monitoring and fault diagnosis of electrical motors – a review. *IEEE Transactions on Energy Conversion*, 20(4):719–729, 2005.

S. Natarajan and R. Srinivasan. Multi-model based process condition monitoring of offshore oil and gas production process. *Chemical Engineering Research and Design*, 88(5-6):572–591, 2010.

J. W. Newman, T. I. John, and T. Gluyas. Condition monitoring system for locomotives, June 21 1988. US Patent 4,752,899.

A. Y. Ng. *On feature selection: learning with exponentially many irrevelant features as training examples*. PhD thesis, Massachusetts Institute of Technology, 1998.

A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems*, pages 841–848, 2002.

F. Nie, S. Xiang, Y. Jia, C. Zhang, and S. Yan. Trace ratio criterion for feature selection. In *AAAI*, volume 2, pages 671–676, 2008.

X. Ning and F. Tsung. A density-based statistical process control scheme for high-dimensional and mixed-type observations. *IIE transactions*, 44(4):301–311, 2012.

G. Niu and H. Li. IETM centered intelligent maintenance system integrating fuzzy semantic inference and data fusion. *Microelectronics Reliability*, 75:197–204, 2017.

P.-E. P. Odiowei and Y. Cao. Nonlinear dynamic process monitoring using canonical variate analysis and kernel density estimations. *IEEE Transactions on Industrial Informatics*, 6(1):36–45, 2010.

M. Onel, C. A. Kieslich, Y. A. Guzman, C. A. Floudas, and E. N. Pistikopoulos. Big data approach to batch process monitoring: Simultaneous fault detection and diagnosis using nonlinear support vector machine-based feature selection. *Computers & Chemical Engineering*, 115:46–63, 2018.

C. H. Oppenheimer and K. A. Loparo. Physically based diagnosis and prognosis of cracked rotor shafts. In *Component and Systems Diagnostics, Prognostics, and Health Management II*, volume 4733, pages 122–133. International Society for Optics and Photonics, 2002.

OREDA. *Offshore Reliability Data Handbook (OREDA)*. Høvik: Det Norske Veritas, 2002.

M. Orkisz. Practical aspects of machine diagnostics accuracy. In *Diagnostics for Electrical Machines, Power Electronics and Drives (SDEMPED), 2017 IEEE 11th International Symposium on*, pages 260–266. IEEE, 2017.

M. Orkisz, M. Wnek, P. Joerg, K. Ruetten, and E. Jellum. Selected data analysis techniques for equipment monitoring using drive's control signals. In *2009 IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives*, pages 1–5. IEEE, 2009.

F. Pacheco, M. Cerrada, R. V. Sánchez, D. Cabrera, C. Li, and J. V. de Oliveira. Clustering algorithm using rough set theory for unsupervised feature selection. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pages 3493–3499. IEEE, 2016.

J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier, 2014.

S. Pedersen, P. Durdevic, and Z. Yang. Review of slug detection, modeling and control techniques for offshore oil & gas production processes. *IFAC-PapersOnLine*, 48(6):89–96, 2015.

H. Peng and Y. Fan. Direct sparsity optimization based feature selection for multi-class classification. In *IJCAI*, pages 1918–1924, 2016.

H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

Y. Peng, M. Dong, and M. J. Zuo. Current status of machine prognostics in condition-based maintenance: a review. *The International Journal of Advanced Manufacturing Technology*, 50(1-4):297–313, 2010.

Z. Peng and F. Chu. Application of the wavelet transform in machine condition monitoring and fault diagnostics: a review with bibliography. *Mechanical Systems and Signal Processing*, 18(2):199–221, 2004.

C. Pohl and J. L. Van Genderen. Review article multisensor image fusion in remote sensing: concepts, methods and applications. *International Journal of Remote Sensing*, 19(5):823–854, 1998.

R. Polikar. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3): 21–45, 2006.

PRONTO Consortium. PRONTO grant agreement - 675215. *European Comission, Research Executive Agency*, 2015. H2020-MSCA-ITN-2015.

W. F. Punch, E. D. Goodman, M. Pei, L. Chia-Shun, P. D. Hovland, and R. J. Enbody. Further research on feature selection and classification using genetic algorithms. In *ICGA*, pages 557–564, 1993.

A. F. R. Rahman, H. Alam, and M. C. Fairhurst. Multiple classifier combination for character recognition: Revisiting the majority voting system and its variations. In *International Workshop on Document Analysis Systems*, pages 167–178. Springer, 2002.

