An Iot-Based Global Manufacturing Big Data Ecosystem For Predictive Maintenance

---

Thesis submitted to

La Trobe University

in total fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

By

WEN JIN YU

December 18, 2020

School of Engineering and Mathematical Sciences

College of Science, Health and Engineering

La Trobe University Victoria

Australia

Table of Contents

List of Figures

List of Tables

Abstract

Artificial intelligence, big data, machine learning, cloud computing, and the Internet of Things are methodologies, techniques and technologies which have driven the fourth industrial revolution. The digital revolution has transformed the manufacturing industry into smart manufacturing.

The problem addressed in this thesis is health state determination of the equipment in a global manufacturing plant for predictive maintenance. The rapid growth of various smart digital sensors connected through the Internet of Things permits real-time monitoring of the equipment in the whole manufacturing plant. However, to be truly capable of application within a global manufacturing plant, it requires data ingestion from multiple data sources, data cleansing including noise removal and data imputation of missing values, data management of big data and predictive analytics coupled with an effective integration framework and visualization capability that is able to deliver the results in near real time. This new technology drives new challenges in terms of multiple data sources ingestion and integration, a grand-scale connected network construction with data security and access protocol issues, data quality with considerable noise and missing values when gathered from industrial factories, efficient big data storage and management, smart interconnection with cloud services and real-time analytics requirements.

In this thesis, a big data ecosystem is developed for health state determination of the equipment for fault detection and diagnosis in predictive maintenance for real industrial big data from large-scale global manufacturing plants. The developed ecosystem provides a complete architecture which is used in industrial IoT-based smart manufacturing in an Industry 4.0 system. The proposed architecture overcomes multiple challenges including seamless integration of big data ingestion, integration, transformation, storage, analytics, and visualization in a real-time environment using techniques of Internet of Things, Big Data and cloud computing. The ecosystem is implemented using various technologies such as the data lake, NoSQL database, Apache Spark, Apache Drill, Apache Hive, OPC Collector, and other techniques. Transformation protocols, authentication, and data encryption methods are also utilized to address data and network security issues.

In a large-scale manufacturing system, not all kinds of failure data are accessible, and the absence of labels precludes all the supervised methods in the predictive phase. Therefore, unsupervised distributed machine learning models are designed and implemented for equipment health state monitoring and diagnosis.

The system architecture is validated through field testing using actual data collected from a real manufacturing plant and demonstrates the effectiveness and practicality of the developed ecosystem. The experiment results demonstrate the effectiveness of both the proposed IoT architecture and techniques which successfully provide an alarm warning several days before the fault happens.

In order to improve the real-time performance and localization of the appropriate processing at the sites, an advanced version of the architecture that incorporates edge computing combined with a centralized cloud computing facility is then also developed and validated.

The proposed big data ecosystem has been implemented in a real-time industrial production system and won the Best Industry Application of IoT at the BigInsights Data & AI Innovation Awards in 2018.

Statement of Authorship

Except where reference is made in the text of the thesis, this thesis contains no material published elsewhere or extracted in whole or in part from a thesis accepted for the award of any other degree or diploma. No other person's work has been used without due acknowledgment in the main text of the thesis. This thesis has not been submitted for the award of any degree or diploma in any other tertiary institution.

25 November 2020

Acknowledgments

This research work was supported by a La Trobe University Postgraduate Research Scholarship and a La Trobe University Full Fee Research Scholarship.

I would like to take this opportunity to sincerely thank my supervisors Prof. Wenny Rahayu and Prof. Tharam Dillon. Prof. Rahayu gave me the opportunity to start my PhD research which has changed my life. Her weekly meetings and chats have taught me how to be a better researcher and person. Prof. Dillon introduced me to the fascinating area of machine learning and I acquired a vast amount of professional knowledge from him.

I would also like to thank my manager Dr Fahed Mostafa. It was an honour to join his research project which gave me the opportunity to learn how to apply academic knowledge to a real industry and bring real business value to the company. I have learned a vast array of skills while working as a data scientist in his company. To my best friend, Yuehua Liu at La Trobe University, thank you for helping me relax and find ways to enjoy life during our PhD candidature. Our successful collaboration also showed me the importance of teamwork.

Finally, but most importantly, I would like to thank the support that my family has given me. My mum is always willing to offer anything I need even when we are in different countries, and I will always remember your sincere love. To my girlfriend, Ying Zhang, thanks for everything you have done to help me along this journey.

Publications

This thesis includes work by the author that has been published or accepted for publication. These publications are the own work of the author of this thesis, and the author has the permission of the publishers to reproduce the contents of these publications for academic purposes.

In particular, some data, ideas, opinions and figures presented in this thesis have previously appeared or may appear shortly after the submission of this thesis as follows:

Published Papers:

- **W. Yu,** T. Dillon, F. Mostafa, W. Rahayu and Y. Liu, "A Global Manufacturing Big Data Ecosystem for Fault Detection in Predictive Maintenance," in IEEE Transactions on Industrial Informatics, vol. 16, no. 1, pp. 183- 192, Jan. 2020. (Q1, Top journal, IF: 9,112, selected as a popular article)
- **Yu, W.**, Dillon, T., Mostafa, F., Rahayu, W., & Liu, Y. (2019, May). Implementation of Industrial Cyber Physical System: Challenges and Solutions. In 2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS) (pp. 173-178). IEEE.
- Y. Liu, T. Dillon, **W. Yu**, W. Rahayu and F. Mostafa, "Noise Removal in the Presence of Significant Anomalies for Industrial IoT Sensor Data in Manufacturing," in IEEE Internet of Things Journal, vol. 7, no. 8, pp. 7084-7096, Aug. 2020. (Q1, Top journal, IF: 9.936)
- Y. Liu, T. Dillon, **W. Yu**, W. Rahayu and F. Mostafa, "Missing Value Imputation for Industrial IoT Sensor Data With Large Gaps," in IEEE Internet of Things Journal, vol. 7, no. 8, pp. 6855-6867, Aug. 2020. (Q1, Top journal, IF: 9.936)

Under consideration:

- **W. Yu**, Y. Liu, T. Dillon, W. Rahayu and F. Mostafa, "An Integrated Framework for Health State Monitoring in a Smart Factory Employing IoT and Big Data Techniques, " Submitted to IEEE Internet of Things Journal. (Current stage: second round revision, Q1, Top journal, IF: 9.936)
- Y. Liu, **W. Yu**, W. Rahayu, T. Dillon and F. Mostafa, "Sensor Selection Using unsupervised learning for efficient fault detection in Industrial IoT, " Submitted to IEEE Internet of Things Journal.
- **W. Yu**, Y. Liu, T. Dillon, W. Rahayu and F. Mostafa, "An IoT-based Real Time Framework for Fault Analytics Embedded with Edge Computing Technique," Submitted to IEEE Transactions on Industrial Informatics.

Chapter 1

Introduction

This chapter presents the background and motivation for the research presented in this thesis. It outlines the context and challenges of predictive maintenance in manufacturing, predictive maintenance in Industry 4.0 and the new challenges of smart manufacturing. The structure of the thesis to address these challenges is then presented in this chapter.

1.1 Traditional Maintenance Practice

For manufacturing, reliability is defined as the probability of a machine or system consistently performing its intended function without degradation or failure over its useful life cycle. Traditional maintenance has been performed using two broad approaches namely:

a) Maintenance is carried out after failure or severe performance degradation of the equipment.
b) Preventative scheduled maintenance is carried out periodically based on understanding the average statistics of failure or the manufacturers' recommendation.

These approaches were found to be unsatisfactory because the maintenance on failure approach often led to collateral damage of surrounding equipment and led to long down times especially in the case of critical equipment which could result in huge economic losses arising from lost production. It was also complicated by the fact that there might be long delays in obtaining specialized parts for the repair. This is the reason, currently, most manufacturing industries engage in preventive maintenance, that is, undertaking maintenance depending on the average or expected life statistics of the equipment as their maintenance strategy instead of performing maintenance interventions after equipment loses performance.

The preventative scheduled maintenance can reduce the chance of some failures. However, as it is based on average statistics of failure, it did not prevent all failures as different pieces of equipment deteriorated at different rates instead of just the average rate.

Therefore, scheduled maintenance may be done sometimes when it was not necessary for a particular piece of equipment leading to unnecessary downtime and sometimes it was not done when a piece of equipment was likely to fail leading to catastrophic failure. In addition, it also required a considerable inventory of parts to be stored. Although the scheduled maintenance approach can prevent failures, unnecessary corrective actions are usually performed which causes an inefficient utilization of resources.

The advent of sensors in conjunction with computers created the opportunity for directly monitoring the condition of equipment to determine the health state of equipment which led to the idea of predictive maintenance which is discussed in the next section.

1.2 Predictive Maintenance in Manufacturing

Along with the rapidly increasing requirements of machine or system quality, reliability, and health and safety in manufacturing, there is a growing demand for early machine or system health monitoring and assessment, and machine health prediction to reduce unscheduled down time. The timely detection of potential breakdowns and immediate repairs contribute to save energy, reduce equipment costs and reduce lost production time. More importantly, it may prevent possible collateral damage in other components linked to the damaged one and increase safety. In a report released by the Department of Trade and Industry (DTI) in the UK, it was found that up to around 50% of the total quality costs were spent on failures and defects because of insufficient funds devoted to maintenance management [1]. This is the reason why predictive maintenance is gradually playing a central role in industry control systems and attracting more attention in manufacturing, business and research.

Key industrial manufacturers have invested in predictive maintenance to maximize machine parts and equipment uptime and deploy maintenance more cost effectively [2]. Unlike traditional preventive maintenance, predictive maintenance is designed to monitor the actual condition of the equipment to alert the system in advance and determine whether corrective maintenance will be required. Maintenance is only performed when the system conditions concerning its health show signs of decreasing performance or upcoming potential failure instead of conducting regularly scheduled maintenance.

Fault detection and condition monitoring are critical components of predictive maintenance and can potentially eliminate catastrophic equipment failures in industrial manufacturing. Among the system control measures, properly identifying the usual and unusual patterns of these machines through sensor data can inform engineers about the machine condition and thus assist them to inspect for the potential defects and then call for maintenance if necessary. Multiple sensors are attached to different machinery components, and these sensors provide multivariable data continuously. Our main goal is to detect the significant anomalies from these observed data by multivariate analysis instead of univariable monitoring, using anomaly detection technology [3]. Anomaly detection is a field of predictive modelling which involves predicting unknown or abnormal patterns, events, or failures using collected data. In manufacturing, once such patterns are detected, an alarm is triggered as an early warning to engineers so that they can make better decisions and arrange proactive rather than reactive maintenance. In this way, industrial manufacturers can obtain great savings in terms of the operating costs due to the prevention of collateral damage to other components and the shutdown of the plant for extended periods and the associated lost production. Great effort has been expended on developing the existing anomaly detection methods for predictive maintenance [4]–[10] and they are generally characterized by the following:

1) an equipment-specific algorithm, such as bearing;
2) high performance with low efficiency using complex ensemble machine learning techniques;
3) small-scale laboratory-collected data for experiments.

1.3 Predictive Maintenance in Industry 4.0

However, with the availability of the grand-scale use of measurement devices, the Internet of Things (IoT) is continuing to grow and IoT-based predictive maintenance is becoming vitally necessary in manufacturing. IoT involves intelligent digital sensors and networking technologies, where all the equipment in factories can communicate with each other and cooperate with human beings through the connected network according to an agreed protocol. Previously, the major form of communication on the Internet was among humans (i.e. human-to-human). IoT extends the capabilities of the Internet to enable machine-to-machine communication. The work in [11] reveals such an evolutionary progress of connection and communication, which discusses the progression of communication from human-to-human to thing-to-things (also known as machine-to-machine).

Traditional sensors have limited sensing capabilities and are mostly used to inspect vibrations. Most of these sensors are bulky, heavy, intrusive, or energy-consuming [12]. They are not able to be deployed in machines that have not been initially equipped with such a capability. In addition, traditional sensors generally work with cables, thus they are inconvenient to install and operate. Cameras, lasers and infrared sensors are the common choices for high-resolution approaches, but suffer from the line-of-sight issues.

In the past few years, wireless and battery-free sensing, e.g., radio-frequency identification (RFID), which leverages backscattered radio-frequency signals to carry information, has gained much attention due to its low-cost, nonintrusive, and easy deployment properties [13], thus making it a promising technique for sensing. RFID-based sensing has been used in various cases, e.g., orientation detection [14, 15, 16], humidity sensing [17], vibration sensing [18, 19], motion detection [20], touch sensing [21], eccentricity detection [22], liquid leakage detection [22, 23], temperature detection [24], etc. Boosted by the recent advances in RFID [20, 22, 25, 26, 27, 28, 29], smart sensors have achieved further developments.

It is the emergence of smart sensors that fosters the IoT and IoT-based smart manufacturing in Industry 4.0. The advances of modern sensor technologies enable the sensibility of more measurements to be made such as temperature, humidity, pressure, flow, speed, leakage, etc. Real-time data collection is based on key technologies such as RFID, which has been already used in production systems in many companies such as Loncin Motor Co., Ltd. [30], Huaiji Dengyun Auto-Parts (Holding) Co., Ltd. [31], Guangdong Chigo Air Conditioning Co., Ltd. [32]. By leveraging these smart sensing technologies in manufacturing, a large volume of data from machines can be collected and processed. With the proliferation of manufacturing data, the health condition of a machine can be derived through big data processing and streaming data analytics, and other insights could be gained to support the smart decision-making process in smart manufacturing.

In addition, IoT is considered a complete system instead of just the sensor devices. It contains all the smart things or objects equipped with sensors, networking and processing technologies integrated and working together to provide an environment in which smart services are taken to the end users [33]. The IoT is now given a broader concept, being envisioned as a convergence of cutting-edge technologies such as ubiquitous wireless standards, big data analytics, machine learning, cloud computing and so on. New market requirements and emerging IoT technologies are moving the manufacturing environment toward smart factories, achieving the evolutional transition from computer control and automation in the digital era of Industry 3.0 to a higher level of IoT-based smart manufacturing in Industry 4.0 [34]. Cyber-physical systems (CPS), IoT, cloud computing, big data, and advanced analytical techniques are the key technologies of Industry 4.0 [35, 36]. A CPS is a mechanism through which physical objects and software are closely intertwined, enabling different components to interact with each other in a myriad of ways to exchange information [37, 38]. Smart manufacturing in Industry 4.0 uses IoT and CPS technologies in physical processes to measure and monitor real-time data from across the factory, which gives rise to unprecedented levels of data production. Therefore, smart manufacturing is expected to manage the demands of an exponential increase in data production, being able to transmit and store such data in an efficient way. The existing predictive maintenance methodology is not a good fit in real industrial manufacturing plants with IoT big data and is incapable of addressing the new challenges of smart manufacturing brought about by the evolution of Industry 4.0.

1.4 New Challenges of Smart Manufacturing

As IoT can serve as a powerful tool to use ubiquitous sensors and microprocessors across the entire factory or even factories from different sites across the country, which generates a huge volume of data well beyond that of data of a traditional size, the new generation of smart manufacturing is marked by big data. Recognizing and digging the meaningful patterns from big data is extremely important to businesses, providing insights and suggestions for decision-making. Prior to the era of IoT, big data have already been a popular topic and a large number of studies have been conducted on big data. Several papers [39, 40, 41, 42] have given a general definition of the nature of big data from different aspects.

- Volume: data volume is a determining factor to consider data as big data or not. High-density sensors installed in the IoT system result in a large volume of data.
- Velocity: the rate at which the data is produced is high. In IoT, the velocity of data makes it possible to provide real-time monitoring.
- Variety: big data comes in different forms, frequency, and types. IoT sensors can generate a wide variety of data with a mixture of structured, semi-structured, and unstructured data. For example, in the manufacturing monitoring scenario, machine name and location, and environmental readings are collected in the form of sentence and number, respectively, which are then saved in text. Sometimes, audio and video are also collected as the data source.

- Veracity: this refers to data quality, which can directly influence the analytic results. Since IoT is intended for real industry and business, the veracity of data should be given special attention.
- Value: big data brings rich information and deeper insights. Discovering the value hidden inside big data by leveraging data mining and machine learning approaches is our ultimate objective.

This nature of big data triggers research motivations on a new form of computing paradigm for efficient big data processing, since traditional computing approaches take an intolerable time to deal with big data. In a big data environment, the data are too complex for conventional data analytic software [43]. The work in [44] highlighted that big data analysis requires more scalable analytical algorithms and techniques to return well-timed results. Large infrastructure and additional applications are necessary to support data parallelism. Over the past few years, a range of platforms and technologies have been developed that facilitate the efficient storage and processing of big data, including the commonly adopted Map Reduce [45], Hadoop [46] and Spark [47]. The work in [48] provides an in-depth analysis of different platforms available for performing big data analytics, assessing the advantages and drawbacks in terms of scalability, data I/O rate, fault tolerance, real-time processing, data size supported and iterative task support. In addition to this, a large body of research has been dedicated to the development of big data analytic platforms such as 1010data [49], cloudera data hub [50], SAP-Hana [51], HP-HAVEn [52], Hortonworks [53], Pivotal big data suite [54], Infobright [55] and so on.

The recent evolution in sensing and computing technologies has opened new avenues for big data processing [56]. Especially with the advancement of IoT, a big data environment has gradually taken shape in smart manufacturing since smart sensors have streamlined the IoT data. The question is whether the collected IoT data can be processed properly in order to provide the right information for the right purpose at the right time [57]. To date, big data analytic technologies have matured over the past few years. As reported in [58], pioneers such as Google and Tesco are not the only ones that have benefited from big data analytics, and an increasing number of manufacturing firms such as General Electric (GE) are also trying to optimize manufacturing processes in a big data environment [58]. The work in [59] proposes a big data analytical framework for IoT applications, focusing on the volume and velocity challenge.

In the area of IoT, the term big data has become a buzzword, and thus is often overused and misunderstood [60]. When it comes to big data related to IoT-based smart manufacturing, it is not just about processing big-sized data using machine learning algorithms in a parallelized way or big data analytics platforms. Rather, it concerns a system involving the 5Vs of big data (volume, velocity, variety, value, and veracity), data transformation flow, and data security infrastructure. The resulting challenges have been discussed in many studies [35, 36, 61].

When IoT-enabled factories rapidly scan many millions of unstructured streaming data items in different formats from a multitude of diverse sources, real-time integration in the industrial production process should be ensured in order to provide data for analytics and optimize the resources in the production chain. Smart manufacturing in Industry 4.0 uses IoT and CPS technologies in physical processes to measure and monitor real-time data from across the factory, which gives rise to unprecedented levels of data production. Therefore, smart manufacturing is expected to manage the demands of the exponential increase in data production, being able to transmit and store such data in an efficient way. Streaming data refers to the data generated within tiny intervals of time. In terms of big data processing, it concerns the ability to process without a strict restriction on time, while streaming data requires fast processing. A data analytic model should finish the processing task on the current data point and be ready before the next data points arrive. Therefore, beyond big data analytics, IoT data also calls for fast and streaming data analytics to support applications with high-speed data streams and time-sensitive response requirements

Big data management and the connectivity between different devices result in another challenge in terms of information protection and privacy issues and internet security. Furthermore, traditional fault detection methods are not suitable for smart manufacturing, as we move toward cloud-based distributed and real-time processing for big data in Industry 4.0. Performing predictive modelling for fault detection and diagnosis in IoT big data is a difficult task, as normally the data is unlabelled and there is a lot of noise in the data collected in a real manufacturing industry. The conventional supervised machine learning techniques are unable to completely address these issues without learning with specific labels. Most machine learning techniques and open source tools also have problems in terms of scalability, ease of use, extensibility, and generalization ability when facing big data issues [60]. The procedures for predictive maintenance based on the big data system are becoming more complex when incorporated with cloud computing [62, 63].

Under the new demands of IoT-based smart factories, some researchers are gradually paying more attention to the big data system. Manogaran et al. [64] propose a new architecture for the implementation of IoT to store and process scalable sensor data for health care applications, where Apache Pig, Apache HBase, and cloud computing are employed for securing integration and the MapReduced-based prediction model is embedded to predict heart disease. For more complex manufacturing with complicated sensor networks, heterogeneous multiprovider service integration, a wider array of monitoring equipment, and bad quality data, research, and applications in the field of industrial manufacturing big data are inadequate. Initially, several papers discussed the challenges and proposed an architecture design [34, 35, 36]. Then, most researchers focused on elaborating separate parts of the big data system, including manufacturing data collection [65, 66, 67], big data storage [68], big data cloud computing [69, 70], predictive maintenance [4, 5, 6], and provided a preliminary theoretical basis. In [71], data collection and cloud-based data processing is performed together in the system, and a prototyping platform is analysed. However, they lack an integrated approach of big data management and infrastructure. In addition, it did not test the proposed architecture in a real manufacturing plant which itself introduces its own difficulties and requirements. In this thesis, we introduce a more integrated IoT-based big data ecosystem, where each part is designed using the proposed methods and the different parts which address different functionalities are connected in a seamless fashion and cooperate in an appropriate way rather than a simple concatenation. Additionally, we provide the application programming interface (API)-oriented implementation guidelines, which were successfully deployed in a real industrial setting in a manufacturing company.

1.5 Structure of the Thesis

The remainder of this thesis is organized as follows.

Chapter 2 provides a comprehensive review and detailed discussion of the major existing works to the field and how they relate to each other.

Chapter 3 presents a definition of the terms and concepts used in the thesis and provides a clearly defined statement of the problem addressed in this thesis including the need for an architectural framework for predictive maintenance.

Chapter 4 provides an overview of the solution to address the problems and outlines the main contributions of the thesis.

Chapter 5 details the proposed IoT-based big data ecosystem for predictive maintenance, which includes the methods of big data ingestion, big data management, data pre-processing and predictive modelling.

An industrial case-based experiment setup, implementation results, performance evaluation and comparison to other technique are presented in Chapter 6.

Chapter 7 details the proposed optimized architectural framework by embedding edge computing technique and distributed stacked sparse autoencoder algorithm.

Chapter 8 concludes this thesis by first providing a summary of the work. Then, we highlight interesting areas that complement the work of this thesis and future research that could be carried out.

Chapter 2

Literature Survey

This chapter presents a discussion of the previous research and recent developments on smart maintenance in manufacturing and provides a complete picture of how it has evolved from traditional preventive maintenance to intelligent predictive maintenance. It also provides an evaluation of previous research on predictive maintenance and identifies gaps in the research.

2.1 Maintenance in Manufacturing

Four industrial revolutions have triggered the paradigm changes in manufacturing, from hand production, mechanization with the use of steam and water power, mass production on assembly lines, automation and digitalization using information technology (IT), to IoT-enabled smart manufacturing. Such changes have also led to the evolution of manufacturing data, thereby having an unprecedented impact on how we monitor and maintain the manufacturing process.

Maintenance approaches generally fall into three groups: corrective maintenance (CM) which is also known as Run-to-Failure (R2F) as discussed in [6, 72], preventive maintenance and predictive maintenance.

R2F or CM: In terms of CM, engineers do not perform maintenance until machines break down. Only when they are informed about the failure of machines is appropriate maintenance undertaken such as repair or even replacement. In this sense, the cost is relatively high including the unplanned production losses and huge maintenance cost of severe damage under CM.

Preventive maintenance, time-based maintenance or scheduled maintenance: To make the maintenance more proactive rather than reactive, preventive maintenance attempts to schedule regular maintenance based on a prefixed basis no matter whether failure has occurred or not, to lessen the likelihood of machine failure in the future. Such scheduled maintenance is able to greatly reduce the cost both in terms of production loss and repair expenditure in comparison to unplanned maintenance such as CM (see [73]). More importantly, the safety of the workers could be significantly improved. However, it also results in unnecessary actions which causes an inefficient utilization of resources.

The first preventive maintenance approaches are introduced in [74, 75, 76]. Since then, more approaches have been proposed. The work in [77, 78] consider an item fails when it wears beyond a certain breakdown threshold. So, the researchers designed a maintenance scheme based on the wear value where the item is preventively replaced if the measured wear at periodic inspections exceeds a wear limit; otherwise it continues to be used. Obviously, the choice of the inspection scheduling and the value of the threshold influence the economic performance of the maintenance policy. These are also the main decision variables that could optimize the maintenance process to minimize the total maintenance cost of the system. To address this issue, several strategies are discussed in papers [79, 80, 81, 82, 83].

Motivations for predictive maintenance: The maintenance strategies most industries use now are scheduled maintenance and sometimes R2F maintenance. The problems with R2F maintenance is that we also get collateral damage. When one part fails, many other things related to it also fail. If we use this kind of technique for solving the problem, maintenance becomes far more expensive because of the collateral failure and also it is not possible to identify the root cause of the failure and which part needs to be replaced because the system has failed. This logistically delays the acquisition of the part which is needed and especially in the manufacturing industry, parts may need to be purchased from overseas. So, R2F maintenance results in large losses in terms of repair, maintenance and production.

Scheduled maintenance is undertaken based on the engineers' advice. Some manufacturers use scheduled maintenance in their systems, checking all the equipment at a fixed period of time. By doing this, scheduled maintenance can, 1) stop some potential failures, thus greatly reducing production loss compared with R2F maintenance. Sometimes collateral failures that may occur in R2F maintenance can be catastrophic and scheduled maintenance can potentially prevent catastrophic failures. Scheduled maintenance can also 2) avoid logistic delay because engineers always prepare all the parts that they might need to replace in advance before scheduled maintenance occurs.

However, scheduled maintenance also has the following limitations: 1) when scheduled maintenance commences, the whole production system may need to be shut down for the inspection, which still leads to production loss; 2) engineers need to maintain all the equipment which also includes equipment that may not need maintenance, which is a waste of resources. 3) engineers may overlook a piece of equipment which is currently not causing problems but it may actually be at a point where is it about to break down.

Predictive maintenance: To overcome the above limitations of R2F maintenance and scheduled maintenance, condition monitoring is utilized to identify possible occurrences of faults. Predictive maintenance utilizes condition monitoring to identify possible faults at an early stage before the faults actually occur. Broadly speaking, predictive maintenance services mainly include machinery condition monitoring, data analysis and diagnosis, prognosis, and maintenance planning over the Internet [84]. Through the data generated from the sensors that are installed in the monitored machines, upcoming potential failures can be predicted and detected in advance through sensor data analysis. Then, maintenance is only performed before the predicted faults happen. As predictive maintenance is carried out depending on the condition of the process or a piece of machinery [85], it is also referred to as condition-based maintenance in [86].

In this case, manufacturers start to perform condition monitoring on critical pieces of equipment. This involves collecting sensor data from critical pieces of equipment, assessing the sensor data, and indicating if an alarm has occurred. Condition monitoring of critical equipment, which is the traditional approach to conditional monitoring, often fails to detect collateral failure. Traditional conditional monitoring or predictive maintenance assumes failures will occur on a single machine. But the condition monitoring of critical equipment itself for predictive maintenance is not enough; adjacent equipment that connects or feeds into it is also critical because the breakdown of these pieces of equipment may cause critical equipment to fail, in which case only monitoring the critical equipment does not help engineers identify the failures that occur in their adjacent equipment. To address this issue, manufacturers try to move predictive maintenance on the critical equipment to manufacturing plant-wide predictive maintenance.

Predictive maintenance of individual critical equipment:

Previously, the development of sensor technologies is in its infancy stage. Many physical metrics cannot be measured by sensors with limited sensing capabilities. Reliability-centered maintenance was initially proposed to concentrate maintenance efforts on the components that are the most critical for safety and business [87]. The collected data or information that could be used to evaluate the health state of machines for fault detection are very limited due to the limitations of sensor technology. During that period, anomaly detection, fault diagnosis and prognosis-related algorithms were the key components for predictive maintenance solutions. At the early stage, they are carried out manually by simply using a threshold. Later on, engineers started to use better techniques and algorithms for critical equipment which is discussed in section 2.2.3 and the predictive maintenance of individual critical equipment can be also referred to as traditional predictive maintenance.

Manufacturing plant-wide predictive maintenance:

Some of the early works only use a hardware connection network to send alarms from all the equipment to a control center for the engineers to view. This is usually called a SCADA system and is the initial plant-wide system. To date, manufacturing is embracing a convergence of cyber and physical systems. The advance of modern sensor technologies enables the sensibility of more metrics with stricter accuracy guarantees. New forms of communication and networking are upgraded for more efficient data transmission to meet the requirement of timeliness. With the rapid growth in the number of sensors, wireless communications and advanced signal processing techniques, the IoT has significantly expanded in recent years [13]. The advent of IoT provides manufacturing plant-wide predictive maintenance a more promising future. Under such circumstances, Industry 4.0 is proposed which essentially equips traditional manufacturing with cyber physical systems and IoT to start a new industrial revolution, aiming at creating smart factories.

Compared with SCADA, IoT allows objects to be sensed or controlled remotely across different networking infrastructures, and it creates opportunities for more direct integration of the physical world into computer-based systems. This results in improved efficiency, accuracy and economic benefit and also reduces the need for human intervention. Industry 4.0 provides an environment in which emerging trend automation and data exchange in manufacturing technologies are allowing for a shift from SCADA to an IoT implemented one.

Manufacturing is becoming smart in Industry 4.0 systems. The equipment in manufacturing in IoT is very flexible. Companies are increasing their use of IoT technologies to capture data at all stages of a product’s life. These data can range from material properties and the temperature and vibrations of equipment to the logistics of supply chains and customer details [88]. By monitoring big data, manufacturing can achieve intelligent products, intelligent processes, intelligent machines and so on [89], where predictive maintenance is considered one of the crucial use cases for smart manufacturing. Thanks to the availability of IoT big data, predictive maintenance is upgraded from purely providing fault detection functions only for the most critical components to building a complete monitoring framework for the whole factory.

Furthermore, the evolution of Industry 4.0 provides more convenient support for the wide development of predictive maintenance in practice. In recent years, predictive maintenance in modern industry has gained extensive attention, being considered to have great potential and practical significance. Compared to the previous stage, the new stage of predictive maintenance poses new requirements and challenges as follows:

- Seamless monitoring for digital twin purposes.
- Deployment feasibility in real industry.
- Predictive maintenance is more than fault detection-related machine learning as also involves an integrated IoT-big data ecosystem.
- The proper integration between each element in an IoT-big data ecosystem.
- The techniques inside each element in the ecosystem need to accommodate big data.

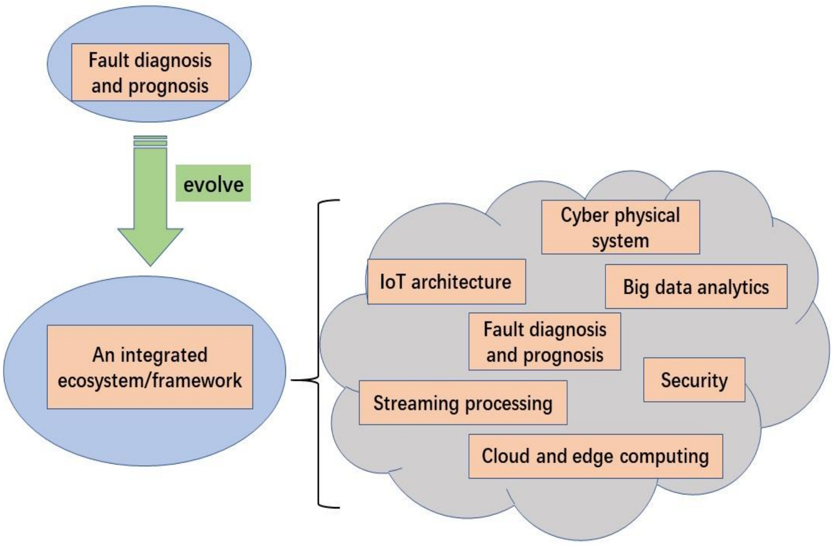


Fig. 2-1: The evolution of predictive maintenance due to the emergence of Industry 4.0

In comparison to previous corrective maintenance and preventive maintenance methods, predictive maintenance predicts failure sufficiently ahead of time so that engineers can arrange appropriate maintenance as soon as possible. Most importantly, the manufacturing process often follows assembly line production, so it is crucial to monitor the whole factory since any failure in the assembly line might result in a domino effect. With the use of IoT big data, predictive maintenance in Industry 4.0 can achieve the fine-grained monitoring of a wide array of equipment across the factory, thus completely addressing this issue. In light of the above analysis, the key components of predictive maintenance in the new stage are not just the diagnosis and prognosis-related machine learning algorithms, it involves an integrated ecosystem, including the IoT architecture, cyber physical system [90], big data analytics, stream processing technologies [89], cloud and edge computing [91, 92, 93] and so on, where the detection, diagnosis and prognosis are only a part of big data analytics. Therefore, predictive maintenance has significantly evolved and current predictive maintenance is distinctively different to traditional maintenance, as shown in *Fig. 2-1*.

We review the related work on predictive maintenance at every stage during the development, which is referred to as “traditional predictive maintenance” and “plant-wide predictive maintenance” in the following sections. The existing approaches improve predictive maintenance considerably but there are still some research gaps. The detail is discussed in section 2.2.3 and a summary is given at the end of this chapter.

2.2 Traditional Predictive Maintenance Approaches

As mentioned earlier in Traditional Predictive maintenance the emphasis was on using sensors to collect information on a single piece of equipment and display and process the collected information at the site of the piece of equipment using a computer. Various machine learning algorithms have been explored for a wide range of applications in predictive maintenance for traditional predictive maintenance. Previous works on traditional predictive maintenance mainly focus on improving the performance of such applications in terms of accuracy. The initial approaches use simple methods like threshold. A lot of research has been carried out, particularly in relation to the prediction-side. Generally, predictive maintenance involves four types of tasks, detecting faults before they occur (fault detection), examining the root cause and reporting the reason for its occurrence (fault diagnosis), predicting the remaining useful life of equipment (fault prognosis), and determining appropriate maintenance decisions. However, most research papers focus on fault detection, diagnosis and fault prognosis, while only a few studies discuss the last task. The work in [94, 95] addresses the post-prognostics issue but it is based on the assumption that the prognostics information of the system is already known. A wide range of methods have been developed for detection, diagnostics, and prognostics and these are extensively discussed in [96, 97, 98, 99, 100, 101, 102]. This section will review the state-of-the-art approaches and provide a more comprehensive outlook of the algorithms involved in predictive maintenance, focusing on these three types of tasks.

2.2.1 Fault Detection and Diagnosis

With the growth of automation in modern industry, the large number of sensors and actuators installed in complicated large-scale industry generate a large amount of data. As a result, traditional model-based approaches, which require prior knowledge of the industry process or physical model obtained from the first principles, become impractical [103]. In large-scale systems, understanding component operation is not straightforward and is so complex that developing an accurate physical model is prohibitively expensive [97]. However, the huge amount of industrial data brings opportunities for data-based approaches. Compared with the well-developed model-based approaches, data-based techniques provide efficient alternative solutions for different industrial issues under various operating conditions. This scenario has triggered numerous research motivations on data-based methods [103, 104] over the past few years, while only a few works focus on model-based approaches.

Model-based methods deal with the exploitation of a mathematical model representing the behaviour of physical components, including their degradation [105]. In [106], fault detection is realized based on a residual current vector generated by the difference between the measured stator currents and the stator currents under the steady-state condition, which are formulated by different models separately. The method proposed in [106] is specifically designed for permanent magnet synchronous machines (PMSMs).

Data-driven methods extract the behaviour patterns hidden behind the observed data to predict the health state of the industrial system and the embedded machine in the future. Artificial intelligence-assisted data-driven approaches are able to deliver intelligent results without an in-depth physical understanding of the system. Data-based approaches can be roughly categorized into two types i.e., data-driven feature and model construction which is built on the feature to make the prediction. Initially, due to the limited sensing capabilities of wireless signals, raw signals alone are not adequate to represent and identify a fault. This commonly adopted time, frequency and time-frequency domain analysis alternatively provides a broad range of measures. For example, time domain analysis includes the acceleration amplitude peak-to-peak, root mean square (RMS) [107], high-order statistics [108] and short impulse methods [109]. Frequency domain analysis includes envelope analysis [110, 111] and high-order spectral analysis [112], short-time Fourier transform [113], wavelet transforms [114, 115], and Hilbert-Huang transforms [116, 117].

In the work in [118], the raw vibration signal is first decomposed in the frequency domain to obtain a set of subsignals. Then several specific fault features are extracted and selected based on cross correlation and spectral kurtosis analysis, followed by PCA to reduce the redundancy of fault features. Finally, according to the new features, faults are identified using the KNN technique. The work in [119] utilizes vibration readings to address the predictive diagnostic problem through signal processing techniques in power supply systems. Prieto et al. [120] apply a hierarchical neural network for bearing fault classification based on the extracted statistical-time features. Yang et al. [121] trained a random forest classifier for induction motor fault diagnosis based on the extracted energy features at various specific frequencies. He et al. [122] constructed a k-nearest neighbour (KNN) classifier based on time-domain features extracted by empirical mode decomposition for bearing fault detection. Regardless of which kind of machine learning models is applied, it is shown that the representation determines the upper-bound performance of the machine learning algorithms [123]. To learn a more discriminative feature space, a range of feature extraction or selection methods such as principal component analysis (PCA) [124], factor analysis [125], Fisher discriminant analysis [126], distance measures [127], etc., have been explored to transform the original features into an informative feature space that could better represent machine conditions.

Such conventional methods are labour-intensive and application-specific as they usually depend on expert knowledge and handcrafted features. Motivated by this scenario and inspired by the success of deep learning, deep learning is considered to be a promising tool to achieve a breakthrough in fault detection and diagnosis. In particular, deep learning [128] has attracted much attention and has demonstrated outstanding performance in various fields, including speech recognition, audio recognition, computer vision, natural language processing, games (e.g. Alphago), bioinformatics, etc., thus gaining popularity in the data analytics of predictive maintenance. The rationale behind deep learning is its capability of extracting hierarchical representations from input data through multiple layers of nonlinear transformations. Over time, the deep learning-based approaches developed into the following two branches: (1) conventional feature extraction and engineering are replaced by automatic feature learning through a deep learning approach. Then, the derived features are fed into another deep learning model to make predictions in relation to the condition of the equipment and the process is conducted in two steps; (2) the aforementioned two steps, namely feature representation learning and fault prediction, are integrated into one end-to-end learning structure and the parameters of the deep learning model are trained jointly.

The success of the machine learning methods generally depends on good data representation through representation learning, which enables the underlying explanatory factors to be captured for the observed input. Examples of the state-of-the-art methods that could be applied for representation learning include autoencoders, deep belief networks [129], sparse coding [130], etc.

A powerful and prevalent technique in various application areas in recent years, convolutional neural networks have also been used in fault detection and fault diagnosis in various aspects, such as bearings [131, 132, 133, 134], gearboxes [135, 136], wind generators [137], rotors [138], and so on. Deep belief networks have been investigated for aircraft engines [139], reciprocating compressors [140], rolling element bearings [141, 142], high speed trains [143, 144], wind turbines [145], reciprocating compressor valves [140] and so on. Autoencoders have always been used for unsupervised feature learning, and the learned features are then fed into a classification model for fault diagnosis. In [146], extreme learning machines based on the features learned through the autoencoders are developed for wind turbine fault classification. The work in [147] proposes a deep wavelet autoencoder to capture the signal characteristics by replacing the traditional sigmoid nonlinear activation function with the wavelet function. Based on the learned features, an extreme learning machine is adopted as the classifier to identify the bearing faults. In the work in [148], various activation functions are employed as the hidden functions to design a series of autoencoders for different characteristics and the softmax classifier is used for the output. Lei et al. [5] propose the use of sparse filtering to learn features in an unsupervised way and softmax regression is implemented for multiclass fault classification. It is based on the vibration signals and tested on the dataset collected from motors. Different variants of autoencoders are also employed, such as sparse autoencoders [149, 150], stacked denoising autoencoders [151, 152], contractive autoencoders [153] and so on. As discussed previously, deep belief networks can also act as a representation learning tool. In the work in [145], features are firstly extracted from both acoustic emission signals and the vibratory signal using the statistical parameters of the wavelet packet transform (WPT). Then, two DBMs are developed for deep representations of the statistical parameters. Based on the representation features, a random forest is employed to classify different operational conditions. The work in [154] suggested utilizing DBM to learn the pattern information from signals in time, frequency and wavelet modalities separately, thus integrating wide modalities with deep learning. Based on the learned deep representations of raw features, a support vector classification is applied for gearbox fault diagnosis.

Hu et al. [155] adopted deep neural networks based on the vibration amplitudes to conduct fault diagnosis for high-speed trains to obtain a higher diagnostic accuracy. To learn more discriminative representations, some researchers focused on the pretraining of deep neural networks (DNNs). The work in [156] builds a DNN with multiple hidden layers, then pre-trains the DNN layer by layer with a stack of autoencoders. The trained parameters of the autoencoder are used to initialize the DNN and the BP algorithm is implemented to fine-tune the parameters of the DNN by minimizing the prediction error. The proposed model is tested on rolling element bearings and planetary gearboxes. In the work in [157], a deep coupling autoencoder is proposed which consists of a Gaussian Bernoulli restricted Boltzmann machine (GBRBM), RBM and a coupling layer. The training process is carried out using an efficient layer-by-layer greedy learning strategy. The model is followed by or combined with a softmax classifier for fault diagnosis. In the work in [158], adversarial learning is introduced as a regularization into convolutional neural networks (CNNs) to make the feature representation robust, boost the generalization ability of the trained model as well as avoid overfitting with a small-sized labeled samples. In addition, LSTM is explored in [159] for the fault diagnosis of wind turbines as well as other energy systems.

In many real cases, the data that could be used for training is insufficient. To solve this issue, the related data are transferred in a certain way such that they can be used in the training process. If the distribution of the source domain data (on which the model is trained) is different from the distribution of the target domain data (where the learned model is actually deployed), it will lead to performance degradation. Transfer learning [160] provides a promising idea to address the problem. Lu et al. [161] proposed a novel deep neural network with domain adaptation for fault diagnosis. The basic DNN is incorporated with the weight regularization term and the maximum mean discrepancy (MMD) regularization term in the objective loss function to learn the transferrable features based on the source domain data and normal category data in the target domain. Then a standard support vector machine is built for the diagnosis task. The work in [162] integrated the MMD term into a basic autoencoder model, in addition to which a softmax classifier is assigned as the output layer for prediction. The training process is as follows: the first step is to pretrain the autoencoder and softmax classifier in a separate manner; the second step is to fine-tune the whole neural network structure. In [163], a distance metric learning is embedded into a convolutional neural network for fault diagnosis to make the trained deep model robust against environmental noise and variations in the working condition. The work in [164] tries to identify the health states of real-case machines (BRMs) by utilizing the diagnosis knowledge from laboratory machines (BLMs). The proposed model is trained by jointly minimizing three regularization terms that includes not only the basic prediction error but also the multi-layer MMD.

2.2.2 Fault Prognosis

Fault detection and fault diagnosis tasks identify whether a fault occurs or not and if it does, which type of fault it is, based on the online monitoring data. In contrast, fault prognosis models temporal behaviour based on historical data, predicting when maintenance should be performed in the future. Fault prognostics can also be done using two main approaches: model-based [165, 166, 167] and data-driven.

Model-based methods set up a mathematical or a physical model that describes the degradation of a system. In the work in [168], the Paris model is employed to simulate the degradation process. Of all the model-based research, the exponential model is one of the most widely used methods. It was first introduced in [169] and its variants were reported in the RUL prediction and health management [170, 171, 172, 173]. The work in [174] proposes an improved exponential model with particle filtering for the RUL prediction of rolling element bearings. Data-driven approaches are strictly dependent on the condition monitoring data acquired from the sensors. From the pool of the existing methods, most efforts are directed to data-driven research. Data-driven methods are suitable for systems where it is easy to obtain monitoring data, whereas for the model-based approaches, it is difficult to develop an accurate model for complex real-world systems. If any, they are generally object-specific, thus it is difficult to generalize them to other machines or systems.

The work in [175] adopts artificial neural networks that take the features processed by the DNA-based computing technique as the inputs to predict the progression of tool-wear with respect to machining time. The work in [105] uses a wavelet packet decomposition tree technique to extract the coefficients of the first six levels of decomposition and the extracted features are then fitted to train a support vector regression (SVR) model to assess the current health state and calculate the RUL of cutting tools. A weighted hidden Markov model and a Gaussian process regression model were used in [176] and [177] for RUL prediction. In addition, a least-squares support vector machine (LS-SVM) [178], manifold regularized logistic regression [179], a neural network approach [180, 181], neuro-fuzzy-based approaches [182, 183, 184] and various Kalman filter-based methods [185, 186, 187] have been also used for life prediction. In the work in [185], an enhanced Kalman filter and an expectation-maximization algorithm are used to estimate the RUL of the bearing adaptively.

Different to fault detection and diagnosis, the only difference in the output layer is the prediction tasks in relation to fault prognosis. So, fault prognosis has also experienced the same revolution from handcraft features or machine learning-based feature engineering as deep learning-based approaches. Layer-by-layer feature learning in deep networks is more likely to learn informative features that includes both low-level and high-level ones. Deep recurrent neural networks have demonstrated their capability to model temporal patterns and thus were investigated to estimate the RUL of mechanical systems or components in [188, 189]. As a significant branch of RNN, the LSTM-based approach is also explored for RUL prediction in [190], and a variation of LSTM, vanilla LSTM is researched in [191]. In [192], an end-to-end deep framework was proposed for RUL estimation based on convolutional and long-short-term memory (LSTM) recurrent units. The work in [193] designed convolutional bi-directional long short-term memory networks which utilize CNN to extract local features and introduce bi-directional LSTM to encode temporal information, capture long-term dependencies and model sequential data. Finally, stacked, fully connected layers and the linear regression layer are built on top of bi-directional LSTMs to predict the target value. The work in [194] proposes local feature-based gated recurrent unit networks to learn the representation from the sequence of local features. The supervised learning layer is added on the top to map the learned representation to the targets. Shao et al. [195] constructed a deep gated recurrent unit network based on the features extracted by the wavelet packet energy moment entropy to the fault prognosis of bearings.

More research has been conducted using convolutional neural networks [196], autoencoders [197, 198], particle filtering [199, 183] etc. In the work in [183, 200], neuro-fuzzy and particle filtering are incorporated into the fault diagnosis framework. The work in [201] integrates a deep belief network and a particle filter for the RUL prediction of hybrid ceramic bearings. The transferability of deep neural networks for fault prognosis is also investigated in [202, 203, 204, 205], such that the prediction of new objects without supervised training can be achieved.

2.2.3 Discussion

As shown in our literature review, diverse approaches based on a variety of machine learning algorithms have been fully explored and developed for predictive maintenance applications. However, there are still some limitations in the previous works which can be summarized as follows.

- They mainly monitor one piece of equipment and use this to undertake fault detection, diagnosis or prognosis. Traditional research generally depends on expert experience to detect and extract failures from relatively small-sized data, especially for certain mission-critical components which might involve safety or large financial costs, such as bearings, pumps and motors. In real applications, there are many pieces of equipment which engineers need to check regularly.

  It is expected that fault diagnosis will be conducted on a wide array of equipment across the whole factory for the digital twin purpose, instead of being limited to human-specified critical components. We need to monitor the whole manufacturing plant and transfer the information to where the expert engineers can monitor it. The data is collected in one place with the help of IoT as a tool and the data comes in different types and the amount of data becomes very large. Hence, we need to develop other techniques to deal with these issues.
- Most research is carried out in a laboratory setting using simulations or measurements on equipment in the laboratory rather than the field data collected directly from a real manufacturing plant. Traditional research has focused more on deep learning algorithms for predictive maintenance in recent years, and the data used for testing algorithm effectiveness are mostly from public datasets. Only a few papers utilize datasets collected from the actual operating equipment [102]. In practice, although performance in terms of accuracy is a key issue, smart predictive maintenance also needs a comprehensive analysis from the perspectives of implementation in real industry and feasibility for field data.