R. B. Randall. *Vibration-based condition monitoring: industrial, aerospace and automotive applications*. John Wiley & Sons, 2011.

A. Rastegari and M. Bengtsson. Implementation of condition based maintenance in manufacturing industry-a pilot case study. In *Prognostics and Health Management (PHM), 2014 IEEE Conference on*, pages 1–8. IEEE, 2014.

T. W. Rauber, F. de Assis Boldt, and F. M. Varejão. Heterogeneous feature models and feature selection applied to bearing fault diagnosis. *IEEE Transactions on Industrial Electronics*, 62(1):637–646, 2015.

J. D. Rennie, L. Shih, J. Teevan, and D. R. Karger. Tackling the poor assumptions of naive Bayes text classifiers. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 616–623, 2003.

N. Ricker. Optimal steady-state operation of the Tennessee Eastman challenge process. *Computers & Chemical Engineering*, 19(9):949–959, 1995.

I. Rish. An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

M. Robnik-Šikonja and I. Kononenko. An adaptation of Relief for attribute estimation in regression. In *Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97)*, volume 5, pages 296–304, 1997.

M. Robnik-Šikonja and I. Kononenko. Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning*, 53(1-2):23–69, 2003.

C. Ruiz-Cárcel, Y. Cao, D. Mba, L. Lao, and R. T. Samuel. Statistical process monitoring of a multiphase flow facility. *Control Engineering Practice*, 42:74–88, 2015.

C. Ruiz-Cárcel, V. H. Jaramillo, D. Mba, J. R. Ottewill, and Y. Cao. Combination of process and vibration data for improved condition monitoring of industrial systems working under variable operating conditions. *Mechanical Systems and Signal Processing*, 66:699–714, 2016.

M. Safizadeh and S. Latifi. Using multi-sensor data fusion for vibration fault diagnosis of rolling element bearings by accelerometer and load cell. *Information Fusion*, 18:1–8, 2014.

T. K. Saha. Review of modern diagnostic techniques for assessing insulation condition in aged transformers. *IEEE Transactions on Dielectrics and Electrical Insulation*, 10(5):903–917, 2003.

B. Santoso. The identification of gas-liquid co-current two phase flow pattern in a horizontal pipe using the power spectral density and the artificial neural network (ANN). *Modern Applied Science*, 6(9): 56–67, 2012.

J. J. Saucedo-Dorantes, M. Delgado-Prieto, R. A. Osornio-Rios, and R. J. Romero-Troncoso. Multifault diagnosis method applied to an electric machine based on high-dimensional feature reduction. *IEEE Transactions on Industry Applications*, pages 3086–3097, 2017.

A. Schön. Evaluation of the life-cycle costs of plants incorporating condition monitoring systems. Master thesis, 2017.

R. R. Schoen, B. K. Lin, T. G. Habetler, J. H. Schlag, and S. Farag. An unsupervised, on-line system for induction motor fault detection using stator current monitoring. *IEEE Transactions on Industry Applications*, 31(6):1280–1286, 1995.

S. M. Schultheis, C. A. Lickteig, and R. Parchewsky. Reciprocating compressor condition monitoring. In *Proceedings of the 36th Turbomachinery Symposium*. Texas A&M University. Turbomachinery Laboratories, 2007.

P. Seleghim Jr and F. E. Milioli. Improving the determination of bubble size histograms by wavelet de-noising techniques. *Powder Technology*, 115(2):114–123, 2001.

J. S. L. Senanayaka and K. G. Robbersmyr. A robust method for detection and classification of permanent magnet synchronous motor faults: Deep autoencoders and data fusion approach. In *Journal of Physics: Conference Series*, volume 1037, page 032029. IOP Publishing, 2018.

G. Shafer. *A mathematical theory of evidence*, volume 42. Princeton University Press, 1976.

Z. Shang, R. Yang, X. Cao, and Y. Yang. An investigation of two-phase flow instability using wavelet signal extraction technique. *Nuclear Engineering and Design*, 232(2):157–163, 2004.

C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

R. K. Sharma, V. Sugumaran, H. Kumar, and M. Amarnath. A comparative study of naïve Bayes classifier and Bayes net classifier for fault diagnosis of roller bearing using sound signal. *International Journal of Decision Support Systems*, 1(1):115–129, 2015.