2.3 Plant-wide Predictive Maintenance

Today, in the Industry 4.0 era, the emergence of the concept of the IoT adds more value to the predictive maintenance process. Plant-wide predictive maintenance now refers to smart predictive maintenance for manufacturing in Industry 4.0. Smart manufacturing in Industry 4.0 refers to a new IoT-enabled manufacturing paradigm where manufacturing machines are fully connected through wireless networks, monitored by sensors, and controlled by advanced computational intelligence [206]. IoT enables manufacturers to collect data from all pieces of equipment and transmit it to one single place which engineers can monitor. This makes traditional predictive maintenance smart and evolved into a new stage called smart predictive maintenance. To achieve the industrial-scale deployment of predictive maintenance, IoT needs to be integrated with data science and analytics capabilities to provide support in the decision-making process involved in maintenance, reaching the ultimate objective of digitalization. Computational intelligence is an essential part of smart predictive maintenance to enable accurate insights for better decision making. So, although predictive maintenance has evolved in smart manufacturing, smart predictive maintenance also highly depends on the results generated by data modelling and analysis to take appropriate maintenance actions and ensure the healthy operation of the machines. It is noted that smart predictive maintenance is more than the fault detection, diagnosis and prognostic-related machine learning methods on which traditional predictive maintenance focuses. It requires a system integrated with many related technologies to achieve efficient data ingestion and management.

Smart predictive maintenance is greatly facilitated and further enabled and transformed by the use of IoT and the incorporation of CPS, machine learning, big data, cloud-based infrastructure and wireless communication technologies, etc. The integration of such enabling factors and technologies form a system and set the base for the development of predictive maintenance. In a smart predictive maintenance system, IoT architecture normally acts as the backbone where other disruptive technologies are embedded into proper positions to perform their own functionalities, including data acquisition/ingestion, data management, data storage, data security, and data analytic.

In recent years, many more efforts have been undertaken to address the challenges brought about by the evolution of Industry 4.0. After 2015, there was a growth in the number of articles [58, 207, 208, 209, 210, 211, 212, 213, 214] focusing on Industry 4.0, smart manufacturing, CPS and IoT which discussed the challenges, opportunities, enabling technologies, and intelligent applications where predictive maintenance is briefly mentioned. There are also several articles [215, 87] that detail predictive maintenance, but they are limited to theoretical analysis and discussion. Although several reviews [216, 102] which focus on predictive maintenance applications have recently been published, they only discuss machine learning applications and data-driven methods in predictive maintenance. Strictly speaking, this is only a part of smart predictive maintenance when it is actually deployed in an Industry 4.0 system in real manufacturing plants, and a system is required instead. With regard to smart predictive maintenance approaches, fewer articles can be found.

The work in [217] designs and implements a large-scale ICPS for monitoring industrial machines in a real-work environment, equipped with three blocks and several characteristics. The architecture includes local data acquisition, a cloud platform, and a front-end integrated with cloud-based distributed file systems for ubiquitous access to data, open-source programming frameworks to process and analyse big data, real-time data collection from cyber-physical devices and storage in the cloud, and an intelligent search engine to answer queries. To achieve these characteristics, Apache Flume, Kafka, Spark Streaming, Mesos and Zookeeper are the supporting technologies. Wan et al. [71] propose a cloud-based system architecture to collect manufacturing big data and explore data processing in the cloud for active preventive maintenance. The work in [71] is the initial work on predictive maintenance in Industry 4.0, proving the possibilities of real-time maintenance in the context of Industry 4.0 through an ecosystem design. However, as a simple architecture or ecosystem, it does not consider many other issues such as data management, data security and so on. The research in [218] provides an IoT embedded cloud control architecture with the suggested framework, where Intel's IoT analytics [219] is used as a cloud module and the deployed IoT device is connected to the cloud server through the related hubs and gateways with certain selected communication protocols. In the work in [220], a cloud and IoT-based mobile health care application is developed for monitoring, predicting and diagnosing disease. The proposed system consists of three phases, the first phase to collect data from the IoT devices, the second phase to store the medical records securely on cloud, and the third phase to predict the severity of the disease.

To better understand the transition and evolution of fault detection-related machine learning methods in smart predictive maintenance, this section also provides reviews which specifically focus on this.

2.3.1 Fault Detection, Diagnosis, and Prognosis-related Machine Learning

After reviewing the papers on smart predictive maintenance, we found that they always proposed a system. No matter whether fault detection, fault diagnosis or fault prognosis is incorporated into the predictive maintenance in the industrial IoT context, they only work in the big data analytics layer inside a system. We only focus on the data analytic part to better understand the transition of machine learning approaches to predictive maintenance applications. It is worth noting that, with the proliferation of IoT manufacturing big data, of the three tasks fault detection, diagnosis and prognosis, fault detection and diagnosis (also known as anomaly detection, and so on) are the main stream for the new forms of predictive maintenance which is based on IoT.

In light of the previous reviews on the fault analysis methods for traditional predictive maintenance, deep learning has shown superior performance in fault detection, diagnosis and prognosis. The work in [221] developed a general framework for predictive maintenance based on deep learning however it does not discuss how the framework can be applied in practice.

The work in [217] presents a large-scale platform for CPS real-time monitoring based on big data technologies. The platform is validated using press machines, but specific details on how to proceed with the data analytics are insufficient due to confidentiality. In [71], a manufacturing big data solution for fault diagnosis is proposed, including big data collection and big data processing based on a cloud-assisted architecture. The active preventive maintenance embedded in the big data processing step comprises two components: real-time monitoring to guide the schedules of maintaining resources and offline prediction to estimate the remaining effective working time. For real-time monitoring, a simple threshold-based algorithm is employed to guarantee timeliness. In the offline process, no such timeliness is required, thus a neural network is utilized for high-performance prediction. The work in [64] proposes an architecture for the implementation of IoT to collect, store and process sensor data from wearable sensor devices for the continuous monitoring of patients' health condition, helping doctors with early diagnosis and taking appropriate actions. The proposed monitoring system uses MapReduce-based stochastic gradient descent with logistic regression to develop the prediction model to identify whether the health condition of a patient is sensitive, critical or normal. The work in [218] provides an effective CPS architecture which is equipped with IoT deployment, the cloud platform and the DBN model to detect defective modules for vehicles' high intensity discharge (HID) headlight and cable modules. But the work in [218] did not specify the data type on which it is working and when dealing with the data coming in seconds or million seconds, its capability to provide real-time services cannot be proved.

The aforementioned reviews indicate that although an increasing number of fault detection, diagnosis, and prognosis algorithms for traditional predictive maintenance are continually proposed, simple and efficient methods generally tend to be considered for smart predictive maintenance instead of complex deep learning-based approaches. Unlike traditional predictive maintenance which concentrates on improving accuracy, the feasibility of deployment is given the priority for smart predictive maintenance.

2.4 Summary and Discussion of the Research Gaps

In this chapter, we conducted a comprehensive review on how maintenance evolves in manufacturing from corrective maintenance to predictive maintenance and then from traditional predictive maintenance to smart predictive maintenance in Industry 4.0, and we surveyed the related works.

Traditional predictive maintenance mainly focuses on the data analytic algorithms for one specific piece of equipment and the experiments are mostly based on the data which is manually collected from a laboratory. To achieve seamless monitoring across the whole manufacturing plant in an integrated fashion, predictive maintenance is expected to collect and bring the data from all pieces of equipment into one place for the centralized monitoring and management by engineers. Early work here used hard wired networks to bring the data from different pieces of equipment to a single monitoring site often called a SCADA system. These initial approaches had considerable inflexibility requiring hardware and network modifications or additions and software changes if new pieces of equipment were introduced, equipment was replaced or upgraded or the structure of the process in the manufacturing plant was changed. To overcome these issues, IoT steps in and traditional predictive maintenance evolves into smart predictive maintenance. Industry 4.0, the fourth industrial revolution [222, 223, 224], aims at creating smart factories and boosts the utilization of smart predictive maintenance for manufacturing and industries. As a key enabler of smart predictive maintenance, the IoT enables physical actions from machines to be translated into digital signals through human-to-human, human-to-machine, and machine-to-machine smart connections. However, the development of smart predictive maintenance is still in its infancy as new requirements need to be met when considering employing them in IoT environments. A lot of research gaps need to be filled which are summarized as follows.

- The first problem is how we collect data from different equipment, bring them together and manage the data in an appropriate way. Sensor data are normally generated at high speed and in a large volume in different formats including structured and unstructured data. We need to work out a way to gather the data from all around the manufacturing plant in a systematic way rather than using small-scale laboratory-collected data in a laboratory setting. This is also the most essential since the data is the basis of all the rest of the processing.

- When collecting data in real industries, it is streaming data from all pieces of equipment at every location in a manufacturing plant. So, other than the simulation data from the human supervised laboratory, it is beyond the capabilities of engineers to check on the data collection process. Any failures inside the process result in errors, noise, and missing points. In addition, to fully monitor the whole manufacturing plant, a large number of sensors are densely installed, leading to redundant features. So, data quality issues such as noise, missing values, and redundant features and so on should be given more attention and data preprocessing is an indispensable step in providing high-quality data for further analytics.
- With the availability of sensor data, data-driven methods, especially those that are based on deep learning for predictive maintenance, have gained wide attention and achieved great success. But these methods are designed for one specific piece of equipment and thus require huge transitions before they can be used for smart predictive maintenance which is envisioned to be applied for all the equipment in the manufacturing plant. Its application of machine learning or deep learning algorithms to smart predictive maintenance is limited due to the unique characteristics of high volume, fast velocity, different formats from various sources, low quality, partial labelling and so on, whereas most existing algorithms suffer from issues relating to a high computational cost, a long running time and they rely on labelled data.
- Once we collect the sensor data from all the equipment, big data comes with the data collection. Most machine learning techniques and open source packages are not feasible in terms of scalability, ease of use, extensibility and generalization ability when faced with the issues of big data. In addition, traditional processing methods for large-scale data often rely on local high-performance computers and parallel operations to improve computation power. New types of distributed processing systems are required for sensor big data analytics.
- For smart predictive maintenance, real-time monitoring is the key requirement in order to report the health condition of machines without delays. In addition to timeliness, smart predictive maintenance also has stricter requirements in terms of accuracy, efficiency, scalability and interpretability. All these requirements not only apply to fault detection, diagnosis, or prognosis-related data analytics in predictive maintenance, they also have to be met during the period of collecting and managing the data.

This thesis seeks to address some of these gaps that currently exist in the research and implementation. It begins with the problem definition and specification of the scope of this thesis in the next chapter.

Chapter 3
Problem Definition

In the previous chapter, a comprehensive literature review related to building a predictive maintenance ecosystem for manufacturing is presented and the current research gaps are also discussed. In this chapter, the problems and research issues addressed in this thesis are defined in a clear and concise manner. We start by defining the terms and concepts used in the context of IoT-based big data ecosystem for predictive maintenance. Next, we give a precise definition of the problem addressed in this thesis. Then, we discuss the methodology applied to solving the problem as stated in the problem definition.

3.1 Terms and Concepts Used

The terms used in this thesis are defined as follows:

Maintenance is the work that keeps the mechanical assets running with minimal downtime, which includes the functional checks, servicing, repairing or replacing devices and equipment as necessary. Maintenance can be done either in advance of failure or after failure occurs.

Predictive maintenance is maintenance that monitors the performance and health condition of equipment during normal operation using sensor data for the early detection and maintenance of possible faults to reduce the likelihood of failure.

Industrial Internet of Things is specifically applied in industrial manufacturing more often in this research project. To be more concrete, this is the concept of connecting manufacturing equipment across the factory to the Internet through the use of sensors. IIoT is composed of a network of sensors and it is able to collect manufacturing data that can represent the health condition of the equipment and these data are transferred over a wireless network to exchange data, optimize processes, and monitor devices.

Cyber-physical system integrates sensing, computation, control and networking into physical objects and processes. In a cyber-physical system, computational and physical resources are tightly interconnected, creating a smart control loop capable of adaptation, autonomy, and improved efficiency.

Data ingestion refers to the collection of metadata and numerical sensor data from all the pieces of equipment in real time in a manufacturing plant.

Data management refers to the effective storage, organisation and maintenance of the collected metadata and sensor data using a suitable data strategy and reliable methods. The well-prepared data can be used for further data analytics.

Data preprocessing in this research project involves transforming raw data into a proper format, standardization, and solving data quality issues, which requires data cleansing by removing noise, data imputation by filling missing values and feature selection by selecting suitable sensors before applying machine learning techniques.

Fault detection is the detection of a fault in a piece of equipment which will prevent the equipment from working according to its specifications.

Anomaly detection involves the identification of unknown, abnormal patterns or events called anomalies in the data. In our case, this indicates the performance of all the equipment instead of only the critical components. A significant anomaly detected in the data from a manufacturing plant is considered to be an indicator of the likelihood of the occurrence of a fault. By monitoring a system or equipment in manufacturing, an alarm will be triggered and a notification will be delivered to engineers once a significant anomaly has been detected.

Fault diagnosis is normally responsible for pinpointing the type of detected fault and the root cause of an issue. In this thesis, fault diagnosis is another term for "variable contribution analysis", which refers to the analysis of which component (reading sensor) causes the failure, if there is an anomaly.

Industry 4.0 (The fourth industrial revolution) is the ongoing automation of traditional manufacturing and industrial practices, using the combination of a cyber-physical system, the IoT, and other modern smart technologies. In manufacturing, these are integrated to increase automation and improve communication and the intelligent monitoring of equipment. Industry 4.0 enables equipment issues to be analyzed and diagnosed without the need for human intervention.

Big data analytics refers to the collected IoT manufacturing data that is characterized by 5Vs as described in Chapter 1. Big data analytics is leveraged in IoT predictive maintenance applications to store the manufacturing data collected from the connected equipment in the big data system, interpret the IoT big data using advanced analytic tools like Hadoop, Spark, etc., and generate the descriptions for accurate and timely decision-making.

Cloud refers to the cloud resources which are used to store and manage the collected sensor data on the manufacturing equipment in a central cloud cluster. It also refers to the technique which distributes data analytics tasks into multiple servers or nodes to achieve parallel computation, namely cloud computing.

Edge computing is computing that takes place at or near the source of data, which is the place where the monitoring equipment is located, and it might be in a department or factory. On one hand, moving a part of the processing tasks from the remote cloud to near the equipment can greatly relieve the burden on the network bandwidth and reduce network delay. On the other hand, it is an effective way to address privacy and security issues.

Fog computing, originated by Cisco, extends the cloud to where things are utilizing edge devices to carry out a substantial amount of computation, storage, and communication locally close to the source of data. Along with it, cloudlets and mobile edge computing together are known as three main types of edge computing models, which have been widely developed and deployed.

NoSQL database is the database that stores data in a format other than relational tables. The related data can be nested within a single data structure. The NoSQL database is used to store huge amounts of the unstructured sensor data of all the equipment and fast-query a billion records within a few seconds in this thesis.

3.2 Problem Definition

The problem tackled in this thesis is to define a method for fault prediction and diagnosis in a manufacturing plant through a complete ecosystem for collecting data directly from all the different elements, providing a seamless transition of the information through various stages and to utilize this information to carry out predictive maintenance for all the equipment in the manufacturing plant. This will entail the conceptual definition and implementation of the conceptual model and validation of the approach utilizing actual data collected from a real practical manufacturing plant. The process of building such a complete ecosystem requires collecting the data, ingesting the data, data cleansing, feature selection, data management, data analytics for fault prediction and diagnosis, and displaying the information. Rather than performing traditional small-scaled data analysis on a local computer, an IIoT-based architectural framework which is deployed on different platforms and provides seamless integration of the various stages is needed. It should be able to give a complete picture of the way the data should be handled or processed through the whole data flow chain from sensors to large data analytics.

To be more specific, this research project will concentrate on addressing the following problems:

1) **collect and manage the sensor data from all equipment;**

   In a real industry setting, the connected sensor data is gathered in an unstructured format and is high in volume, large in size and dimensions, with a large amount of noise and constantly changing patterns due to changes in the environment and sensor failure and replacement. As a large amount of data arrives every second, data acquisition is the first issue to address, involving ingesting data from various sensors, transmitting the raw source data into a storage facility, and transforming the unstructured data into a well-structured format that can be processed. To deal with the large volume of streaming data, data management is another issue to be resolved to ensure ease of queries, efficient storage and access control-related security.

2) **preprocess the collected sensor data;**

   Smart predictive maintenance requires a long process to collect and manage the sensor data through a system. Before the data arrives at the data analytic layer, it is first sensed by the sensory devices, stored in a proper format after a series of reading and writing operations, then it is transmitted through the wireless network, and might go through encryption and decryption if necessary. It is very likely that it is subject to errors and noise. Especially in industry, a large number of sensors push it beyond the capability of engineers to make sure all the equipment is working effectively. Specifically,

   - The raw data from IoT devices suffer from quality issues due to hardware imperfections or unreliable wireless transmissions, possibly leading to inaccurate analytics. For instance, *Fig. 3-1(a), (b), (c)* show compressor sensor data with selected features from global chemicals, fertilisers and explosives manufacturing. It can be clearly seen that the collected data from various timestamps have different range values, since sensors may read data incorrectly or there may be transmission errors.

- The data from massive multi-source and densely distributed sensors are likely to be redundant and even contradictory.

In this case, data quality issues are going to be resolved by proper data preprocessing steps.

3) **analyze the streaming sensor data;**

On completion of the data collection, transmission, and preprocessing, the sensor data is analyzed for the early detection of possible faults and the results are delivered to the engineers. For the data analytics, performance in terms of accuracy and time cost both need to be considered. This sensor data collected from all the equipment in a manufacturing plant not only inherits the characteristics of big data, it also has other ones such as partial labelling. In real industrial settings, the generated high frequency data flows in a streaming fashion, which is impractical for fine-grained manual labelling. So, most approaches used in traditional predictive maintenance are restricted to these sensor data. The first problem is to develop a method which is suitable for such sensor data. The second problem is introduced in terms of computation complexity which concerns the time cost. How to analyze the sensor data which consists of the information from all the equipment also needs to be resolved in both an effective and efficient way.

4) **seamlessly integrate all elements into one system;**

The defined problem requires a process to solve the issue and this process needs to undertake data ingestion, management, preprocessing, and data analytics for real-time monitoring. Generally, most works have addressed one of the subproblems without understanding the interactions with the other subproblems. Instead of a simple concatenation, the integration of heterogeneous layers, methods and tools has to be addressed in an appropriate way. In this sense, all the elements need to be compatible with each other and work properly across the whole system.

5) **ensure the system works in a real-time environment.**

The whole system needs to be designed to be capable of providing real-time monitoring services. In terms of real-time monitoring, it relies on both the processing time of the system and the rate of incoming data. As long as the system is able to finish the processing of current data points including collection, transmission, and analytics before the next new data point arrives, this data analytic system is capable of real-time analysis.

These defined problems are solved in Chapter 4, Chapter 5 and Chapter 7.

utc_timestamp	45xi25500x_s_pv	45xi25500y_s_pv	45xi25501x_s_pv	45xi25501y_s_pv	45xi25502x_s_pv	45xi25502y_s_pv
2015-01-16 01:00:01.0	0.209790229797363	0.2797691822052	0.209716901183128	0.0699790045619011	0.0699300765991211	0.139811262488365
2015-01-16 01:00:02.0	0.209790229797363	0.2797691822052	0.209716901183128	0.0699790045619011	0.0699300765991211	0.139811262488365
2015-01-16 01:00:03.0	0.209790229797363	0.2797691822052	0.209716901183128	0.0699790045619011	0.0699300765991211	0.139811262488365
2015-01-16 01:00:04.0	0.349650382995605	0.2797691822052	0.27962252497673	0.0699790045619011	0.139860153198242	0.139811262488365
2015-01-16	0.349650382995605	0.2797691822052	0.27962252497673	0.0699790045619011	0.139860153198242	0.139811262488365

(a) 5 observations with few features on date 2015-01-16

utc_timestamp	45xi25500x_s_pv	45xi25500y_s_pv	45xi25501x_s_pv	45xi25501y_s_pv	45xi25502x_s_pv	45xi25502y_s_pv
2015-01-20 01:00:03.0	10.0000009536743	10.6312294006348	12.5830135345459	11.5465354919434	18.1818199157715	17.1967849731445
2015-01-20 01:00:04.0	10.0000009536743	10.6312294006348	12.5830135345459	11.5465354919434	18.1818199157715	17.1967849731445
2015-01-20 01:00:05.0	10.0000009536743	10.6312294006348	12.5830135345459	11.5465354919434	18.1818199157715	17.1967849731445
2015-01-20 01:00:06.0	10.1048965454102	10.6662006378174	12.5480613708496	11.5465354919434	18.1818199157715	17.1967849731445
2015-01-20	10.2097911834717	10.7011709213257	12.513108253479	11.5465354919434	18.1818199157715	17.1967849731445

(b) 5 observations with few features on date 2015-01-20

utc_timestamp	45xi25500x_s_pv	45xi25500y_s_pv	45xi25501x_s_pv	45xi25501y_s_pv	45xi25502x_s_pv	45xi25502y_s_pv
2017-11-09 00:00:02.0	9.9300708770752	10.0716905593872	13.2121639251709	-0.503651976585388	18.7675075531006	17.2727298736572
2017-11-09 00:00:03.0	10.0699310302734	10.4214019775391	13.2121639251709	13.016095161438	15.7342672348022	17.2727298736572
2017-11-09 00:00:04.0	84.2553176879883	13.2121639251709	80.8932342529297	-0.50588873427594	18.7675075531006	-0.284935355186462
2017-11-09 00:00:05.0	84.2516098022461	13.1888618469238	80.8947372436523	-0.505143046379089	18.8141937255859	-0.285494595766068
2017-11-09	84.2479095458984	13.1655607223511	80.896240234375	-0.504397511482239	18.8608779907227	-0.286053866147995

(c) 5 observations with few features on date 2017-11-09

Fig. 3-1: Illustration of compressor sensor data samples from global chemicals, fertilisers and explosives manufacturing

3.3 Research Method

There are two common research methodologies, namely social science research and science and engineering-based research, which can be adopted in this thesis. Social science research can either be quantitative or qualitative. It is often carried out through survey or interview processes. Quantitative research involves extensive data gathering usually using methods such as surveys and the statistical analysis of the gathered data in order to prove or disprove various hypotheses that have been formulated. Qualitative research frequently involves in-depth structured or semi-structured interviews that allow one to pursue particular issues of interest that may arise during the interview. It does not normally involve a large amount of data and the gathered information may not be in a form that easily allows statistical analysis. This kind of research can indicate the extent to which the methodology is or is not accepted and sometimes might be able to provide the reason for this. However, unlike engineering-based research, this type of research does not explain what a methodology should be and how to produce a new methodology for problem solving. This research only tests or evaluates a method that has already been produced from science and engineering research.

On the other hand, science and engineering-based research is concerned with verifying theoretical predictions or forecasts. In the engineering field, the spirit of 'making something work' is vital and has three levels: the conceptual level, perceptual level and the practical level, which are explained as follows [225]:

- Conceptual level (level one): the creation of new ideas and concepts through analysis.
- Perceptual level (level two): formulating a new method and approach by designing and implementing the tools or system or environment.
- Practical level (level three): testing and validating the new method and approach via experimentation using real-world examples.

Science and engineering research can lead to new or improved techniques, architectures, methodologies, or a set of new concepts which together form a new theoretical framework. Most often, it not only addresses the issues at hand, but also proposes solutions. The objective of this research is to design and implement a novel architectural framework for predictive maintenance which is capable of implementation in a real smart manufacturing plant. To achieve the objective, an integrated framework will be developed by utilizing various technologies and techniques. Therefore, this research clearly falls into the domain of system building as an Information Systems Research Methodology [226], a science and engineering research approach.

3.4 Conclusion

This chapter defines the main terms which we use throughout the thesis. We also define the problem being addressed in thesis and the research methodology, namely the science engineering research methodology we utilize in this thesis. In the next chapter, we demonstrate an overview of the solution which is proposed to solve the defined problem.

Chapter 4

Overview of Solution

With the advancement of smart sensor technologies, more physical metrics can be measured. To resolve the problem of monitoring the whole manufacturing plant in a centralized way, the sensor data from all the equipment become available with the utilization of modern smart sensors and can then be transmitted using the IoT as a tool through the wireless sensor networks into one center place which engineers can remotely monitor. IoT-based industrial big data bring new perspectives for richer information, deeper insight, and smarter decision making. Recently, many global manufacturing industries have started to build a data fabric to develop smart predictive maintenance in the manufacturing plant. They start at Phase-1, as depicted in *Fig. 4-1*, by ingesting and providing insights into the data from different pieces of equipment through the utilization of sensor data from a relatively large number of sensors. Of these industries, Incitec Pivot Limited (IPL), which is our partner company in Australia, is one of the pioneers and has invested in a project especially to research smart predictive maintenance based on IoT. Based on this project in a real industry, to address the problems discussed in Chapter 3, we propose an IoT-based big data ecosystem for smart predictive maintenance and implement an integrated framework in an intelligent manufacturing system.

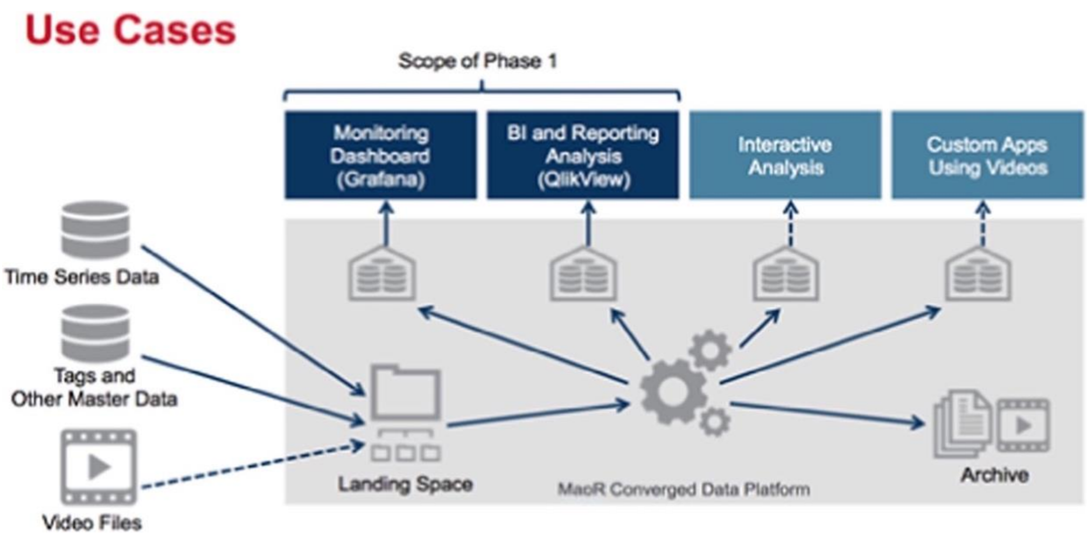


Fig. 4-1: Use cases of data ingestion from different manufacturing plants

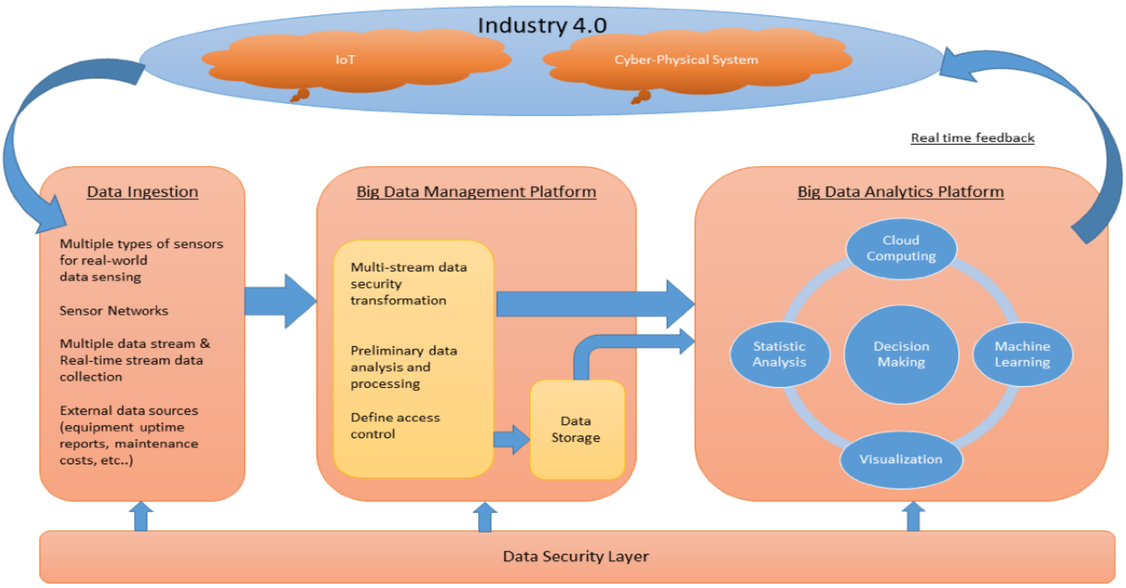


Fig. 4-2: The proposed architecture for IoT based big data ecosystem in industry 4.0

Our approach splits big data issues into various layers, namely the big data ingestion layer, the big data management layer, the data pre-processing layer, the data analytics layer, and the data visualization layer, to achieve a processing system where each layer performs a specific function. *Fig. 4-2* shows the practical implementation of the architecture of a manufacturing big data ecosystem. We provide an integrated framework including big data ingestion and a management phase supported by an optimized IoT architecture driven by advanced applications to address data acquisition, data storage and data security needs; a data preparation phase for data quality issues; predictive modelling for analysing the equipment health state; and alarm sequence analysis to generate a comprehensive report for end-users. The proposed ecosystem is integrated, practical, and effective in terms of both working on big data and in a real-time environment, achieving the following effectively and efficiently:

1) It ingests structured and unstructured data from various data sources to a central cluster in a real-time environment.
2) It implements a data lake structure for effective big data management with the support of the Optimized Hadoop Distributed File System (HDFS), Apache Hive, NoSQL, etc.
3) It addresses data security issues using access control expressions (ACEs), combined protocols, and encryption techniques.
4) It performs real-time data processing and predictive modeling on the Apache Spark Platform.

5) It monitors streaming sensor data and analysis results through the connection between databases, the back-end API server, and the dashboard for the presentation of information to maintenance engineers.

The proposed ecosystem not only offers a complete solution which includes every step that might be needed for manufacturing equipment health state monitoring in predictive maintenance in industrial IoT-based smart manufacturing, it also resolves data quality issues by incorporating a data preparation phase, which is continuously upgraded during the process of our research. Initially, data preprocessing operations such as noise detection, feature selection and so on, are performed by data scientists manually based on their expert knowledge. However, we propose an upgraded ecosystem equipped with an automatic noise detection and feature selection algorithms. To address the partial labelling issues as discussed in the problem definition, the proposed framework is incorporated with unsupervised learning techniques with the ability to be executed as an autonomous agent and perform in a streaming fashion. Thus, it has high portability which implies its applicability to various types of mechanical equipment, without the need for human intervention and supervision.

The integrated framework aims for practicality in real production and the business community. It can be easily used in Industry 4.0 systems, taking advantage of the simplicity of distributed Principal Component Analysis (PCA) and its deployment feasibility in Spark using the cloud and distributed computing. As the external factors (temperature, weather, pressure, humidity) and internal environment (degradation) of industrial settings continually change over time, the sensor measurement values also change even under controlled conditions. The predictive model is equipped with a retraining phase when the environment changes, and the retraining mechanism leads to an even heavier workload, in which case the distributed PCA offers more advantages than most of the other approaches. Our experiments using more than six months data show that it takes less than five minutes for the framework to complete the whole process, whereas it takes more than eight hours for gradient tree boosting using a Spark distributed engine in the same environment. Specifically, the proposed framework makes the following contributions.

- The proposed framework is integrated, covering all the elements of the equipment health state monitoring procedure in predictive maintenance, including data ingestion, data management, sensor selection, noisy data cleaning, fault detection, contribution analysis and alarm sequence modelling, hence arriving at a comprehensive health condition report. Each element is equipped with effective and efficient techniques. This is the first work that includes all the necessary steps and details to construct an integrated manufacturing intelligent system.

- The proposed framework is generic, so it is not limited to task-specified equipment or components, rather, it can be applied to a wide array of machines or systems. There is no need for human-assisted signal processing for feature engineering. Instead, a high-noise feature elimination technique is designed in the framework to perform automatic unsupervised sensor selection without human intervention.
- The proposed framework focuses on data quality issues and in particular, includes a data preparation phase to distil the raw dirty data. We introduce the new challenges of sensor selection and noise detection problems for IoT big data and provide novel solutions.
- The proposed framework is practical and applicable to Industry 4.0 in two aspects: real-time processing of industrial big data and ensuring its integrity for diagnostics from data acquisition and management to fault identification, which is applicable to real production. The existing methods do not work well when applied to big data. The data analytics layer in our framework avoids the need for a complex model structure and resource-hungry algorithms, hence taking advantage of distributed computation on the Spark platform. In addition, we focus equally on data acquisition, for example efficient data ingestion, storage and query, and data preprocessing, for example data quality assessment and noisy data cancellation which is essential to provide structured and healthy data in an industrial production system.
- The proposed framework has been successfully implemented on a four-stage syngas compressor in a real industrial environment and won the Best Industry Application of IoT at the BigInsights Data & AI Innovation Awards [227]. A case study involving more than a million rows of data collected over a six-month period is described and analysed in detail. Extensive experiments are conducted to demonstrate the effectiveness of the framework and the necessity of the data preparation phase in relation to data quality issues. Through the case study, we elaborate on the guidelines as to how to generate a comprehensive analysis report.

Recapitulation

In this chapter, we gave an overview of the solution. In particular, we defined the elements of the IoT-based big data ecosystem to be employed. We concluded with a discussion of the contributions made. In Chapter 5 and 6 we provide details of this solution and verification of this.

Chapter 5

An IoT-based Global Manufacturing Big Data Ecosystem for Predictive Maintenance

The previous chapter gave an overview of the solutions utilized to solve the research problems defined in Chapter 3. We noted that in the discussion of the development of the architectural framework and ecosystem in Chapter 4 that we utilize IoT, big data and cloud computing. Therefore, it is useful in this chapter to start by reviewing the relevant previous research on IoT, big data and cloud computing to provide a context for the discussion of the work detailed in the rest of this chapter. Then we elaborate on our proposed framework in terms of data ingestion, data management, data preprocessing, predictive modelling and data visualization.

5.1 Review of Relevant Research on IoT, Big Data and Cloud Computing

The use of intelligent IoT sensors enables Industry 4.0 systems in smart manufacturing and poses new requirements and challenges for predictive maintenance. *Fig. 5-1* summarizes the new elements accompanying industrial IoT-based predictive maintenance. IoT, cloud computing, big data analytics and advanced analytical techniques are the key technologies to develop an IoT-based architectural framework. Smart IoT is responsible for continuously generating real-time sensor data from various sources and the collected data are growing in an explosive manner due to the recent proliferation of IoT techniques. As huge volumes of data are becoming available at a high velocity, it is beyond the capability of an analyst to undertake immediate processing simply by observing the data, and an efficient big data analytics system is highly desirable. Traditional fault detection and diagnosis techniques no longer meet the new requirements of smart manufacturing. Therefore, we propose an IoT-based big data ecosystem for predictive maintenance in this thesis. To better understand our proposed ecosystem, this section provides a brief review on the existing IoT architectures and the related work on cloud computing and big data analytics to provide the background to introduce our proposed ecosystem in detail.

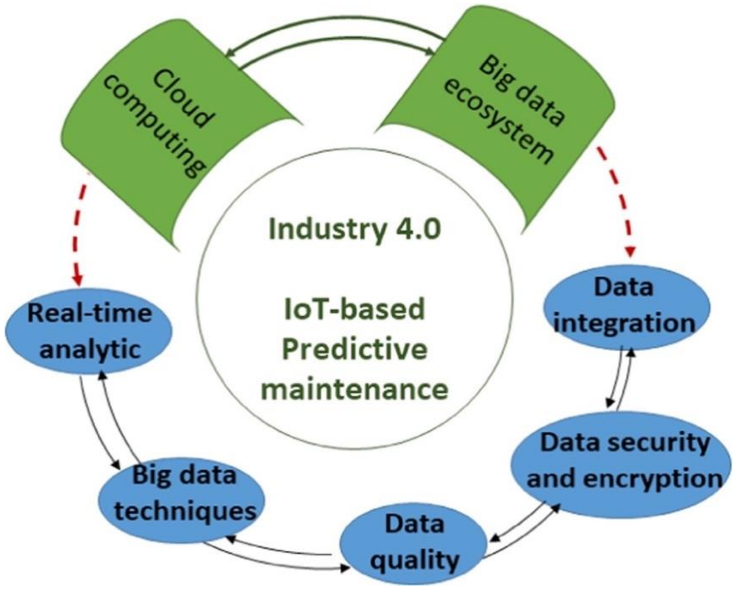


Fig. 5-1: New challenges for predictive maintenance in industrial IoT-based smart manufacturing

5.1.1 IoT Architecture

The IoT is dramatically evolving and creating various opportunities for smart manufacturing in Industry 4.0. Thus, many researchers have conducted studies on the IoT starting from the architecture design and implementation. It is not new research field and is actually quite mature, but its future architecture is still evolving and under construction. There also have been diverse surveys focusing on different aspects. Initially, a four-layer-based IoT architecture was proposed, consisting of a perceptual layer, aggregation layer, network layer, and application layer [228]. Hsieh et al. [229] proposed a new IoT architecture based on two communication networks: power line communication (PLC) and 3G networks, to integrate different layers of the IoT framework. Qin et al. [230] proposed a software defined networking architecture for IoT, aiming to achieve a high-level quality of services for different IoT tasks in the IoT multinetwork environment. Ning et al. [231] proposed two architectures which may be used in the future, namely Unit IoT and Ubiquitous IoT. Unit IoT is designed to focus on a special application, while Ubiquitous IoT is the integration of multiple Unit IoTs, realizing “everything connected, intelligently controlled, and anywhere covered”.

The architectural aspects of the IoT is an actively studied research area. While there is no agreed architecture for the IoT in the existing research, the IoT has gradually converged to a multi-layered approach, with each layer dedicated to certain functions [232]. A three-layer architecture was proposed in the early stage of the IoT evolution [233, 234, 235]. The three-layer architecture is the most basic with three layers, a perception layer for data collection, a network layer for data transmission, and an application layer for data analytics. The work in [236, 237, 232] proposed four-layer and five-layer architectures that contain one additional processing layer and one business layer compared to the typical three-layer architecture. The five-layer architecture proposed in [11] adds a middleware layer and business layer on top of the three-layer architecture. Lee et al. [232] presented a modified five-layer architecture for the enterprise IoT, where the processing layer is included and the service management layer replaces the business layer. With billions of objects connected to the IoT, many factors need to be considered when designing a new IoT architecture such as reliability, scalability, modularity, interoperability, interface, QoS, privacy, security, and so on. A common solution is to embed a generic IoT architecture with technologies, focusing on addressing several specific requirements [238]. Recently, the most commonly adopted architecture typically contains three layers, the sensing layer, network layer, and application layer [239].

5.1.2 Big Data Analytics

Under the context of IIoT in smart manufacturing, massive data are generated at an unprecedented level. The work in [240] reveals that it is the adoption of IoT in manufacturing that enables the transition of traditional manufacturing systems into modern ones, and also generates industrial big data. The paper illustrates how industrial big data is generated by the IoT paradigm through a case study. Big data analytics, in turn, enables IIoT systems to deliver value for the data collected from physical machine components to decision makers.

The work in [241] notes the need for an integrated IoT big data analytics framework and proposes one such framework for the storage and analysis of high-speed real-time data from a smart building. The framework is implemented on top of the Cloudera big data platform (virtual machine for the Apache Hadoop environment), utilizing the Transmission Control Protocol and Apache Flume technologies for big data management and PySpark scripts for real-time data analytics. The work in [242] proposes an IoT-based CPS for big data exchange and analysis, which consists of five layers, i.e. presentation layer, analytical layer, data layer, infrastructure layer, and IoT layer. The work in [243] proposes a new computing paradigm, Firework, which is designed for big data processing in a collaborative edge environment. The work in [244] proposes an IoT-based system using big data analytics for the application of a smart city. The complete system starts from data generation and collecting, aggregating and processing and finishes at decision making, where efficient big data processing is realized under the support of Spark over Hadoop. The work in [245] discusses an integrated framework based on the Internet of Underwater Things (IoUT) and big data, and the collected data can be saved and analyzed through a platform based on the Hadoop Multi-Node Cluster. The work in [246] presents a new platform with the use of fog nodes and the cloud system to enable innovative analytics on IoT big data from smart homes and addresses the challenges of resource demands for data processing, storage, and classification analysis.

A big data computing architecture for the smart grid is proposed in [247]. The architecture involves data resources, transmission, storage, and analysis. Some key technologies are emphasized to enable big-data-aware wireless communication, including software-defined network (SDN), cloud and cloudlet, crowdsourcing, and cache control. In the work in [248], an overall architecture of big data-based analytics for the product lifecycle was proposed. The architecture is composed of four layers, i.e. application services, big data acquisition and integration, big data processing and storage, big data mining and knowledge discovery in the database. Inside the architecture, the Storm real-time computing framework and Hadoop computing framework are used to process the real-time and non-real-time data respectively. The distributed database system (DDBS), the Hadoop distributed file system (HDFS) and the structure query language (SQL) data management system are used to store heterogeneous big data. There are also other important works [249, 250, 251, 252, 253, 254, 255, 256] on big data analytics applications in IoT. Generally, they deploy big data analytics applications on a well-developed big data processing platform. With the advent of IoT, researchers have started to take further advantage of cloud services or edge computing to expedite big data analytics processes. However, based on the reviews on these studies, we observe that when big data analytics is related to IoT, a big data analytics framework is proposed to deal with big data issues, where IoT, CPS, and cloud computing are integrated together. Such frameworks can also be called IoT architectures. Similar concepts include CPS/cloud-based/edge-based/cloud-edge-based architectures/systems/frameworks.

Although numerous studies have been conducted on IoT, IIoT [257, 258, 259, 260, 261], big data analytics [262, 263, 41, 264, 265], and IoT-based big data analytics [44, 246, 266, 267, 268] separately, only a few studies [269, 56] have explored the convergence of IIoT and big data analytics. The work in [269] discusses the lack of implementation of big data analytics in IIoT systems in the existing literature. As described in [56], the processes of big data analytics are executed as a result of multistage highly interdependent application components. Instead of only focusing on data analytics, big data analytics is involved in the whole lifecycle of big data which includes data engineering, data preparation, data analytics, and data management. This corresponds with our discussion that in IIoT systems, a complete big data analytics framework is desirable to cope with the big data issues that occur in multiple stages from data collection to data analysis.

5.1.3 Cloud Computing

In light of the above reviews, the cloud is regarded as a promising tool to support big data analytics in terms of data storage and analytics. Cloud computing is the key technique and has traditionally served as a reliable and cost-effective means for the management, storage and processing of data. The integration of IoT devices and CPS leads to the generation of big data, which requires huge resources for the execution of big data analytics processes. The most common method is to leverage the capabilities of cloud infrastructures and services [270, 271] for the processing of big data and streaming data.

Jiang et al. [68] proposed a data storage framework to overcome the challenges brought about by the IoT data which is generated rapidly in a huge volume and is of various types. The framework enables the efficient storage and integration of both structured and unstructured data by combining multiple databases and Hadoop to manage diverse types of data collected from different sensors. The work in [272] overviews potential IoT big data storage systems in cloud computing based on the processing process of the IoT application and discusses the challenges and opportunities. Resources including storage and computation that are provided by the cloud enable the management of massive data to be optimized centrally through the network [247].

A large number of studies [273, 274, 275, 276, 277, 278] discuss the opportunities and challenges of IoT and cloud computing integration. It is obvious that the integration of cloud computing and IoT has been significantly facilitated by the development of IoT systems applied in various fields. Doukas et al. [279] presented a platform based on cloud computing with IoT to efficiently manage the sensor data in pervasive healthcare applications. The IoT and cloud computing are adopted in enterprise systems in [70] to evolve assembly modelling systems into advanced ones capable of automatically dealing with complexity and changes. In the work in [69], the application of IoT and cloud computing in manufacturing are investigated to realize the full sharing, free circulation, on demand use, and optimal allocation of various manufacturing resources. Based on the studies, a cloud computing and IoT-based cloud manufacturing service system is proposed. Cloud computing and IoT technologies are integrated and used in [280] to build a multilayered platform for vehicular data to resolve the increasing transportation issues. The work in [281] conducts an in-depth analysis and discussion on the integration challenges to establish an intelligent transportation system that is expected to address the issues of high fuel prices, the high level of CO_2 emissions, traffic congestion and road safety. Cloud computing has been also utilized in conjunction with IoT for sustainable smart agriculture [282, 283], healthcare [284, 285], marine surveying and mapping [286], etc.

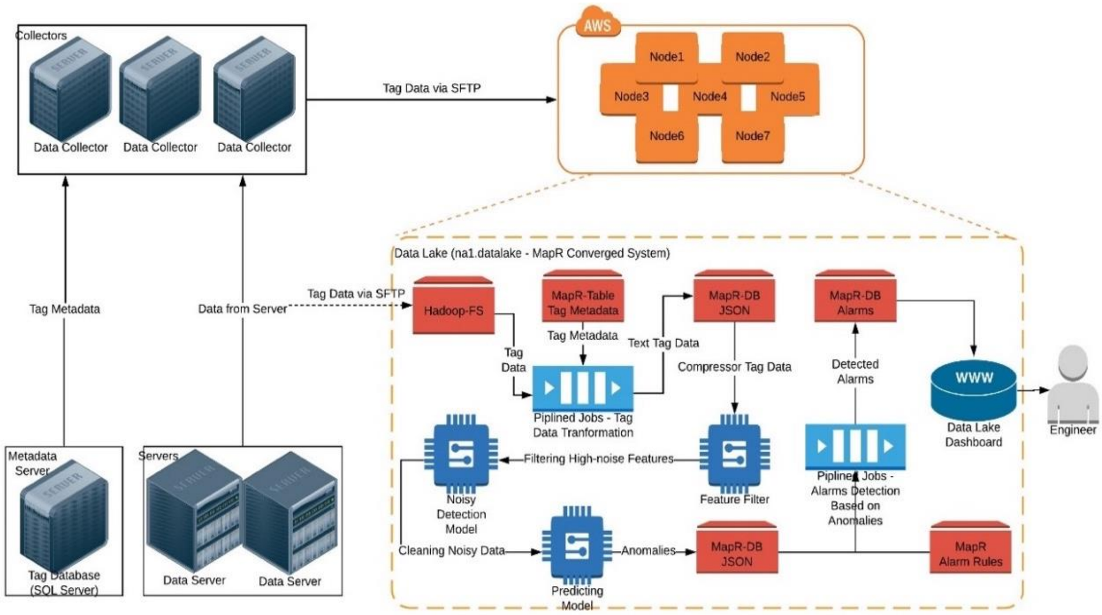


Fig. 5-2: The pipeline of the proposed practical and integrated framework

5.2 Proposed Architectural Framework

The originality of this work lies in its proposal of an integrated framework for manufacturing equipment health state monitoring and diagnostics using IoT big data in a real-time industrial setting, starting with how the multi-source data is collected, transformed and stored. The proposed framework consists of five phases: 1) data ingestion, 2) data management, 3) data preparation, 4) predictive modelling, and 5) visualization. It is practical for both IoT-based big data and real-time environments since the techniques adopted in these five phases are especially designed to give equal consideration to accuracy and efficiency. The framework can be effectively implemented through the big data Apache Spark platform, which enables it to be easily deployed in Industry 4.0 systems. The embedded pipeline is shown in *Fig. 5-2*. *Fig. 5-3* depicts in detail the cloud-assisted architecture and how the different layers are arranged and integrated. In what follows, the framework is elaborated phase by phase.