R. Sheikhpour, M. A. Sarram, S. Gharaghani, and M. A. Z. Chahooki. A survey on semi-supervised feature selection methods. *Pattern Recognition*, 64:141–158, 2017.

X.-S. Si, W. Wang, C.-H. Hu, and D.-H. Zhou. Remaining useful life estimation–a review on the statistical data driven approaches. *European Journal of Operational Research*, 213(1):1–14, 2011.

A. Siddique, G. Yadava, and B. Singh. A review of stator fault monitoring techniques of induction motors. *IEEE Transactions on Energy Conversion*, 20(1):106–114, 2005.

J. Sikorska, M. Hodkiewicz, and L. Ma. Prognostic modelling options for remaining useful life estimation by industry. *Mechanical Systems and Signal Processing*, 25(5):1803–1836, 2011.

B. W. Silverman. *Density estimation for statistics and data analysis*. Chapman and Hall, 1986.

A. Slimani, P. Ribot, E. Chanthery, and N. Rachedi. Fusion of model-based and data-based fault diagnosis approaches. *IFAC-PapersOnLine*, 51(24):1205–1211, 2018.

D. Sodkomkham, D. Ciliberti, M. A. Wilson, K.-i. Fukui, K. Moriyama, M. Numao, and F. Kloosterman. Kernel density compression for real-time Bayesian encoding/decoding of unsorted hippocampal spikes. *Knowledge-Based Systems*, 94:1–12, 2016.

C. Song, K. Liu, and X. Zhang. Integration of data-level fusion model and kernel methods for degradation modeling and prognostic analysis. *IEEE Transactions on Reliability*, 67(2):640–650, 2018.

A. Srinivasan, D. Dornfeld, and R. Bhinge. Integrated vibration and acoustic data fusion for chatter and tool condition classification in milling. In *Flexible Automation (ISFA), International Symposium on*, pages 263–266. IEEE, 2016.

W. J. Staszewski, K. Worden, and G. R. Tomlinson. Time–frequency analysis in gearbox fault detection using the Wigner–Ville distribution and pattern recognition. *Mechanical Systems and Signal Processing*, 11(5):673–692, 1997.

A. Stief and R. Tan. Multiphase flow facility case study with heterogeneous data, 2018. URL `10.5281/zenodo.1341583`.

A. Stief, J. R. Ottewill, M. Orkisz, and J. Baranowski. Two stage data fusion of acoustic, electric and vibration signals for diagnosing faults in induction motors. *Elektronika ir Elektrotechnika*, 23(6):19–24, 2017.

A. Stief, J. R. Ottewill, and J. Baranowski. ReliefF-based feature ranking and feature selection for monitoring induction motors. In *2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR)*, pages 171–176. IEEE, 2018a.

A. Stief, J. R. Ottewill, R. Tan, and Y. Cao. Process and alarm data integration under a two-stage Bayesian framework for fault diagnostics. *IFAC-PapersOnLine*, 51(24):1220–1226, 2018b.

A. Stief, R. Tan, Y. Cao, and J. R. Ottewill. Analytics of heterogeneous process data: Multiphase flow facility case study. *IFAC-PapersOnLine*, 51(18):363–368, 2018c.

A. Stief, J. R. Ottewill, and J. Baranowski. Fault diagnosis using interpolated kernel density estimate. In *58th IEEE Conference on Decision and Control (CDC)*. IEEE, 2019a. submitted.

A. Stief, J. R. Ottewill, and J. Baranowski. Investigation of the diagnostic properties of sensors and features in a multiphase flow facility case study. In *12th IFAC Symposium on Dynamics and Control of Process Systems*, 2019b. doi: 10.5281/zenodo.3235443.

A. Stief, J. R. Ottewill, J. Baranowski, and M. Orkisz. A PCA - two stage Bayesian sensor fusion approach for diagnosing electrical and mechanical faults in induction motors. *IEEE Transactions on Industrial Electronics*, 2019c. doi: 10.1109/TIE.2019.2891453.

A. Stief, R. Tan, Y. Cao, J. R. Ottewill, N. F. Thornhill, and J. Baranowski. A heterogeneous benchmark dataset for data analytics: Multiphase flow facility case study. *Journal of Process Control*, 79:41–55, 2019d. doi: 10.1016/j.jprocont.2019.04.009.

Y. Sun. Iterative RELIEF for feature weighting: algorithms, theories, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6), 2007.