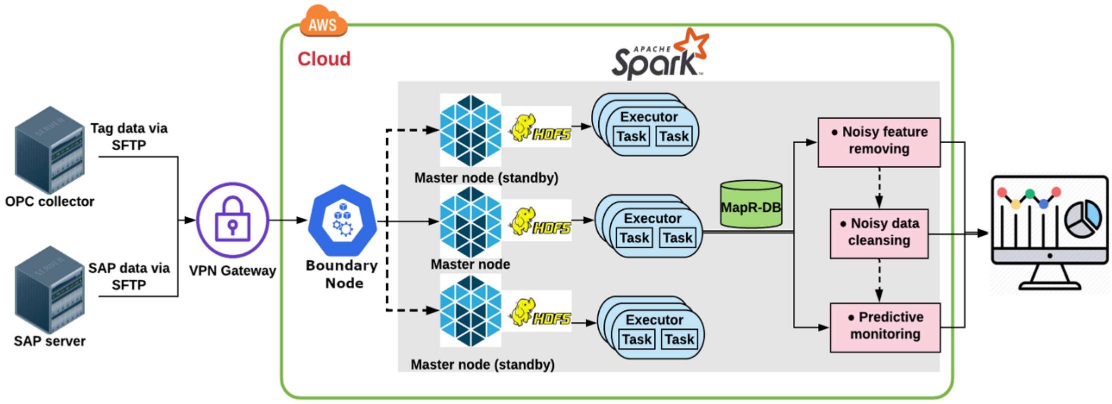


Fig. 5-3: The proposed cloud-assisted architecture

5.3 Big Data Ingestion Layer

IoT brings rich data that boost the popularity of data-driven methods for predictive maintenance, while data-driven methods make data acquisition in data ingestion layer an indispensable step for a complete ecosystem. As huge volumes of data are becoming available at a rapid rate in digital industries, an efficient data ingestion method is a prerequisite for further data monitoring, analytics and modelling. Data ingestion is responsible for extracting different types of data, especially unstructured data, from various data sources of different sizes, formats, speed and frequency into a system. As shown in *Fig. 5-4* for the data ingestion phase, our framework ingests the numerical sensor data of all the equipment from OPC Servers [287] by OPC Collectors, which is a platform-based Windows Service Application and is installed in the LAN where the OPC server is running. The measurement logs with structured data are stored in the metadata server and historical servers are also deployed to store historical sensor data in case they need to be extracted in the future. The ingested data arrives as text files every minute on the boundary node of the AWS cluster through the Secure File Transfer Protocol (SFTP), as the system applies a batch processing approach due to its suitability for data transformation in terms of processing efficiency. The boundary node is the interface between the Hadoop cluster and the outside network. It is used to run applications and cluster administration techniques in our system. SFTP is used to provide reliable connection-oriented data transfer over the Transmission Control Protocol (TCP) with the capability to resume in the face of a potentially unstable or slow network connection between the data sources and the cloud environment. Furthermore, a VPN gateway is set up between different OPC Collectors and boundary nodes to secure the entire network at the data ingestion phase. After the evaluation of the implementation, the proposed big data ingestion method in Algorithm 1 ensures the ability to provide high performance and stability.

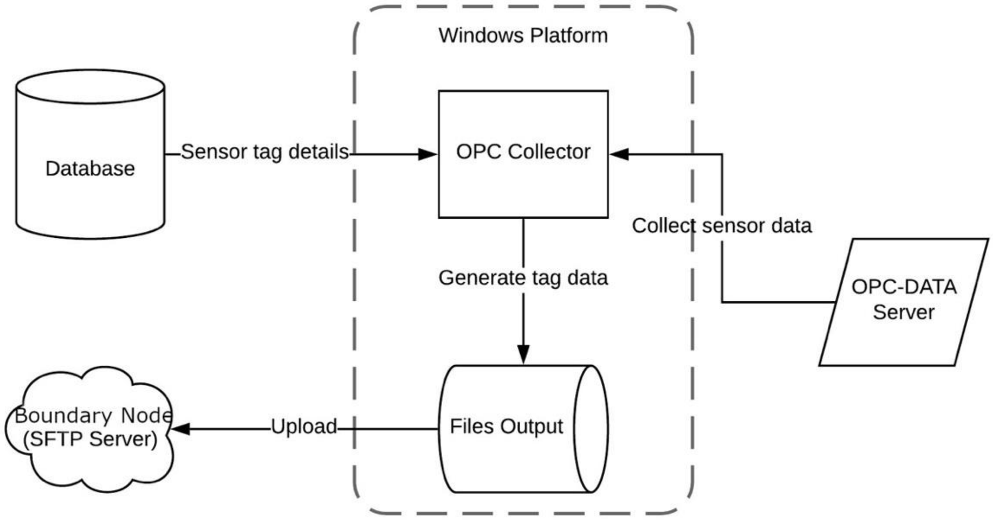


Fig. 5-4: Diagram of the big data ingestion procedure

Algorithm 1 Ingest the sensor data into the boundary node of the AWS cluster

---

Environment1: Network configuration, including set up VPN and port number to connect OPC servers of various manufacturing plants to clusters;

Environment2: Install virtual machine for OPC Collector.

Step1: Retrieve metadata of sensor tags that users want to collect;

Step2: Match and monitor the collected tags from OPC Server to generate new tag value data by: if collected sensor names in Metadata Server exist in OPC Servers;

Step3: Write the tag values data into a text file;

Step4: Compress and upload the text file into the boundary node through SFTP protocol every minute.

5.4 Big Data Management Layer

In our big data management phase, every piece of collected data from the boundary node is replicated three times, randomly distributed and stored on the cloud nodes using an optimized Hadoop Distributed File System (HDFS), which can preserve high volumes of large text files. For such huge data, the files are stored across multiple nodes in a redundant fashion to avoid possible data loss in case of system failure and provide the availability of parallel processing. This type of storage is called a data lake, which is a new type of cloud-based enterprise architecture that structures data in a more scalable way. Within the data lake, structured and unstructured data can be stored on any scale. It accepts all types of data delivered from the boundary node, stores the data in clusters, and retrieves and manages them in the warehouse and database. Once data is collected on the cluster, it is transformed into suitable formats for each use case to provide an optimized data structure by utilizing any of the processing engines, such as Apache Spark. In addition, all the data transmissions between cloud nodes in this secure cluster are encrypted to prevent any potential attackers accessing communication to obtain the contents of the transmission. Encryption restricts the ability of an external party to read or modify data. For instance, the Secure Sockets Layer and the Transport Layer Security protocol are deployed in our system to secure several channels of HTTP traffic for communication security. The cluster also applies Access Control Expressions (ACEs) which is a powerful model to control the access to data using Boolean logic expressions.

It is essential to choose an effective data storage facility and format for each use case to satisfy the application requirements including response time, throughput and the number of concurrent accesses, etc. Therefore, in the proposed framework, all the transformed, processed data and data analysis results are stored in the MapR database with different table formats: JSON and Binary. The MapR Database is an extremely scalable, reliable, globally distributed NoSQL database for building powerful, intelligent, and mission-critical applications. It stores structured data as a nested series of maps. The MapR JSON table supports the query, read and write for a billion records within a few seconds, while the MapR Binary table is used for time series data visualization in a real-time environment.

5.5 Data Pre-processing Layer

After data acquisition has occurred through the process of collecting, transferring, transforming, storing, and querying during the big data ingestion and management phase, the gathered raw data is delivered in the data preprocessing phase, including sensor selection and noise detection processing to produce clean and healthy data of high quality, which facilitates a promising model in the predictive modelling layer. It is noteworthy that data preprocessing (which is also known as data cleaning) is key to fault diagnosis based on industrial big data since an issue with data quality can lead to significant bias in model training. Clearly, decision making is vulnerable to a misspecified model as it causes various false positives and true negatives. However, as lab-simulated data normally behave well, the data quality issue has not attracted much previous attention. This is not the case in practice, especially for industrial big data. In this study, we incorporate sensor selection and noise detection into the data analytic layer as the data pre-processing phase to complete the practical and integrated architecture.

As shown in *Fig. 5-3*, the IoT raw sensor data are collected through a cyber-physical system. We first propose a feature selection technique to filter out deleterious or noisy features that have a large amount of noise (which can be described as high-noise features), and then a noise detection and cleansing model is applied to the remaining features to obtain clean data for further data analytics. As high-noise features can be identified based on the training data, it is an offline process. Only the remaining features are used for further processing in the fault detection model. On completion of processing the collected data in the data preparation phase, they are passed to the predictive model which performs the diagnosis. The effectiveness of equipment health state detection strongly depends on the quantity and quality of the available sensor data. Thus, preprocessing procedures are of vital importance.

5.5.1 Feature Selection

The choice of a feature selection method is task specific. The collected sensor data is used to monitor the health condition of the manufacturing equipment. Several feature samples from our case study are shown in *Fig. 5-5,* from which we can see that the variables tend to behave stably and when a fault occurs, this is indicated by a large decrease in the variables. Tag 1,2 and tag 6,7 have high variance but are not very useful for detecting faults. If we do not execute feature selection, high-variance features will be involved in the distributed PCA-based predictive modelling phase. The principal idea of PCA is to reduce the dimensionality of a dataset while preserving as much variability as possible. In this sense, these high-variance features will likely be further captured in the PCA model, leading to biased results.

On one hand, this analysis reveals why PCA-based feature selection methods [288] are not suitable for sensor selection before fault detection, as features with large variance can be independent of the detectability of the anomalies. On the other hand, high-variance features should be precluded during the feature selection procedure, in case these noisy features are subsumed into the detection model.

The framework is designed for big data and real manufacturing, where thousands of tags of sensor data are recorded and delivered from the equipment. Rather than elaborately selecting or extracting the discriminate features from dozens of features to improve performance on classification or regression, the selection objective in this thesis is to roughly preclude the redundant and noisy features. Based on the above observation and analysis, to identify such features, we introduce an intuitive and computationally feasible method based on the idea of high variance feature removal.

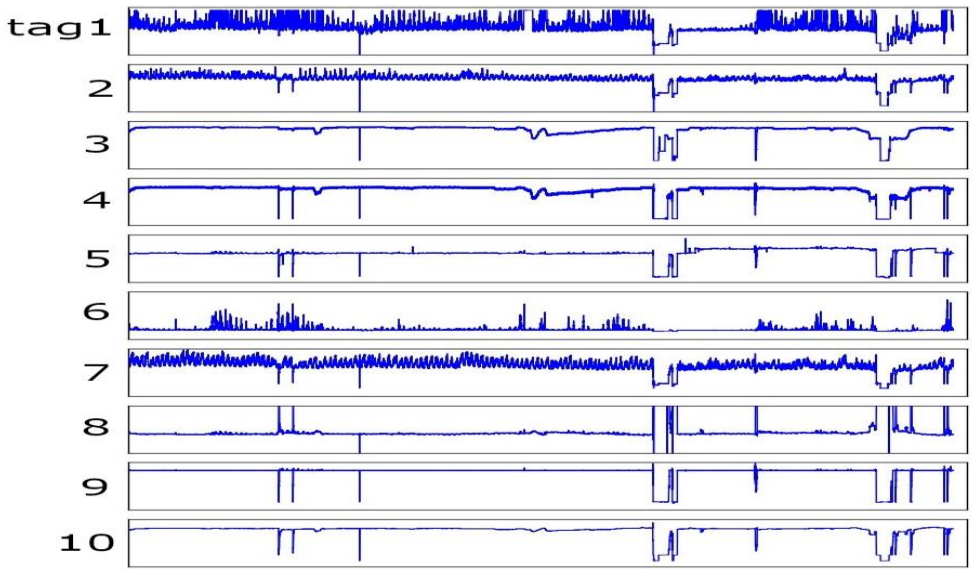


Fig. 5-5: Feature samples over six months

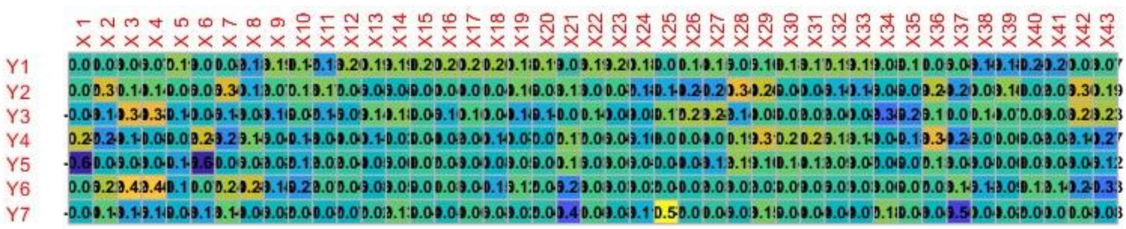


Fig. 5-6: Demonstration of each feature contribution on 40+ features based on the PCA model (without feature selection)

High variance feature removal:

Removing high-variance features is a priority since they are more likely to affect the prediction model compared with the other features. To demonstrate this, we perform PCA on the raw variables without feature selection, which includes the ten features displayed in *Fig. 5-5*. The trained PCA model for fault detection is represented with loading matrix P, which can be used to separate the principal component (PC) subspace and residual subspace. We select 7 PCs to build the PC subspace and define the loading matrix of the PC subspace as $\hat{P}$. When a data point x is projected onto the PC subspace, the compressed value can be expressed as $x\hat{P}$. Therefore, the value in matrix $\hat{P}_{i,j}$ indicates the contribution of the i-th feature to the j-th principal component. *Fig. 5-6* shows the contribution of each feature to the principal components. It can be observed that the previously mentioned unwanted features such as tag 1 and tag 6, however, contribute more to 5-th PC, playing important roles in the trained PCA model. This is why we should pay more attention to removing the noisy features with high variance. Specifically, every feature $x_i(i = 1,2, \cdots, n)$ is assigned a score s using a variance metric, which can be expressed as $s = var(x_i)$, where n is the number of data points in that feature, and $var(\cdot)$ represents the variance of the samples.

A more detailed and advanced method for feature selection has been researched and developed by Yue Hua Liu in a complementary thesis which addresses the problems as they are identified above.

5.5.2 Noise Detection

Noise detection for the equipment health state monitoring of intelligent manufacturing is expected to detect noise while retaining information on anomalies. Compared with general noise detection or outlier detection, handling noise in the presence of significant anomalies is more difficult, as noise and anomalies both belong to outliers and the boundary between them is not easy to define. They are related but distinct. To the data analyst, anomalies are meaningful, whereas noise acts a hindrance to data analysis by obfuscating the true patterns of normal data. In our study, we introduce a novel noise detection method to address the challenges. If this is of interest, more details can be found in our work in [289].

As mentioned in Chapter 3 the problem of the work on Noise Removal was outside the scope of the problem tackled in this thesis and was carried out in a complementary thesis and a summary is reported here for completeness.

The novel noise detection approach in our research is motivated by the aforementioned observations. If only referring to the point value, it is difficult to distinguish noise from anomalies using distance-based threshold methods (*Fig. 5-7 (a)*). So, the unit of our work analysis is not a point, but a window, possibly containing a set of noisy points. The criterion used for measuring the noisiness is calculated based on the similarity between windows, instead of comparing the point value. To measure the similarity, we develop two contrast cues, neighbour and background contrast, through which the difference between noise and anomaly is enlarged and easier to distinguish (*Fig. 5-7 (b)*).

Specifically, the noisy score based on the neighbour contrast of window w_i is calculated by

$$s_{nc_i}(w_i) = d(w_i, w_{il}) \qquad (5.1)$$

where $d(w_i, w_{il})$ is the chi-square distance between the current window w_i and its left adjacent window w_{il}.

The noisy score based on the background contrast of window w_i is,

$$s_{bc_i}(w_i) = \frac{1}{|B|}\sum_{w \in B} d(w_i, w) \qquad (5.2)$$

where $|B|$ is the cardinality of the background base.

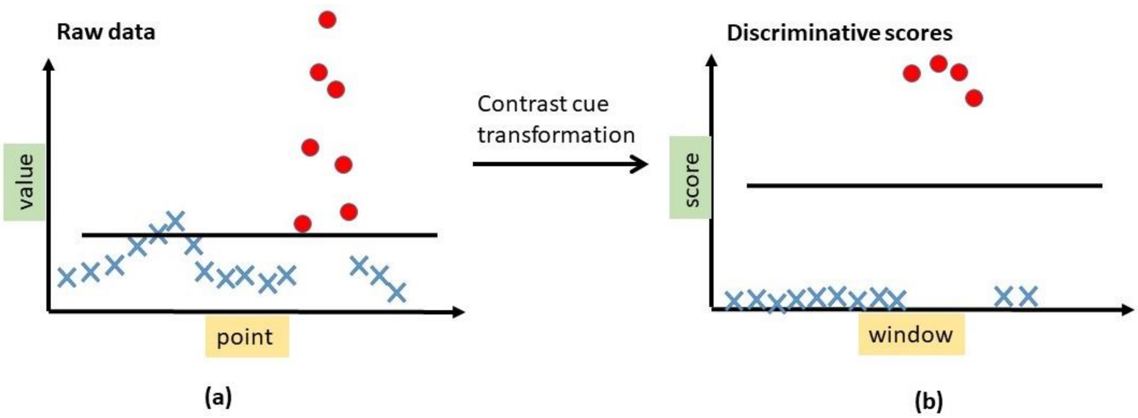


Fig. 5-7: Contrast cue based on window unit

5.6 Predictive Modelling Layer

The previous big data ingestion, management, and preprocessing layers offer rich information and knowledge, and the embedded machine learning techniques in the predictive modelling layer support the intelligent production system to deliver smart business decisions for industrial manufacturing. In a big data and real-time environment, conventional machine learning techniques always suffer from issues relating to a high computational cost and a long running time. Cloud computing is the key technique to address this challenge. In our integrated framework for predictive maintenance, Apache Spark is deployed as the data processing engine to efficiently process the huge amount of data, and MapReduce-based Distributed PCA (DPCA) is introduced as the key technique in the predictive modelling layer for immediate processing, ease of implementation, and extension to a wide range of applications. This layer involves four stages: DPCA model training, streaming anomaly detection, contribution analysis and alarm sequence analysis.

5.6.1 Fault Detection (Anomaly Detection) Model

Faced with the challenges of real manufacturing big data, including uncleaned sensor data without precise labels and the complexity of calculation for high-velocity streaming observations and large quantities of monitoring tags, most conventional machine learning techniques do not work with this. We propose the development of DPCA model to address the above challenges. In the following section, we will first describe the reasons for adopting PCA technique, and then followed by the reasons for the development of the DPCA approach.

The reasons for adopting PCA are threefold. Firstly, PCA can transform the data into a lower dimension by separating the observation space into two subspaces. The produced representations have a better generalization capability than the entire dimensionality of the observation space, and therefore improve proficiency and efficiency. Secondly, the transformed subspace with principal components (PCs) captures the systematic trends of the process, while the other essentially contains the noise or error. Thirdly, such an error subspace can assist in further contribution analysis to identify which variables are responsible for the developing potential fault. The procedures of PCA monitoring the industrial process are listed in *Fig. 5-8*.

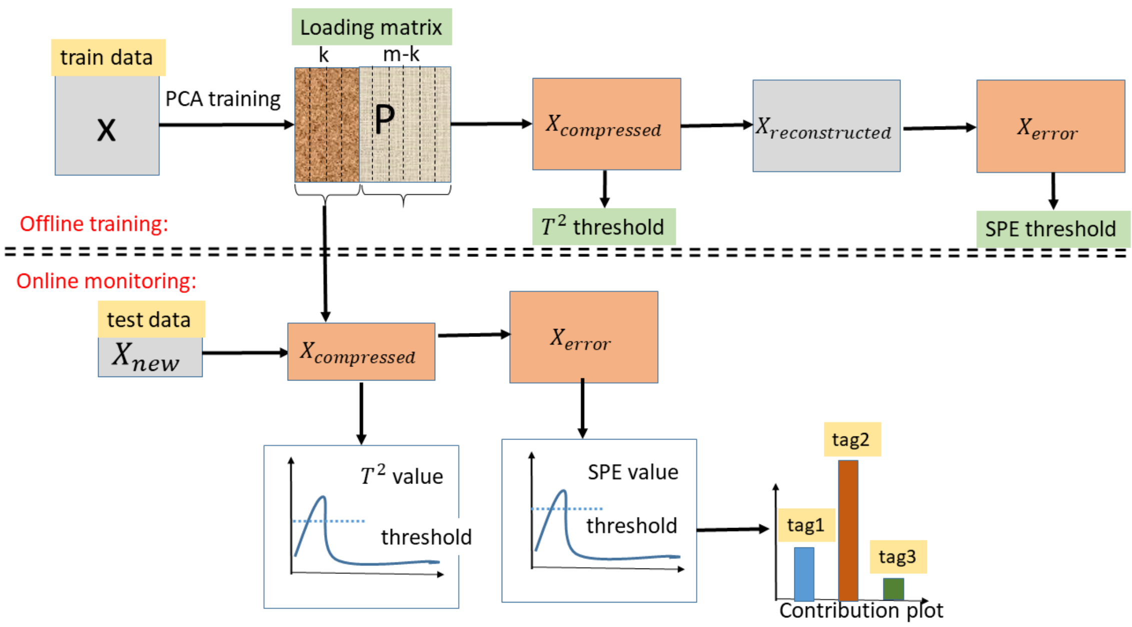


Fig. 5-8: PCA model working principle

But same to other traditional machine learning algorithms, PCA still has implementation bottlenecks when dealing with IoT big data. When we apply PCA in the situation of IIoT, the difficulty is that a large amount of data coming in high velocity, while requiring the immediate response in real time. Such big data real-time processing using PCA may not be able to cope with a high computation load.

Therefore, we need to produce an approach using DPCA to support parallel computation and workload distribution to achieve real-time processing. In DPCA, we break up the processing into several different components each of which is done separately and can be done in parallel. The implementation of DPCA can be enabled on top of the Apache Spark processing engine for fast processing. This is an extra advantage of PCA that it can be modified and updated to accommodate IoT big data by being implemented in a distributed manner. The details of DPCA implementation are described in section 5.6.4.

In addition, all historical data we utilized to train the DPCA model is stored in the Apache Hive data warehouse, which provides a SQL syntax and scheme layer on top of HDFS files. In order to effectively query this large volume of data from Hive when training the DPCA model, we develop a MapReduce-based DPCA model which significantly improves the reading and writing data efficiency.

Technically, PCA transforms the original correlated variables into a new set of uncorrelated variables. The expectation of PCA is that the original variables are sufficiently well correlated so only a relatively small number of new variables (PCs) account for most of the variance. Mathematically, the investigated variables are represented as vector $X = [x_1, x_2, x_3, \dots x_n]^T$, where $x_i = [x_{i1}, x_{i2}, \dots x_{im}]$, n and m represent the number of observations and features.

PCA decomposes the observation vector X into a set of new directions P as:

$$X = TP^T = t_1P_1^T + t_2P_2^T + \cdots + t_mP_m^T \qquad (5.3)$$

where P and T are defined as the loading matrix and score matrix, respectively. Rearranging Eq. 1 such that all terms are grouped into two subspaces, we have the following form,

$$X = T_KP_k^T + T_{m-K}P_{m-K}^T = T_KP_k^T + E \qquad (5.4)$$

where T_K and P_K are the score and the loading matrix of PCs, and E represents a residual error matrix.

Developing potential fault detection based on the DPCA model can proceed in a streaming fashion through two phases.

Phase I: model training

Through the visualization of the observed data, and according to the prior knowledge obtained from engineers, we note that the characteristics of the data variations are relatively unchanged unless a fault occurs in the system. This implies the statistical properties of the data are repeatable and stay the same for the normal operating conditions. Thus, this could be recorded as a reference to identify abnormal conditions, as any observations not consistent with the recorded normally occurring value for certain measures should be recognized as under the out-of-control status. The model training phase is to extract such statistical properties based on the normal data from the in-control status, including the mean and standard deviation, which are used for scaling the continuously incoming streaming data, as well as the threshold of measures, like Hotelling's T-squared and Squared Prediction Error (SPE), to help decide whether new data is abnormal.

Algorithm 2 Training Model

Input: Training dataset $\{x_1, x_2, \dots, x_n\}^T$;

Output: DPCA Model {μ, δ, P, K} and threshold $\{T_a^2, Q_a\}$;

1. Standardize the training dataset, and record the statistical mean μ and standard deviation δ as the reference to preprocess the new coming test data;
2. Estimate the eigenvectors (loadings) and eigenvalues (variance) using efficiently distributed SVD;
3. Determine the meaningful number of components A according to principal components rule;
4. Calculate the threshold of Hotelling's T-squared T_a^2 and SPE Q_a;
5. Store standardization model {μ, δ}, PCA projection model {P, K}, and threshold as baseline parameters $\{T_a^2, Q_a\}$.

The Hotelling's T^2 statistic measure relies on the assumption that the observations are randomly sampled from a multivariate normal distribution. With the level of significance α, the threshold defining unusual patterns is expressed as,

$$T_a^2 = \frac{K(n-1)}{n-K} F_a(K, n-K) \qquad (5.5)$$

where K is the number of PCA components retained in the training model, n is the number of training samples, and $F_a(K, n-K)$ is the upper $100_a\%$ critical point of the F-distribution with m and n − K degrees of freedom.

Additionally, the threshold of SPE can be defined as,

$$Q_a = \theta_1[\frac{h_0 c_a \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2}]^{\frac{1}{h_0}} \qquad (5.6)$$

where:

$$\theta_i = \sum_{j=a+1}^{m} \lambda_j^i \,, h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2} \qquad (5.7)$$

and c_a is the normal deviation corresponding to the upper (1-a) percentile.

Then the steps to implement the training stage are summarized in Algorithm 2.

Phase II: streaming detection

PCA is a reconstruction process of the dataset. While in the online process monitoring system, streaming data keeps coming, and the estimation of new observation x_{new} is the projection t_{new} onto the PCA subspace,

$$t_{new} = x_{new} P_K \qquad (5.8)$$

where P_K is the loading matrix from the pre-trained PCA model, in which case the residual error e_{new} can be expressed by,

$$e_{new} = x_{new} - t_{new} P_K^T = x_{new}(1 - P_K P_K^T) \qquad (5.9)$$

For continuously incoming data, the streaming version of Hotelling's T^2 and SPE can be calculated as:

$$T^2 = \sum_{i=1}^{K} \frac{t_{new,i}^2}{\lambda_i} = t_{new} S^{-1} t_{new}^T$$

$$= (x_{new} P_K) S^{-1} (x_{new} P_K)^T \qquad (5.10)$$

$$SPE = \sqrt{e_{new} \cdot e_{new}^T} \qquad (5.11)$$

where S is a diagonal matrix, which is the estimated covariance matrix of the principal component scores, $t_{new,i}$ and λ_i are the new score and variance for the i-th PC, respectively.

T^2 value indicates the deviation from normal values inside A-dimensional PC subspace. A larger SPE value means that the data point x_{new} goes into the residual subspace, which implies the current observation is far off and inconsistent with the model trained using normal data, indicating a potential problem occurring within the industrial process. To provide meaningful information to the engineer, a deterministic mechanism is designed based on the count of the number of times the threshold is exceeded for every 5 minutes with rolling windows to identify whether it is a true anomaly or just noise. Only a small sample data set in the industrial process cannot indicate the sensor fault caused by random error or accidental error. For instance, if only 3 observations out of 300 records exceed the threshold in 5 minutes, these unusual patterns will be considered as noise instead of true anomalies. The statistical result of anomaly detection is defined as

$$I_s: \begin{cases} anomaly, \ if \ \beta \geq \gamma; \\ normal, \ if \ \beta < \gamma, \end{cases} \tag{5.12}$$

where $\beta = \frac{|T^2 > T_a^2|(or|SPE > Q_a|)}{L_{rw}}$, $|T^2 > T_a^2|(or|SPE > Q_a|)$ indicates the number of points which satisfy the condition within $|\cdot|$, and L_{rw} represents the length of 5 minute rolling windows, γ is the deterministic boundary experimentally set based on expert knowledge. The steps for streaming anomaly detection are summarized in Algorithm 3.

Algorithm 3 Detect Anomalies

Input: streaming observation x_{new} and pre-trained model {μ, δ, P, K, T_a^2, Qa};

Output: signal of anomaly or normal;

1. Standardize x_{new} using μ, δ;
2. **for** each observation x_{new} **do**
3. Compute score values t_{new};
4. Estimate Hotelling's T-squared and SPE values in streaming manner;
5. Perform deterministic mechanism to identify anomalies.
6. **end for**

The historical data in the Apache Hive data warehouse collected from the historical server is used to train the DPCA model to represent the production in-control condition, which can be treated as the reference to check whether the real-time incoming monitoring sensor data is inconsistent with the in-control condition. Due to the characteristics of IoT monitoring sensor data, it is noted that the measurement readings normally stay stable within a certain range. However, the trained DPCA model and the involved thresholds need periodical necessary updates due to equipment degradation which leads to an aging trend in the predictive modelling phase. In particular, when a sensor is replaced, the update operations are extremely important. The most straightforward and simple way is to check and update the thresholds periodically by retraining the PCA model. Encouragingly, the data distribution will not change in the short term. As the aging trend is always a long-term slow process, periodical updates should solve this problem and allow model evolution to be tracked due to aging. Given the pre-trained DPCA representation model, every batch of optimized structured data available in Apache Hive will be fed into the prediction model in nearly real time to produce anomaly detection and root cause results, which are then stored in an anomaly MapR table as shown in *Fig. 5-2*.

5.6.2 Variable Contribution Analysis

Once an anomaly is detected, it is important to diagnose which component in the large-sale industrial manufacturing system is causing the failure. The delivered information provides evidence to trace back to the parts of the machine potentially likely to breakdown. Residuals can be utilized to help suggest the possible candidate and its degree of contribution to the fault. Comparing each term's residual error e_{new} is one of the most straightforward contribution analysis methods. The responsible sensor which has the largest error values should be identified as the root cause and given the necessary treatment.

After applying the PCA method, a compressed value which is transformed into the principle component subspace is derived for Hotelling's T^2 to detect the failure, and a reconstructed value is derived for the computation of the SPE metric. The SPE value is computed as the difference between the original value and the reconstructed value, and SPE has the same dimensions as the original value which is exactly the number of features. Therefore, the feature with the largest difference implies possible failures are associated with that feature.

5.6.3 Alarm Sequence Analysis

When the T-squared or SPE statistics exceed the pre-trained thresholds, the monitoring system starts to alarm. The alarm sequence is formed when a number of alarms occur. Alarm correlation is the analysis of the alarm relation, transforming a number of alarms into fewer but more informative alarms through alarm compression and conversion. One of the important types of alarm sequence analysis is alarm compression $[A, A, \ldots, A] \rightarrow A$, which means if the alarms are received in a time window and there are multiple occurrences of the same event, then the multiple alarms are replaced by a single alarm [290]. In addition, based on 5-min alarm conditions, noisy anomalies' minutes which only send an alarm for a period of less than 5 min are suppressed and the proposed framework only reports real anomalies to warn the system.

The detected alarms are delivered to an alarm database and the data lake dashboard extracts the detected alarms from the MapR database alarm table and displays the results to the engineer in the data visualization layer. Given detailed alarm information and logs, including the time period of the alarm occurrence and the contribution vector, more comprehensive alarm analysis modelling can be carried out for the purpose of generating a comprehensive report, thereby giving the decision-maker more insight into the equipment health state. The proposed ecosystem for predictive maintenance enables the engineer to analyze the anomalies before the deleterious events happen, and the architecture provides implementation guidelines for the development of IoT-based smart manufacturing, which can be used in different domains for various applications to build smart Industry 4.0 factories.

5.6.4 DPCA Implementation Through Distributed Singular Value Decomposition

Generally, in order to build a PCA model, we need to estimate the eigenvectors (loadings) and eigenvalues (variance) by singular value decomposition (SVD). The single processor PCA can use the single-core ARPACK package for calculating the eigenvalue decomposition of a matrix and was used widely. Generally, for an $n \times m$ matrix A, the SVD factorizes to the form

$$A_{n \times m} = U \Sigma \mathrm{V}^T \qquad (5.13)$$

However, this approach is not suitable for the DPCA. We need to have a different approach for the matrix decomposition in order to achieve distributed and parallel processing, particularly when the input matrix is using the training dataset which contains the recorded readings from a large number of different sensors. The proposed approach we utilized in our DPCA training phase utilizes a matrix decomposition which is as follows.

Since matrix $A_{n \times m}$ is tall and skinny (m ≫ n), a different algorithm is used to compute U. First, we compute Σ and V by solving

$$A^T A = U \Sigma^2 \mathrm{V}^T \qquad (5.14)$$

which is of dimension $n \times n$, therefore it is small enough to fit in the driver memory. Σ and V can be retrieved directly and locally on the driver node. Once they are computed, U can be recovered by

$$U = A \mathrm{V} \Sigma^{-1} \qquad (5.15)$$

which is derived from $A = U \Sigma \mathrm{V}^T$. The last two terms $\Sigma \mathrm{V}^T$ are easy to compute, and we can distribute the computation of $U = A(\mathrm{V}\Sigma^{-1})$ by broadcasting $\mathrm{V}\Sigma^{-1}$ to all nodes holding rows of U in a parallel way. This is then used to determine the eigenvectors and eigenvalues in a distributed fashion.

This new decomposition approach allows us to carry out distributed and parallel processing which is necessary during the training phase implementation. In this sense, the implementation of DPCA enables to deal with big data collected from IoT systems and can greatly save the processing time.

5.7 Conclusion

The emerging IoT has resulted in a new generation of intelligent digital sensors, which are gradually being implemented in smart manufacturing. IoT technology bridges the gap between physical equipment and cyber systems, enabling continuous data gathering, comprehensive data analysis, and real-time monitoring. This chapter proposes an integrated framework for manufacturing equipment health state monitoring and diagnostics using IoT big data in a real industrial setting, which can be easily used for smart manufacturing and implemented in Industry 4.0 systems. The intelligent sensors play a pivotal role in the success of IoT and the connectivity across networks results in massive volumes of streaming data. Big data brings many challenges, and equipment health state monitoring for IoT smart manufacturing requires evolution. The proposed framework provides an integrated solution to this issue, comprising a big data ingestion phase, big data management phase, data preparation phase, and predictive modelling phase. In the data ingestion and management phase, data lake techniques are employed for flexible data storage and faster data retrieval. The big data flows between the collectors, servers, and clouds across the connected ingestion system. Sensor selection and noisy detection approaches are incorporated in the data preparation phase for efficient modelling and analytics. Given the large volume of rapidly arriving real-time data, based on the proposed big data ingestion system, unsupervised distributed PCA is embedded in predictive modelling for event diagnosis due to its capability for online processing, its efficiency, and ease of deployment. The overall framework is implemented in the Apache Spark platform, which provides an environment to make it available for distribution and cloud computing to process the big data.

Chapter 6
Industrial Case Study, Results and Evaluation

To evaluate the architectural framework and techniques proposed in Chapter 5, this chapter conducts two industry case studies by deploying the proposed framework in a global manufacturing system at our partner company in Australia, Incitec Pivot Limited (IPL), which invested in the research project aiming to improve the operational efficiency of its manufacturing business by predicting and preventing equipment breakdown. As shown in *Fig. 6-1*, IPL is a global manufacturing facility which owns and operates 20 manufacturing plants in the US, Canada, Australia, Mexico, Indonesia and Turkey, producing a wide range of explosives, fertilisers and industrial chemicals. Our proposed IoT-based big data ecosystem was implemented and tested in a central office located in Melbourne to collect and manage all the data from various manufacturing plants in Australia to monitor all equipment health states to predict and prevent any failures. This research project won the Best Industry Application of IoT at the BigInsights Data and AI Innovation Awards in 2018 [227]. The experiment results are demonstrated in two case studies. The first case study aims to validate the whole ecosystem and the fault detection algorithm by utilizing expert knowledge for data preprocessing, while the second case study compares the predictive model performance before and after embedding our noise detection and feature selection techniques.

6.1 Industrial Case Study I

As shown in *Fig. 6-2*, our proposed ecosystem deployed in the central office in Melbourne is collecting data from different manufacturing plants in Australia to monitor the health state of all the equipment, such as various types of compressors, pressure swing adsorbers and so on. To verify the proposed ecosystem, both case studies in this thesis concentrate on the Ammonia plant at the Phosphate Hill site in North West Queensland with ingested data from 2013 to 2018 based on one of these compressors, namely the syngas reciprocating compressor. This site is located in a remote area and approximately 2500km from Melbourne. The big sensor data ingestion, management, and analytics are incorporated to detect unusual patterns in the syngas reciprocating compressor, so that engineers can make decisions according to the monitored operating conditions. The whole architecture is deployed on the cloud cluster with multiple nodes to achieve large-scale data flowing and processing. One example test case is reported in this section in detail to demonstrate the validity of the proposed architecture.

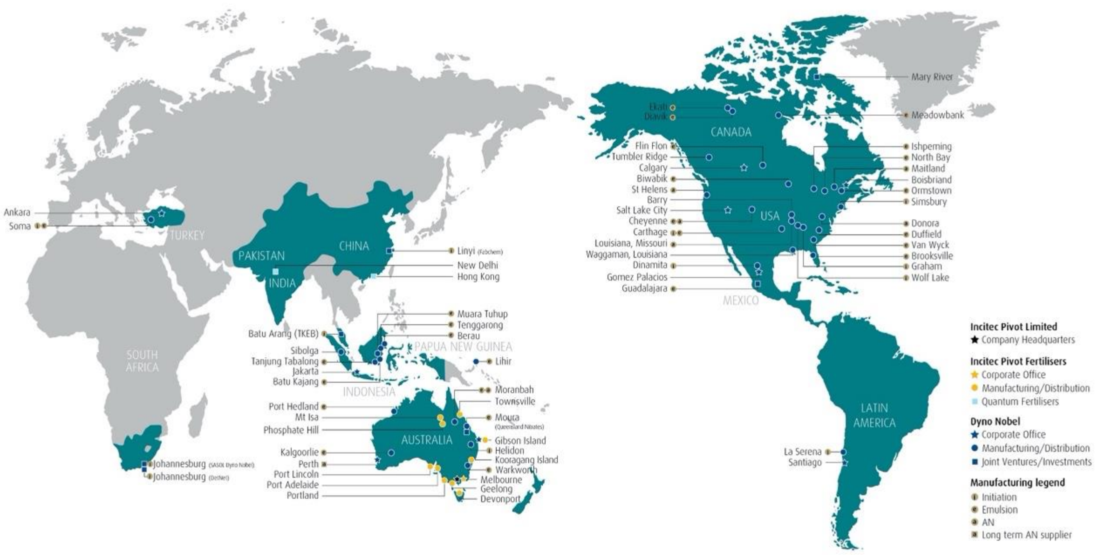


Fig. 6-1: IPL manufacturing plants across the world

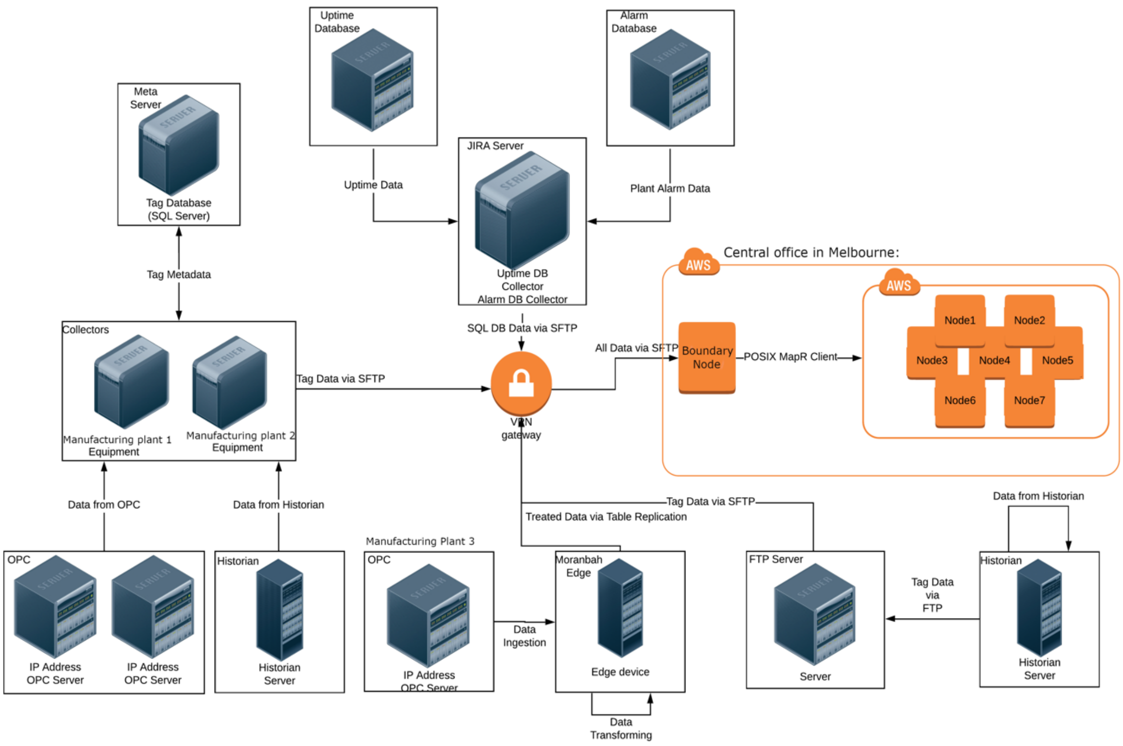


Fig. 6-2: Data ingestion architecture and flow, collecting data from different manufacturing plants

6.1.1 Experiment Environment Setup

The whole architecture is deployed on the Apache Spark cluster with multiple nodes to achieve large-scale data flowing and processing. The Apache Spark is a fast and general cluster computing system for big data, and it has a unified analytics engine for large-scale data processing. In 2012, Spark and its unique resilient distributed dataset were developed to overcome the limitations of the MapReduce cluster computing paradigm, which forces a particularly linear dataflow structure on the distributed program. Spark runs data science workloads which are up to 100× faster in memory, or 10× faster on disk than Hadoop. It has convenient APIs for operating on large datasets and provides high-level libraries, including support for SQL queries, streaming data, machine learning, and graph processing.

DPCA Performance with Apache Spark: The DPCA model's performance was significantly improved with Apache Spark. A fast run time can be achieved when handling real-world big datasets. For input matrix $A_{n \times m}$, the following conditions hold.

1) If we compute U locally on the driver, this requires a single pass with $O(n^2)$ storage on each executor and on the driver, and $O(n^3)$ time on the driver.
2) Otherwise, we compute U in a distributed way on the driver node. This requires O(n) passes, O(n) storage on each executor, and $O(n^2)$ storage on the driver.

The developed framework has been tested over the years on an actual production compressor. Six months of monitoring results are reported in this section from August 2014 to February 2015 to demonstrate the validity of the proposed framework. According to the engineers' feedback, no fault occurred during the first three months, so the data collected during this period are used for training the model, and the data collected in the last three months are used for testing the model. In addition, the online detection demonstration for 2018 events which was done before the developing breakdown actually occurred is even more impressive in validating the methodology.

6.1.2 Test Data Description

A reciprocating compressor is a special piece of equipment which demands predictive maintenance. It has been widely used in a variety of industries and manufacturing plants, for instance, in the chemical industry. Many companies use reciprocating compressors like an energy source, powering equipment and tools. For many industrial uses that require the safety of a non-heat producing power source and a reliable flow of that power, the reciprocating compressor is the only option. In manufacturing processes, whether in refineries, plastics, assembly plants or metal fabrication, a compressor system is the central power source that keeps the business in production. However, due to the complicated structure of the reciprocating compressor and the requirement to often work in harsh environments, it is one of the machines most likely to fail from a variety of faults during operation. Therefore, there is a great need to monitor the compressor health condition in an online fashion so that potential developing faults can be detected and addressed in advance of actual breakdown.

The informative sensor data that is employed in developing the fault detection system is collected from the turbine syngas compressor every second. *Fig. 6-3* describes the working principle of the four-stage compressor and demonstrates the sensor location. Hundreds of sensor features including various vibrations, temperature, pressure, speed and so on are generated continuously. The detection model is implemented and computed on Apache Spark using the data dragged from two different sources, the OPC server and Historian. *Table 6-1* summarizes the ingested IoT data from the historical servers over the last five years (01/08/2013–01/08/2018). From the analysis results based on the implemented data lake architecture, the proposed ecosystem runs stably and demonstrates the ability and effectiveness to handle a massive amount of equipment data. In addition, the records can be queried from over a billion data within a few seconds for any further data analysis tasks using our designed and deployed NoSQL database.

Table 6-1: Ingestion analysis

Total ingested data range (syngas compressor)	01/08/2013 – 01/08/2018
Average ingested data size daily	663MB
Average ingested data entries daily	57 million
Total archived data size (including log files, maintenance records, uptime, tags, etc.)	3TB+

To evaluate the model performance and verify the experiment results, we extensively proceed with testing and streaming detection based on sensor data from 2013 to 2018, and the engineer provided two main functional failures of the syngas compressor during the testing case time, which occur at 26/07/2014 02:24 and 04/09/2014 17:11. Detecting anomalies at the indicated time when they occurred will not be useful for engineers to prevent machine functional failure, as engineers need sufficient time in advance of the deleterious event to determine solutions for the detected anomalies. Therefore, the primary objective of these experiments is to detect anomalies in advance of the indicated occurrence time of the event, which could be a few hours or a few days before the equipment functional failure.

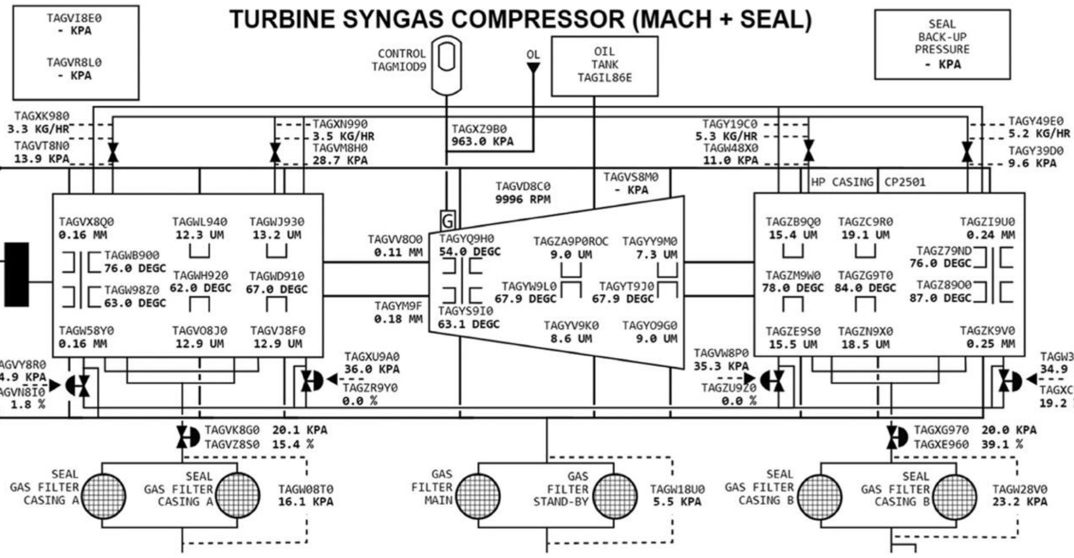


Fig. 6-3: Working principle of the four-stage turbine syngas compressor showing the sensor location.

6.1.3 Data Preprocessing Using Expert Knowledge

To effectively extract the information in the data which is relevant to developing a potential fault detection in predictive maintenance, it is necessary to preprocess the data. This procedure generally consists of three tasks: removing irrelevant variables, autoscaling, and removing outliers [291]. The data may contain variables that have no information relevant to detection, which should be removed to improve the proficiency of the detection method. The recognition of such meaningless variables often needs prior knowledge from domain experts. In this case study for our proposed framework, after the associated statistical analysis, 17 variables are selected to predict the compressor health condition.

The industrial process data with various variables are often located at different levels and have arbitrary units. *Fig. 6-4* shows the summary statistics (number of observations, mean, standard deviation, minimum and maximum values) for the year 2014 to 2015 compressor sensor data with a few selected features. It can be clearly seen that multiple attributes have a wide range of values. Hence, raw data needs to be scaled to avoid particular variables dominating the predictive method, especially for PCA. For example, when performing an unscaled dimensionality reduction procedure on the six measurements shown in *Fig. 6-4*, tag 5 varying around 20K would dominate even though it may be no more important than tag 1 for monitoring the process. Centering and scaling are the two most common types of pre-processing approaches to standardize the dataset.

summary	45xi25500x_s_pv	45xi25500y_s_pv	45si25601a_s_pv	45ti25512_s_pv	45fic25001_s_pv	45pi25512_pv
count	26481815	26481815	26481815	26481815	26481815	26481815
mean	29.460029763204727	46.637517128631316	10439.96282101341	69.84513830566745	19246.610010505843	1024.6785756249617
stddev	7.717686337499979	14.410414174820069	307.511146413452	5.999506214052661	2338.54345953253	34.63487699033685
min	3.9300708770752	2.006760597229	8000.13134765625	25.1748275756836	10000.00390625	45.8456420898438
max	77.0064697265625	79.9906768798828	11550.626953125	101.0	28806.79296875	4615.72607421875

Fig. 6-4: Summary statistics of several selected features

Centering removes any bias terms from the data by subtracting the mean value from each column in the matrix. For the k^{th} column:

$$X_{k,center} = X_k - \bar{X}_k$$

Scaling removes the fact that the raw data could be in diverse units:

$$X_{k,scale} = \frac{X_k - \bar{X}_k}{\delta(X_k - \bar{X}_k)}$$

where $\bar{X}_k$ and $\delta(\cdot)$ represent the mean value and standard deviation operation. Then each column X_k is collected back to form matrix X. This pre-processing is also called autoscaling which centers each column to zero mean and then scales it to have unit variance. Autoscaling standardizes the variables in a way that ensures each variable is given equal weight before the application of the detection method. After this pre-processing, each column will have a mean of 0.0 and a variance of 1.0. Note, centering and scaling does not alter the overall interpretation of the data. For example, if two variables are strongly correlated before pre-processing, they will still be strongly correlated after auto-scaling.