Z. Sun and H. Zhang. Neural networks approach for prediction of gas–liquid two-phase flow pattern based on frequency domain analysis of vortex flowmeter signals. *Measurement Science and Technology*, 19(1):015401, 2007.

P. Sundin, N. Montgomery, and A. K. Jardine. Pulp mill on-site implementation of cbm decision support software. In *Proceedings of International Conference of Maintenance Societies*, 2007.

T. Sutharssan, S. Stoyanov, C. Bailey, and C. Yin. Prognostic and health management for engineering systems: a review of the data-driven approach and algorithms. *The Journal of Engineering*, 1(1), 2015.

A. Swetapadma and A. Yadav. Protection of parallel transmission lines including inter-circuit faults using naive Bayes classifier. *Alexandria Engineering Journal*, 55(2):1411–1419, 2016.

R. Tan and A. Stief. Multiphase flow facility case study technical report, 2018. URL `10.5281/zenodo.1341583`.

N. Tandon and A. Choudhury. A review of vibration and acoustic measurement methods for the detection of defects in rolling element bearings. *Tribology International*, 32(8):469–480, 1999.

P. Tchakoua, R. Wamkeue, T. A. Tameghe, and G. Ekemb. A review of concepts and methods for wind turbines condition monitoring. In *2013 World Congress on Computer and Information Technology (WCCIT),*, pages 1–9. IEEE, 2013.

R. Teti, K. Jemielniak, G. O'Donnell, and D. Dornfeld. Advanced monitoring of machining operations. *CIRP Annals-Manufacturing Technology*, 59(2):717–739, 2010.

W. T. Thomson. On-line mcsa to diagnose shorted turns in low voltage stator windings of 3-phase induction motors prior to failure. In *IEMDC 2001. IEEE International Electric Machines and Drives Conference (Cat. No. 01EX485)*, pages 891–898. IEEE, 2001.

N. Thornhill, M. Oettinger, and P. Fedenczuk. Refinery-wide control loop performance assessment. *Journal of Process Control*, 9(2):109–124, 1999.

N. F. Thornhill. Finding the source of nonlinearity in a process with plant-wide oscillation. *IEEE Transactions on Control Systems Technology*, 13(3):434–443, 2005.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

K. Tidriri, N. Chatti, S. Verron, and T. Tiplica. Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges. *Annual Reviews in Control*, 42:63–81, 2016.

K. Torkkola. Feature extraction by non-parametric mutual information maximization. *Journal of Machine Learning Research*, 3(Mar):1415–1438, 2003.

L. N. Trefethen. *Approximation theory and approximation practice*, volume 128. Siam, 2013.

L. N. Trefethen and J. Weideman. Two results on polynomial interpolation in equally spaced points. *Journal of Approximation Theory*, 65(3):247–260, 1991.

K. L. Tsui, N. Chen, Q. Zhou, Y. Hai, and W. Wang. Prognostics and health management: A review on data driven approaches. *Mathematical Problems in Engineering*, 2015, 2015.

G. Tuerhong and S. B. Kim. Gower distance-based multivariate control charts for a mixture of continuous and categorical variables. *Expert Systems with Applications*, 41(4):1701–1707, 2014.

M. Tutkan, M. C. Ganiz, and S. Akyokuş. Helmholtz principle based supervised and unsupervised feature selection methods for text mining. *Information Processing & Management*, 52(5):885–910, 2016.

C. J. Van Rijsbergen. *The geometry of information retrieval*. Cambridge University Press, 2004.

K. Verbert, R. Babuška, and B. De Schutter. Combining knowledge and historical data for system-level fault diagnosis of hvac systems. *Engineering Applications of Artificial Intelligence*, 59:260–273, 2017.

K. Vernekar, H. Kumar, and K. Gangadharan. Engine gearbox fault diagnosis using empirical mode decomposition method and naïve Bayes algorithm. *Sādhanā*, 42(7):1143–1153, 2017.

S. Verron, T. Tiplica, and A. Kobi. Fault diagnosis with Bayesian networks: Application to the tennessee eastman process. In *2006 IEEE International Conference on Industrial Technology*, pages 98–103. IEEE, 2006.

L. Wang, Y. Wang, and Q. Chang. Feature selection methods for big data bioinformatics: A survey from the search perspective. *Methods*, 111:21–31, 2016.