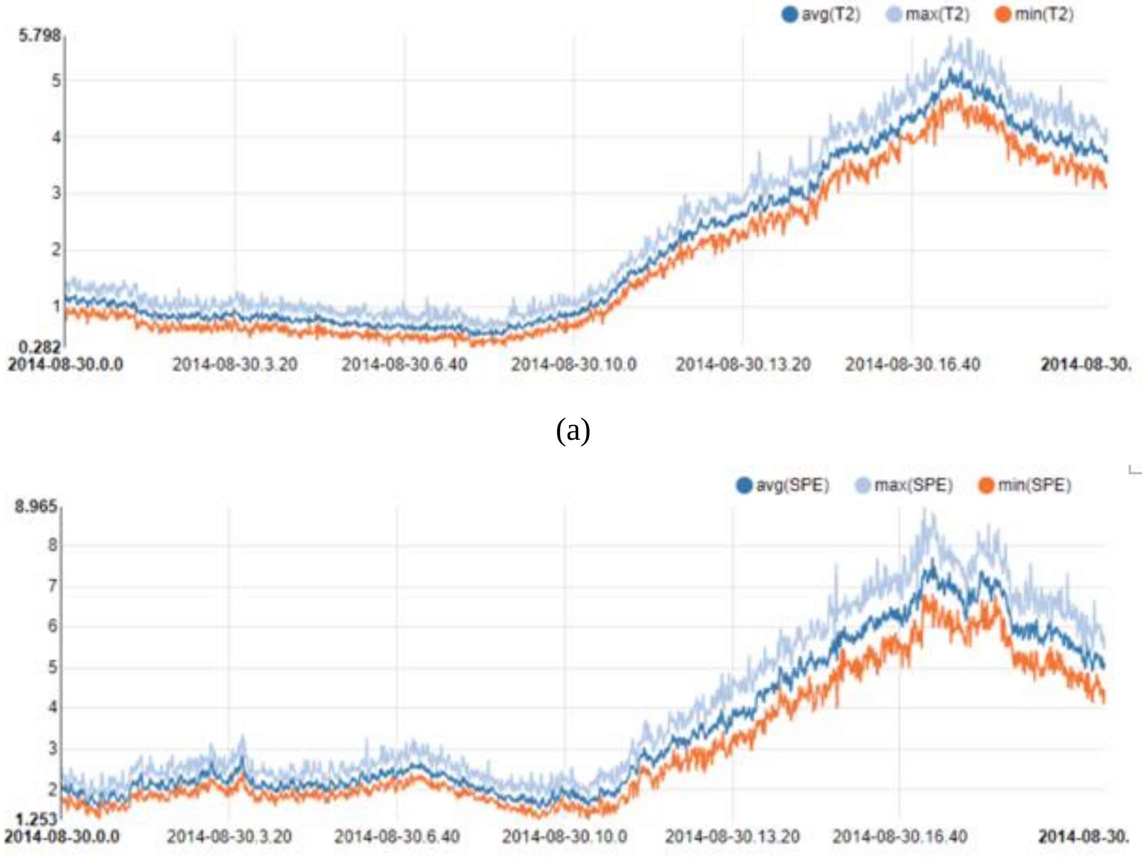
Outliers are isolated values that are erroneous. These values may greatly influence the parameter estimation involved in the detection model. Based on the knowledge from the engineer visually inspecting the data, several specific rules are defined to undertake the pre-treatment of data to remove outliers arising from bad data or noise. For example, all the variables of one observation are removed if the speed feature value of the syngas compressor is lower than 8000, as a syngas compressor speed lower than 8000 indicates that the machine has shut down at this moment. The collected sensor data should be discarded to avoid noisy information when machines do not operate.

6.1.4 Test Results and Evaluation

Table 6-2 presents the test results for the main functional failures (1) and (2) of the syngas compressor in 2014. A clean dataset from 01/01/2014 to 31/05/2014 is selected to train the model. Rotating equipment functional failure time at 26/07/2014 02:24 is detected successfully by both T-squared at 19/07/2014 01:30 (seven days in advance) and SPE algorithms 20/07/2014 12:00 (six days in advance). For the other rotating equipment functional failure at 04/09/2014 17:11, both T-squared and SPE deliver warnings three days in advance, at the end of 01/09/2014. *Fig. 6-5(a)* demonstrates the T-squared statistic count of average, maximum, and minimum values based on the observations every minute from 2014-08-30 00:00 to 2014-08-31 00:00, while *Fig. 6-5(b)* shows the SPE statistic average, maximum, and minimum changes. Both the T-squared and SPE detected results suddenly increase at 2014/08/30 10:00 compared to the normal condition, and the model starts to alarm the system on 01/09/2014 by T-squared and 31/08/2014 by SPE. The times between the alarmed timestamp and the occurrence of the actual event give the engineer sufficient time to determine a solution.

Table 6-2: Test results containing two outages in 2014

Testing Case 1 (Outage 1, 2)			
Pre-condition	Use specific range to filter dirty data based on engineer feedback (such as speed over 8000)		
Test Fault ID	Test Title	Test Result	Test Comment
(1)	Detect anomalies using T-squared Algorithms	Success	Detect anomalies start at 19/07/2014 01:30
(1)	Detect anomalies using SPE Algorithms	Success	Detect anomalies start at 20/07/2014 12:00
(1)	Variable Contribution analysis using SPE Algorithms	Sensor variable 45xi25502y_s_pv contributes most when anomalies occur	
(2)	Detect anomalies using T-squared Algorithms	Success	Detect anomalies start at 01/09/2014 22:56
(2)	Detect anomalies using SPE Algorithms	Success	Detect anomalies start at 31/08/2014 11:52
(2)	Variable Contribution analysis by SPE Algorithms	Sensor variable 45ti25525_s_pv contributes most when anomalies occur	



(a)

(b)

Fig. 6-5: Curve change of statistic average, maximum and minimum per minute based on (a) T-squared and (b) SPE.

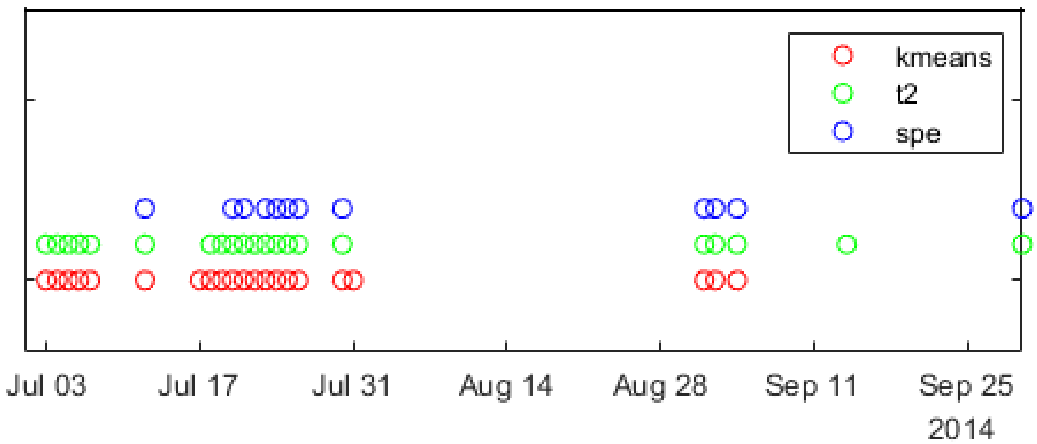


Fig. 6-6: Comparative results based on K-means and DPCA

K-means clustering is one of the most commonly used unsupervised anomaly detection algorithms in real industry. *Fig. 6-6* presents the comparative results of the distributed *K*-means clustering, DPCA-based T-squared, and SPE algorithms. The detected results show the following.

1) In addition to the compressor failures in the historical alarm records of which the

engineers are already aware, all the tested algorithms capture more outages. To verify if there are real anomalies during this period instead of treating these detected results as false positives, we further check the monitoring of the syngas compressor running speed with expert knowledge. *Fig. 6-7* shows the equipment speed drops significantly on 12/07/2014, 26/07/2014, 30/07/2014, 01/09/2014, and 04/09/2014. This implies that the detected results successfully match the equipment running condition, and the machine downtime periods could be reflected in the additional outages.

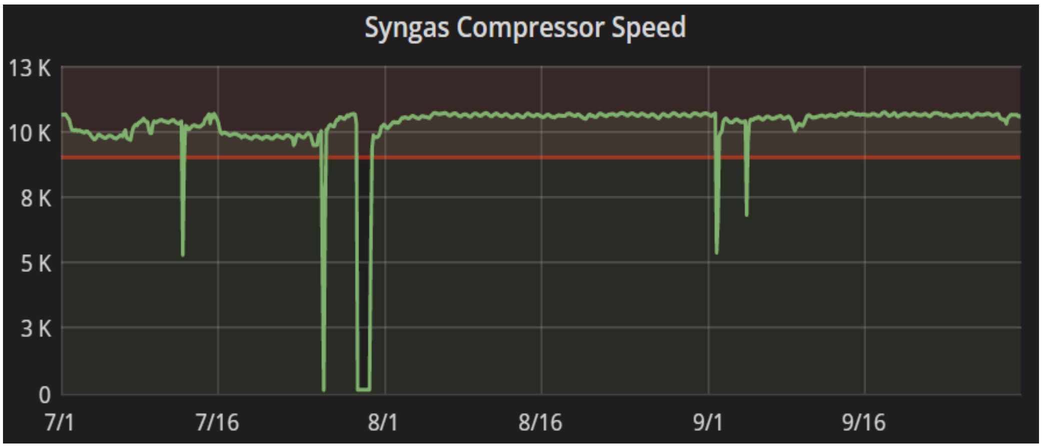


Fig. 6-7: Speed sensor monitoring of the equipment

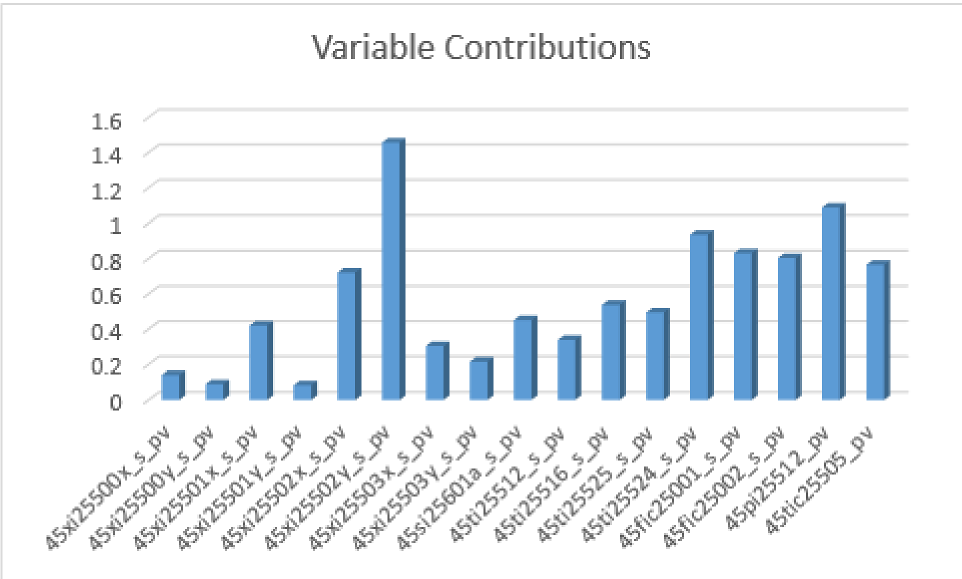
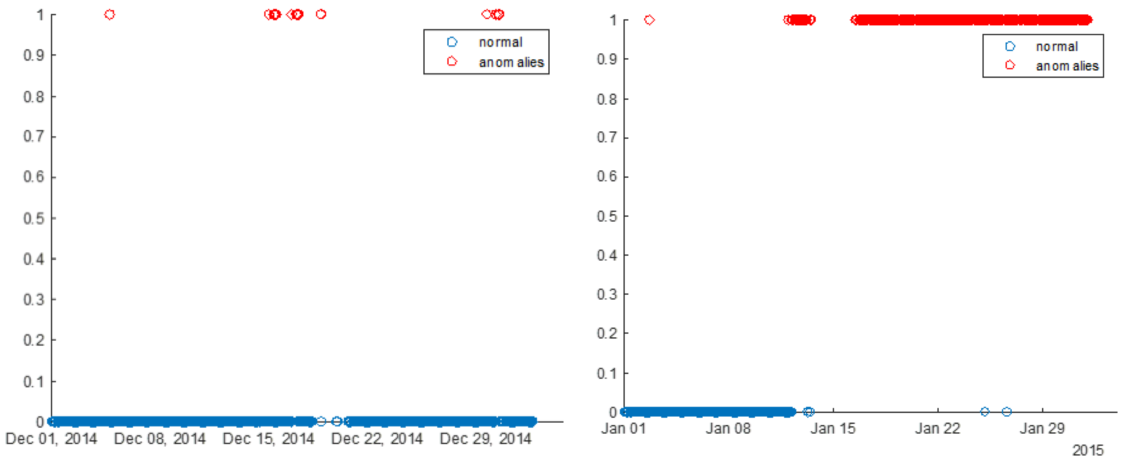


Fig. 6-8: Variable contribution analysis of fault (1)

2) Both the T-squared and distributed *K*-means algorithms capture more outages than SPE from 03/07/2014 to 07/07/2014, which turn out to be false positives. It can be concluded that SPE is more stable in terms of fault detection, while *K*-means and T-squared algorithms are more sensitive than SPE for multivariable value changes.

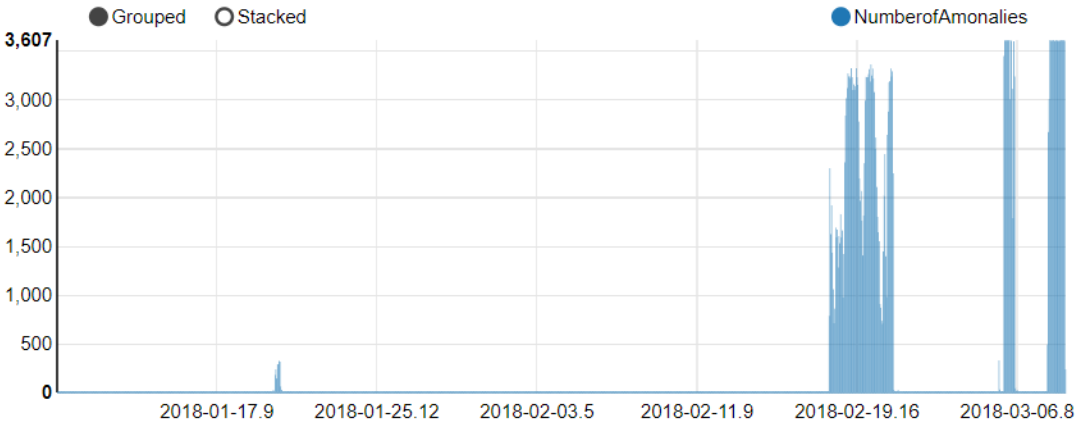
Industry prefers stable detection results instead of having to investigate more alarms and thereby use more resources. Once the system has been alarmed, it is essential for engineers to know the main reason for the occurrence of the alarm to conduct specific maintenance. *Fig. 6-8* shows the weights of each variable's contribution to the SPE values at time 2014/07/26 02:00:01. The sensor 45xi25502y_s_pv contributes most significantly to the detected SPE value. In addition, the ranking of all variable contributions can also provide useful information for investigating the cause of an alarm.



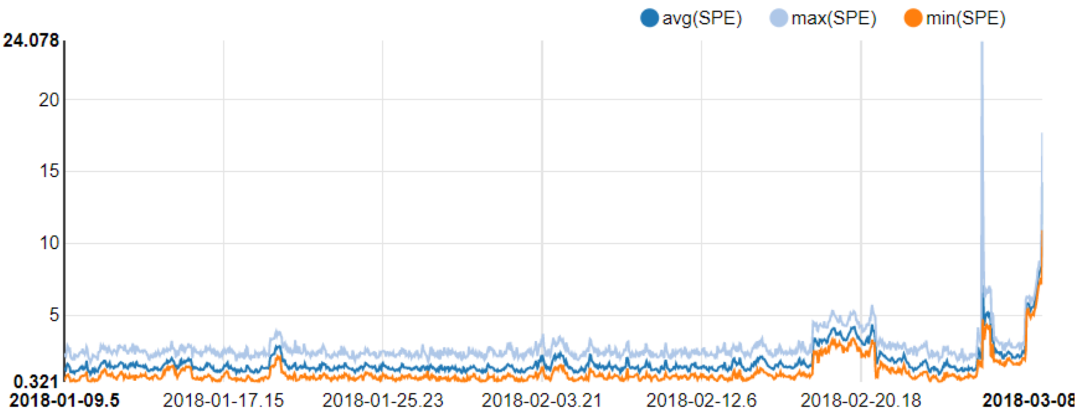
(a) (b)

Fig. 6-9: Record of anomalies in 5-min rolling windows

5-min Deterministic Mechanism*:* During the detection process, small inconsistent anomalies occur. The values only exceed the thresholds a few times, rather than triggering the alarm consistently for hours. These anomalies may not reflect the real condition of the compressor and can be considered noise instead of true significant anomalies due to the incorrect data information. The deterministic mechanism is applied to distinguish a real fault event. In every 5-min window, if more than 15 anomalies out of 300 observations are detected, then the timestamp points to 1, otherwise 0. The determined values 0 or 1 will be delivered to the engineer to monitor the condition of the compressor. If the observations continue to result in the consistent recording of anomalies over the following 5-min sliding windows, the engineer will consider that it is likely that significant abnormal behaviors for the syngas compressor have occurred during this time. *Fig. 6-9* shows the SPE records of anomalies in the 5-min rolling windows from 01/12/2014 to 31/12/2014 and 01/01/2015 to 31/01/2015 in *Fig. 6-9(a)* and *(b)*. The observations pointed out in 04/12/2014 only alarm the system for a few minutes, after which the compressors operate in stable mode again. These small interval anomalies are considered to be noise instead of true significant anomalies. It appears that anomalies occurring at 12/01/2015 continue to deliver warnings to the system until the end of February. A serious issue is reported to the engineer during this period.



(a)



(b)

Fig. 6-10: Streaming detection results. (a) Number of detected anomalies per hour based on SPE for online 2018 data, (b) Statistical average, maximum and minimum SPE value during the corresponding time periods.

Streaming Results in Actual Production: Unknown Event Detection Online in Real Time

In addition to testing for the previous faults provided by the engineer, the detection for unknown events in 2018 was also tested by the trained model with the SPE algorithm. Sensor data from January to March are fitted into the model to detect whether there are anomalies. *Fig. 6-10(a)* shows the detected number of anomalies based on every hour for the test dataset in 2018. *Fig. 6-10(b)* demonstrates the statistical average, maximum and minimum to capture the change of SPE. The model detects three events, including one small outage at 2018-01-20 06h and two large outages at 2018-02-18 05h and 2018-03-05

16h. There are gaps in the data between 2018-03-02 and 2018-03-05 due to machine shutdowns. The small outage alarms the system within one hour, then the SPE values behave normally and stably again, while the other two outages keep alarming the system for several days. Thus, the first outage is considered to be an anomaly and is reported to the engineer to check the compressor situation.

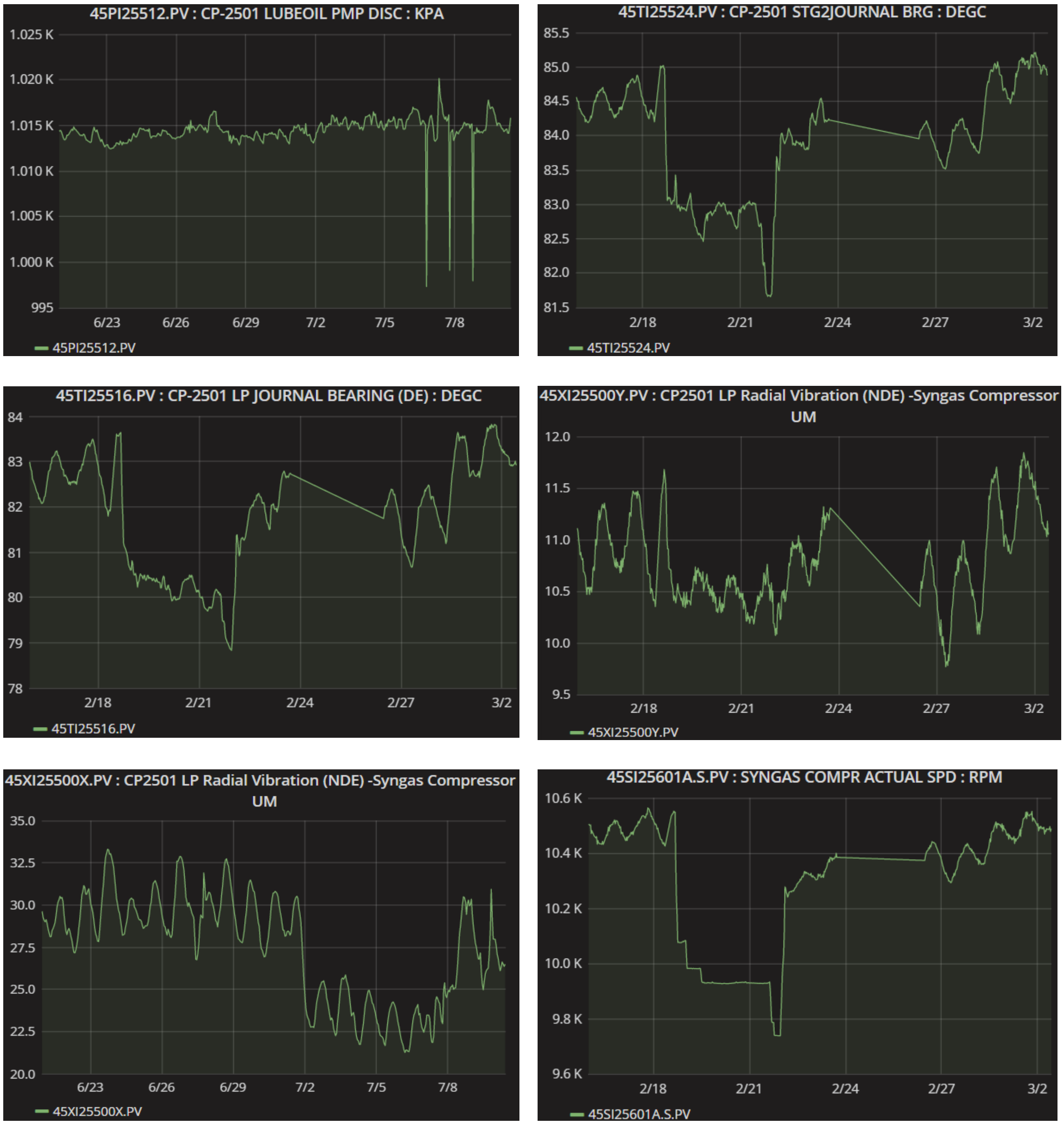


Fig. 6-11: Monitoring of several features in 2018

For the first large outage, the SPE values start to increase significantly at 2018-02-18 05:24, and the value range stays between 2.5 to 3.5 instead of the previous 1.0 to 2.0. This significant change of SPE indicates an unusual event may happen to the syngas compressor. *Fig. 6-11* describes the monitoring of some features during this event time. It appears that temperatures, vibration, pressure and speed from 2018-02-18 to 2018-02-22 are abnormal compared to normal operation. Machine speed decreases to be lower than 10000 but greater than 8000, which indicates the machine is still operating during this time in an unnatural condition. Information on the warning alarm for the operational status of the syngas compressor was delivered to the engineer from this detected outcome to avoid possible damage. The second large outage was detected 3 days before the engineers were aware of the compressor breakdown on 2018-03-08, providing sufficient time to prevent the potential event. The SPE value gradually increases over 2.0 after 2018-03-02 12:40, then dramatically increases over 4.0 after 2018-03-05 16h. This detected result delivers a message that there may be a compressor issue which started on 2018-03-05 16h. Subsequent to the feedback to the engineer for the detected results, they learned that two unscheduled downtime losses occurred between 2018-02-18 to 2018-02-21, and another after 2018-03-08 for the syngas compressor, which caused the closure of the machine. Therefore, our model accurately detected these two anomalies for new unknown events even before the engineers were aware that they would occur, and 3 days in advance for the syngas compressor downtime on 2018-03-08.