X. Wang, G. Foliente, Z. Su, and L. Ye. Multilevel decision fusion in a distributed active sensor network for structural damage detection. *Structural Health Monitoring*, 5(1):45–58, 2006.

K. Weinreb, M. Sułowicz, and J. Petryna. Faults detection in cage induction motor with parallel branches. *Czasopismo Techniczne*, 2016(Elektrotechnika Zeszyt 2-E 2016):53–64, 2016.

P. Welch. The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15 (2):70–73, 1967.

D. S. Weld and J. De Kleer. *Readings in qualitative reasoning about physical systems*. Morgan Kaufmann, 2013.

A. Widodo and B.-S. Yang. Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, 21(6):2560–2574, 2007.

B. M. Wise and N. B. Gallagher. The process chemometrics approach to process monitoring and fault detection. *Journal of Process Control*, 6(6):329–348, 1996.

T. Xie, S. Ghiaasiaan, and S. Karrila. Artificial neural network approach for flow regime classification in gas–liquid–fiber flows based on frequency domain analysis of pressure signals. *Chemical Engineering Science*, 59(11):2241–2251, 2004.

B. G. Xu. Intelligent fault inference for rotating flexible rotors using bayesian belief network. *Expert Systems with Applications*, 39(1):816–822, 2012.

M. Xu, H. Ishibuchi, X. Gu, and S. Wang. Dm-KDE: dynamical kernel density estimation by sequences of KDE estimators with fixed number of components over data streams. *Frontiers of Computer Science*, 8(4):563–580, 2014.

S. Xu, B. Lu, M. Baldea, T. F. Edgar, W. Wojsznis, T. Blevins, and M. Nixon. Data cleaning in the process industries. *Reviews in Chemical Engineering*, 31(5):453–490, 2015.

S. Xu, X. Jiang, J. Huang, S. Yang, and X. Wang. Bayesian wavelet PCA methodology for turbomachinery damage diagnosis under uncertainty. *Mechanical systems and signal processing*, 80:1–18, 2016.

H. Yan, K. Liu, X. Zhang, and J. Shi. Multiple sensor data fusion for degradation modeling and prognostics under multiple operational conditions. *IEEE Transactions on Reliability*, 65(3):1416–1426, 2016.

J. Yan, M. Koc, and J. Lee. A prognostic algorithm for machine performance assessment and its application. *Production Planning & Control*, 15(8):796–801, 2004.

K. Yan, N. Wang, Q. Zhai, Y. Zhu, J. Zhang, and Q. Niu. Theoretical and experimental investigation on the thermal characteristics of double-row tapered roller bearings of high speed locomotive. *International Journal of Heat and Mass Transfer*, 84:1119–1130, 2015.

B.-S. Yang and K. J. Kim. Application of Dempster–Shafer theory in fault diagnosis of induction motors using vibration and current signals. *Mechanical Systems and Signal Processing*, 20(2):403–420, 2006.

B.-S. Yang, T. Han, and Z.-J. Yin. Fault diagnosis system of induction motors using feature extraction, feature selection and classification algorithm. *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing*, 49(3):734–741, 2006.

Z. Yang, J. Wang, and T. Chen. Detection of correlated alarms based on similarity coefficients of binary data. *IEEE Transactions on Automation Science and Engineering*, 10(4):1014–1025, 2013.

C. Yao, Y.-F. Liu, B. Jiang, J. Han, and J. Han. LLE score: a new filter-based unsupervised feature selection method based on nonlinear manifold embedding and its application to image recognition. *IEEE Transactions on Image Processing*, 26(11):5257–5269, 2017.

X. Yao, S. Li, and A. Zhang. Equipment condition monitoring and diagnosis system based on evidence weight. *International Journal of Online Engineering (iJOE)*, 14(02):143–154, 2018.

T.-H. Yi, H.-B. Huang, and H.-N. Li. Development of sensor validation methodologies for structural health monitoring: A comprehensive review. *Measurement*, 109:200–214, 2017.

S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang. A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process. *Journal of Process Control*, 22(9):1567–1581, 2012.

L. Yong, X. Congfu, and P. Yunhe. A new approach for data fusion: implement rough set theory in dynamic objects distinguishing and tracing. In *2004 IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 3318–3322. IEEE, 2004.

J. Yu, M. Bai, G. Wang, and X. Shi. Fault diagnosis of planetary gearbox with incomplete information using assignment reduction and flexible naive Bayesian classifier. *Journal of Mechanical Science and Technology*, 32(1):37–47, 2018.

Z. Yuan, L. Zhang, and L. Duan. A novel fusion diagnosis method for rotor system fault based on deep learning and multi-sourced heterogeneous monitoring data. *Measurement Science and Technology*, 29 (11):115005, 2018.

D. Zappalá, N. Sarma, S. Djurović, C. Crabtree, A. Mohammad, and P. Tavner. Electrical & mechanical diagnostic indicators of wind turbine induction generator rotor faults. *Renewable energy*, 131:14–24, 2019.

L. Zarri, M. Mengoni, Y. Gritli, A. Tani, F. Filippetti, G. Serra, and D. Casadei. Detection and localization of stator resistance dissymmetry based on multiple reference frame controllers in multiphase induction motor drives. *IEEE Transactions on Industrial Electronics*, 60(8):3506–3518, 2013.

X. Zeng, Q. Wang, C. Zhang, and H. Cai. Feature selection based on ReliefF and PCA for underwater sound classification. In *Computer Science and Network Technology (ICCSNT), 2013 3rd International Conference on*, pages 442–445. IEEE, 2013.

C. Zhang, H. Yan, S. Lee, and J. Shi. Multiple profiles sensor-based monitoring and anomaly detection. *Journal of Quality Technology*, 50(4):344–362, 2018a.

H. Zhang. The optimality of naive Bayes. *AAAI*, 1(2):3, 2004.

N. Zhang, L. Wu, J. Yang, and Y. Guan. Naive Bayes bearing fault diagnosis based on enhanced independence of data. *Sensors*, 18(2):463, 2018b.

S. Zhang, L. Ma, Y. Sun, and J. Mathew. Asset health reliability estimation based on condition data. pages 2195–2204, 2007.

X. Zhang, R. Xu, C. Kwan, S. Y. Liang, Q. Xie, and L. Haynes. An integrated approach to bearing fault diagnostics and prognostics. In *American Control Conference, 2005. Proceedings of the 2005*, pages 2750–2755. IEEE, 2005.

X. Zhang, Q. Zhang, M. Chen, Y. Sun, X. Qin, and H. Li. A two-stage feature selection and intelligent fault diagnosis method for rotating machinery using hybrid filter and wrapper method. *Neurocomputing*, 275:2426–2439, 2018c.

Y. Zhang, C. Ding, and T. Li. Gene selection algorithm by combining reliefF and mRMR. *BMC Genomics*, 9(2):S27, 2008.

Z. Zhang, X. Si, C. Hu, and X. Kong. Degradation modeling–based remaining useful life estimation: A review on approaches for systems with heterogeneity. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 229(4):343–355, 2015.

C. Zhao and B. Huang. A full-condition monitoring method for nonstationary dynamic chemical processes with cointegration and slow feature analysis. *AIChE Journal*, 64(5):1662–1681, 2018.

A. Zhou, Z. Cai, L. Wei, and W. Qian. M-kernel merging: Towards density estimation over data streams. In *Eighth International Conference on Database Systems for Advanced Applications, 2003.(DASFAA 2003). Proceedings.*, pages 285–292. IEEE, 2003.

Y. Zhou and W. Xue. A multisensor fusion method for tool condition monitoring in milling. *Sensors*, 18 (11):3866, 2018.

H. Zhu and O. Basir. A novel fuzzy evidential reasoning paradigm for data fusion with applications in image processing. *Soft Computing*, 10(12):1169–1180, 2006.

AUTHORS DECLERATION

I DECLARE, WHILE AWARE OF CRIMINAL RESPONSIBILITY FOR STATING AN UNTRUTH, THAT I HAVE CARRIED OUT MY DOCTORAL DISSERTATION PERSONALLY AND INDEPENDENTLY, AND HAVE NOT USED ANY SOURCES OTHER THAN THOSE LISTED IN THE BIBLIOGRAPHY.

OŚWIADCZENIE AUTORA PRACY

OŚWIADCZAM, ŚWIADOMA ODPOWIEDZIALNOŚCI KARNEJ ZA POŚWIADCZENIE NIEPRAWDY, ŻE NINIEJSZĄ PRACĘ DOKTORSKĄ WYKONAŁAM OSOBIŚCIE I SAMODZIELNIE, I NIE KORZYSTAŁAM ZE ŹRÓDEŁ INNYCH NIŻ WYMIENIONE W PRACY.

. . . .

DATA I PODPIS