6.2 Industrial Case Study II

Case study I demonstrates the effectiveness of both the proposed IoT-architecture and techniques for predictive maintenance in both big data and real-time environments. However, specific rules are defined by the engineers based on the domain knowledge to select sensors and filter noise. This case study focuses on solving the data quality issues to improve the predictive model performance by implementing our proposed feature selection and noise detection methods instead of using expert knowledge for filtering features and noise. In addition to the framework, a comprehensive report is generated to help engineers make smart decisions in a proactive mode, instead of the traditional reactive mode to fix an equipment failure which has already occurred, greatly increasing customer satisfaction.

In the previous case study, 17 variables were eventually chosen after the associated statistical analysis, but in general, there were hundreds or even thousands of features, in which case identifying the representative and contributive features was not easy and largely dependent on the domain expert's knowledge. In this case study, the installed sensors read data every second for 43 features across the four-stage compressor, including temperature, pressure, vibration, speed and so on.

6.2.1 Feature Selection

After the sensor data is ingested and stored in the cluster in the proper format, 43 features from different sensor locations show the presence of noise. Ten of these samples are illustrated in *Fig. 5-5*. Initially, feature selection is employed to filter out the noisy features to ensure there are no misleading results on system events. The original 43 features are reduced to 36 after the high-variance noisy features are filtered. A visual comparison of the T-squared and SPE statistics before and after the removal of the noisy features is given in *Fig. 6-12*.

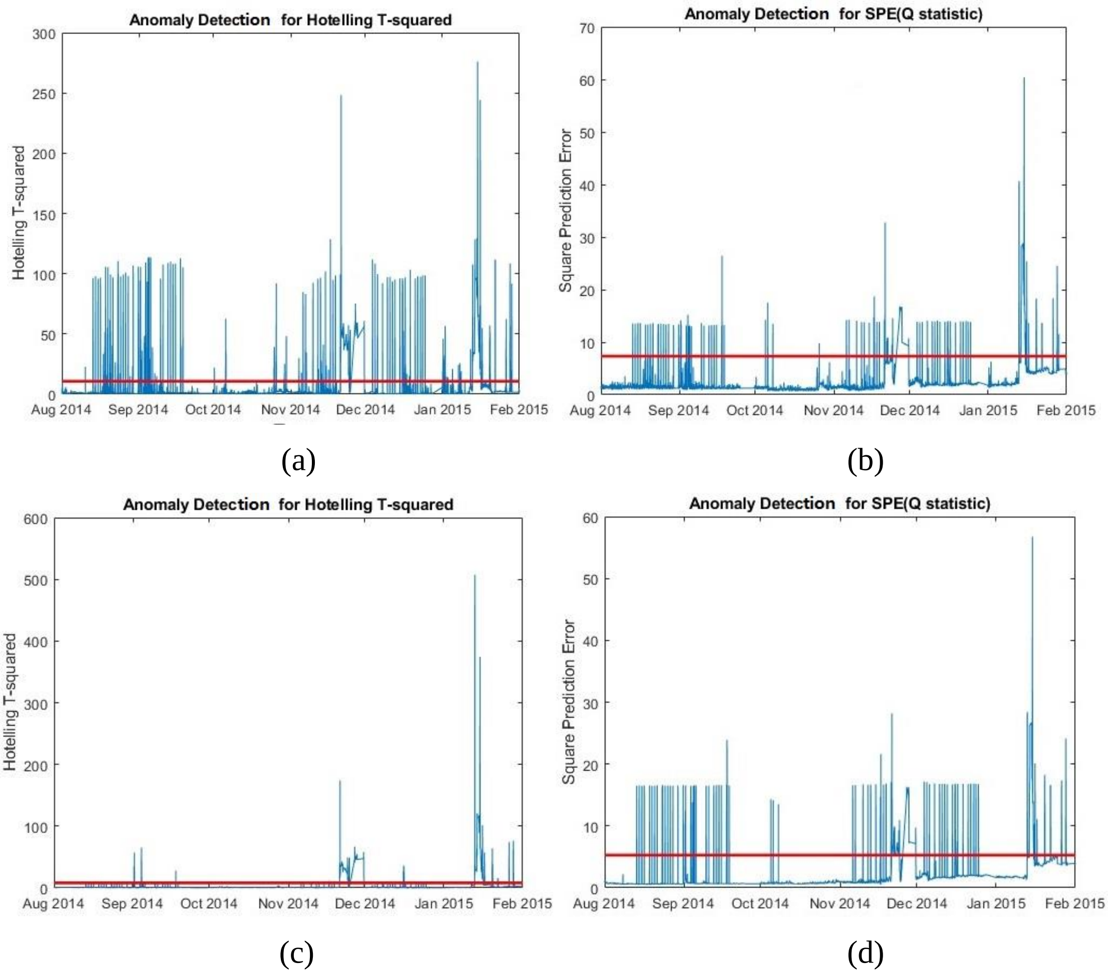


Fig. 6-12: Comparison of the PCA-based fault detection results. (a) and (b) show the results based on the T-squared and SPE thresholds before noisy feature removal. (c) and (d) show the results after noisy feature removal.

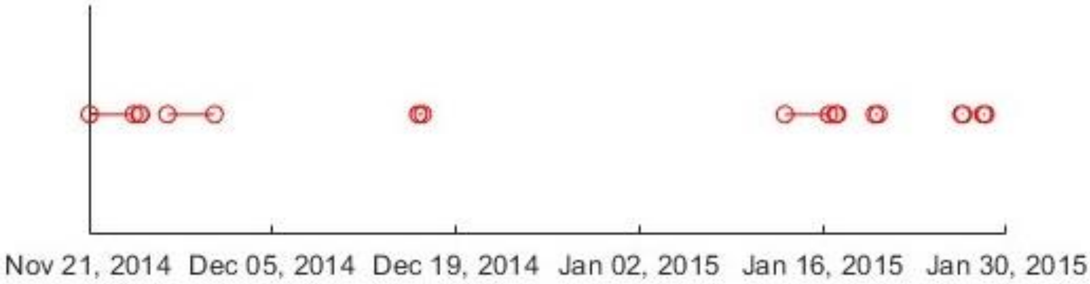
The results of the experiments which do not employ feature selection are confusing because alarms are triggered almost constantly. SPE cannot activate the alarms in a timely manner when the first large outage occurs during December. After removing the noisy features, it can be seen that for T-squared and SPE statistics, the number of false alarms is reduced significantly, and SPE is able to identify the December event accurately. Note that although detection has been improved greatly for both T-squared and SPE statistics, there are significant differences in the detection results.

Fault detection results based on T-squared:

During the testing period, the T-statistics triggers a total of 128 alarms, all of which are recorded, no matter how long they continue. After a 5-minute deterministic mechanism being implemented, only 10 alarms are left. The detection system triggers an alarm on 21 Nov 2014 at 01:33:03 for the first outage and continues for three days and temporarily stops on 24 Nov 2014 at 10:20:17. Then the T-indicator exceeds the threshold again after ten hours and continues for two hours. The third alarm is detected on 27 Nov 2014 at 00:00:00 and continues for three days. From Nov 21 to Nov 30, three alarms are identified, warning the system but with a short time interval. All three alarms could be reported as one large outage. The next alarm occurs on 16 Dec 2014 at 02:57:31, sending intermittent alerts which stop on 16 Dec 2014 at 11:20:54. There are another two outages from 13-Jan-2015 02:26:50 to 20-Jan-2015 05:09:47, and from 26-Jan-2015 13:47:23 to 28-Jan-2015 09:27:52. *Fig. 6-13(a)* gives a more intuitive visual display of the detected events where each alarm is represented by a line with two circles at the start and end time. Corresponding to the above T-squared alarm analysis, the figure shows that after consolidating ten alarm events, there were four main outages with the parameter time window set at more than three days.

Fault detection results based on SPE:

Compared to the T-squared detection results, SPE detects many more small outages than the T-squared statistics. SPE detects 1871 alarms but after filtering for 5 minutes, the number of alarms is reduced to 28. The alarm sequence is depicted in *Fig. 6-13(b)* and the start and end time for each alarm is summarized in *Table 6-3*. To determine what causes such a large difference between the results of SPE and T-squared, we check the monitoring features during those small outages which were only detected by SPE. It is found that most of them result from one feature (see *Fig. 6-14*). SPE is more sensitive and can capture every tiny abnormal pattern, even when it is present in only one of the features. This is because when SPE computes the reconstruction error, even if only one feature appears to have an abnormal value, its fault measurement could be smeared into all the other features/variables through the reconstruction process. Observing the feature in *Fig. 6-14*, these spikes down to zero do not act like chaotic changes, more likely indicating some potential issues. These features were correctly not identified as noisy features to be filtered out during feature selection. This scenario results in a further problem, this being whether we should assign more weight to the T-squared results or the SPE results as T-squared detects fault events from a global perspective but SPE concentrates more on the details. To address this, a comprehensive analysis report can be generated to provide the engineers with further insight.



(a) T-squared

(b) SPE

Fig. 6-13: Alarm sequences

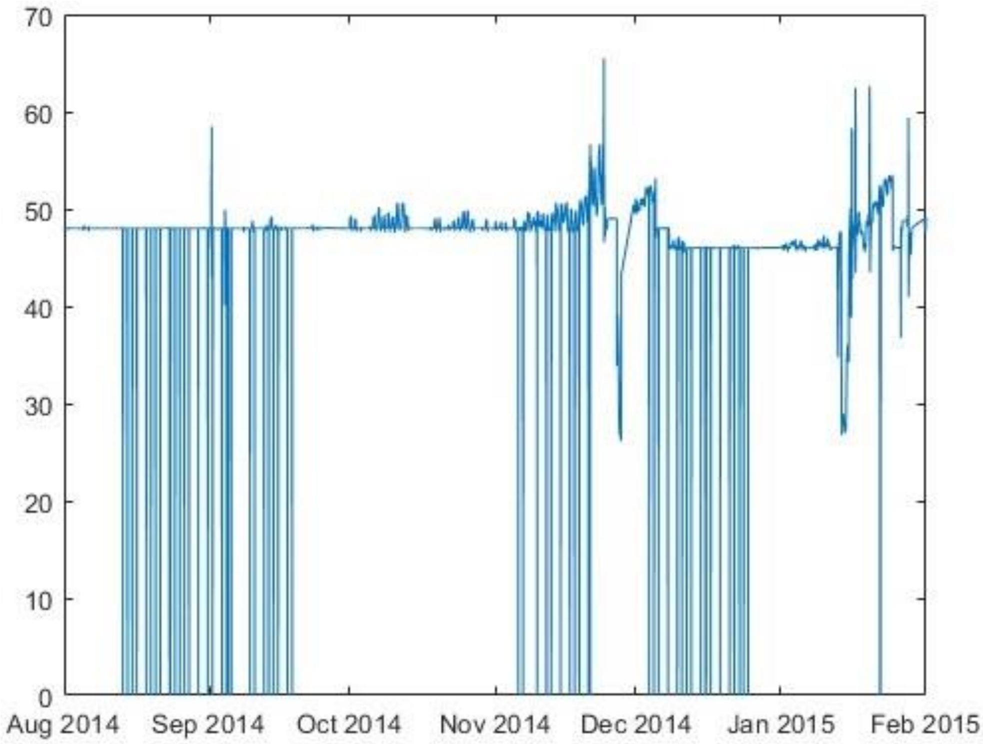


Fig. 6-14: The feature which results in lots of small outages

Table 6-3: Alarm logs

Alarm ID	Starting time	Ending time
1	16-Nov-2014 19:57:40	16-Nov-2014 21:42:49
2	16-Nov-2014 21:42:51	16-Nov-2014 21:48:01
3	16-Nov-2014 21:48:08	16-Nov-2014 22:13:55
4	16-Nov-2014 22:13:57	16-Nov-2014 22:21:34
5	16-Nov-2014 22:21:56	16-Nov-2014 22:59:40
6	16-Nov-2014 23:00:17	16-Nov-2014 23:19:38
7	16-Nov-2014 23:19:40	17-Nov-2014 00:16:11
8	17-Nov-2014 00:20:44	17-Nov-2014 00:26:23
9	21-Nov-2014 05:39:25	22-Nov-2014 10:20:38
10	22-Nov-2014 10:55:20	22-Nov-2014 11:00:32
11	22-Nov-2014 22:05:58	24-Nov-2014 05:50:47
12	24-Nov-2014 06:22:36	24-Nov-2014 06:46:05
13	27-Nov-2014 00:00:00	30-Nov-2014 14:00:00
14	16-Dec-2014 02:57:39	16-Dec-2014 09:03:30
15	13-Jan-2015 03:15:34	13-Jan-2015 08:19:22
16	13-Jan-2015 21:40:33	15-Jan-2015 16:08:29
17	16-Jan-2015 00:59:41	16-Jan-2015 01:07:26
18	16-Jan-2015 01:21:47	16-Jan-2015 01:27:46
19	16-Jan-2015 04:09:05	16-Jan-2015 06:35:52
20	16-Jan-2015 07:25:42	16-Jan-2015 07:34:16
21	16-Jan-2015 23:14:15	17-Jan-2015 00:03:36
22	17-Jan-2015 00:05:58	17-Jan-2015 00:29:11
23	19-Jan-2015 23:08:02	20-Jan-2015 01:54:00
24	20-Jan-2015 02:38:17	20-Jan-2015 03:02:43
25	26-Jan-2015 14:40:03	26-Jan-2015 16:00:30
26	28-Jan-2015 05:25:05	28-Jan-2015 05:33:10
27	28-Jan-2015 06:06:55	28-Jan-2015 06:15:36
28	28-Jan-2015 07:13:59	28-Jan-2015 07:31:11

Although the SPE detection results show that a lot of small outages have been identified, contribution analysis indicates that most come from the same event ID. These small outages can be compressed into several large ones using alarm association rules. These alarm logs can be written into the comprehensive report as additional information for the engineers working in the control room. By providing them with additional information, they can operate more effectively and check what actually happens in the compressor in a timely fashion.

Contribution analysis:

As previously discussed, contribution analysis is applied to each detected anomaly to check whether they are caused by certain features or the joint effect of all features. T-squared statistics does not detect one particular feature which predominantly contributes to the occurrence of an anomaly. The final analysis report shows that there were four outages in one alarm sequence on the following days: 21 Nov 2014 01:33:03, 16 Dec 2014 02:57:31, 13-Jan-2015 02:26:50, and 26-Jan-2015 13:47:23.

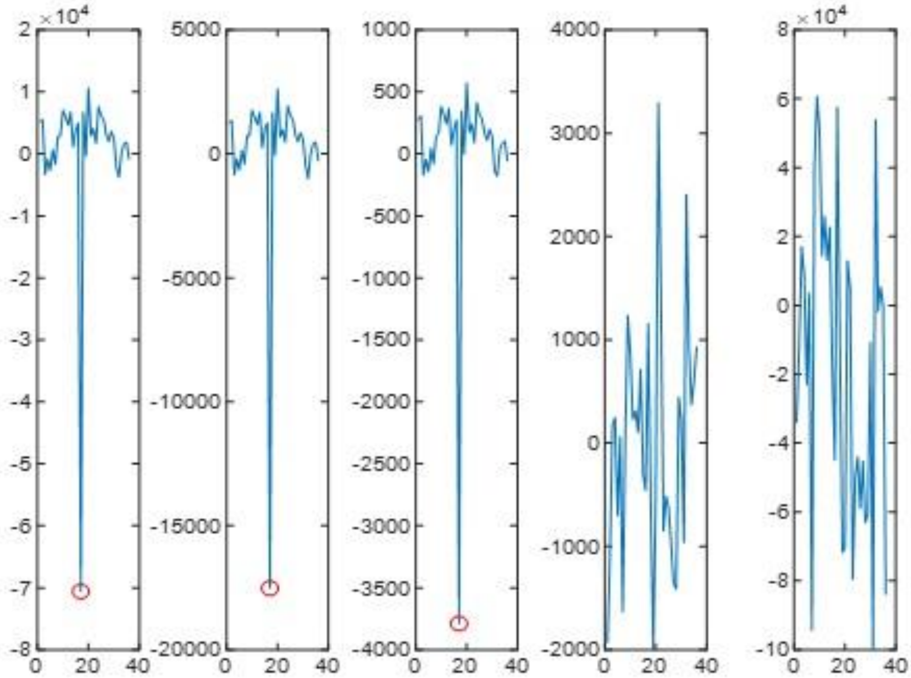
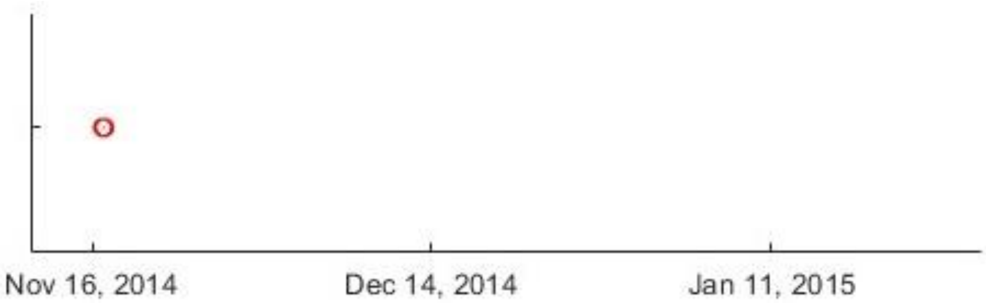


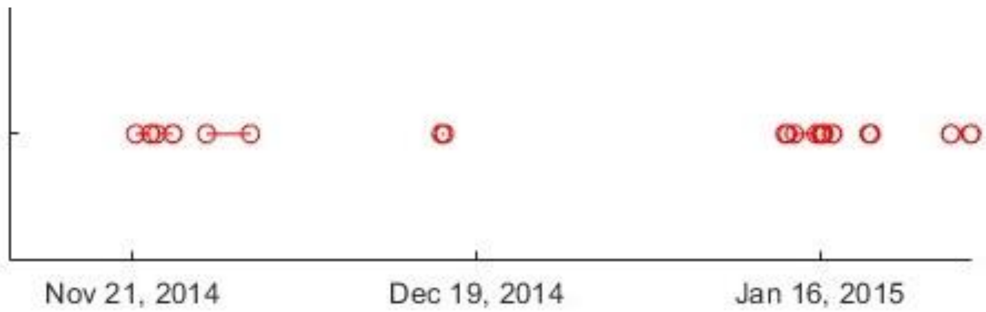
Fig. 6-15: Contribution plot of several time periods when detecting anomalies

However, the results for SPE show that it detected that eight of the 28 alarms are caused by the same feature. *Fig. 6-15* shows several examples and each subfigure illustrates the contribution (reconstruction error) of all the features for one detected alarm. The three figures on the left indicate that the feature circled in red has the highest SPE value and should be identified as the root cause. The two figures on the right indicate that every feature contributes almost equally. Then the alarm sequence can be separated into two sequences. As shown in *Fig. 6-16*, the top one consists of the alarms caused by only one dominating feature, and the bottom one consists of the remaining alarms which result from many features. From the top sequence, one outage from 16-Nov-2014 19:57:40 to 17-Nov-

2014 00:26:23 is detected after alarm consolidation according to the association rules and is kept for additional reference information for the engineers. Four main outages are analysed from the bottom alarm sequence, showing that alarms were triggered on 21-Nov-2014 05:39:25, 16-Dec-2014 02:57:39, 13-Jan-2015 03:15:34, and 26-Jan-2015 14:40:03 respectively. Compared with the SPE results, the T-squared detection results show that it is able to warn the system hours earlier.



(a) Alarms resulting from one dominated feature



(b) Alarms reflected from most of the features

Fig. 6-16: Separate alarm sequences

6.2.2 Noise Detection

Noise detection ensures the PCA model has clean training data without noise. Theoretically, after noisy data cleaning, the standard deviation is smaller, which indicates the training data is more compact and clustered compared to the data before cleaning. Due to the clean training data, the model is closer to the normal condition and more able to reflect the in-control condition. The T-squared and SPE statistics detection results based on the clean training data are displayed in *Fig. 6-17*, and the detailed alarm records with the start and end time are listed in *Table 6-4* and *Table 6-5*. *Table 6-6* summarizes the trained PCA model based on the training data before and after cleaning. Even though the data we use in this case study are quite clean, and the noise removed by the algorithm only represents 0.39% of the data, the trained model after cleaning changes greatly.

Table 6-4: Detected alarm logs using T-squared statistics

Alarm ID	Starting time	Ending time
1	16-Nov-2014 13:16:48	16-Nov-2014 13:25:23
2	16-Nov-2014 13:38:01	16-Nov-2014 13:47:02
3	16-Nov-2014 13:50:37	16-Nov-2014 14:03:11
4	16-Nov-2014 14:09:51	16-Nov-2014 15:01:42
5	16-Nov-2014 15:07:11	16-Nov-2014 15:29:54
6	16-Nov-2014 15:33:22	16-Nov-2014 15:41:04
7	16-Nov-2014 15:41:27	16-Nov-2014 16:18:37
8	16-Nov-2014 16:21:47	16-Nov-2014 16:38:15
9	16-Nov-2014 16:38:51	16-Nov-2014 17:22:33
10	20-Nov-2014 18:55:14	24-Nov-2014 03:15:43
11	24-Nov-2014 14:20:36	24-Nov-2014 16:08:08
12	24-Nov-2014 17:19:42	27-Nov-2014 17:15:16
13	15-Dec-2014 20:56:08	16-Dec-2014 03:44:40
14	12-Jan-2015 20:11:16	16-Jan-2015 04:55:29
15	16-Jan-2015 04:57:36	16-Jan-2015 05:05:30
16	16-Jan-2015 16:29:24	16-Jan-2015 19:44:15
17	19-Jan-2015 16:21:15	19-Jan-2015 22:28:02
18	26-Jan-2015 07:07:53	26-Jan-2015 08:58:31
19	26-Jan-2015 08:58:47	26-Jan-2015 09:51:46
20	26-Jan-2015 22:35:17	28-Jan-2015 02:44:59

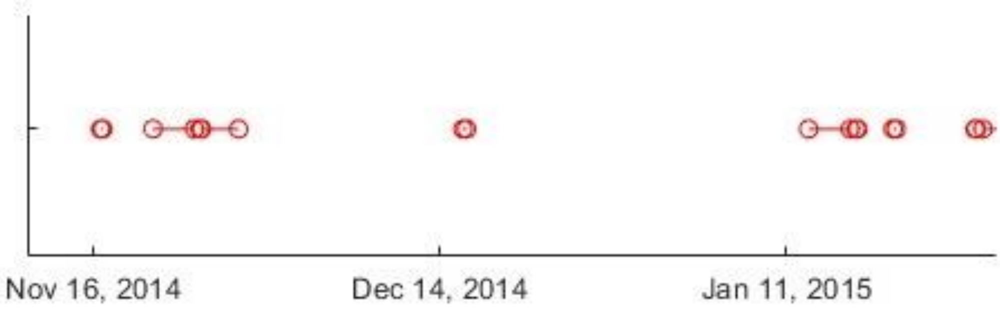
After data cleaning, the thresholds for both T-squared and SPE increase. The total explained variance decreases a little, even with one more principle component, thus the new model acts more robustly. After alarm sequence analysis, T-statistics successfully detects the outage on the 16 Nov 2014 which is not detected using unclean training data. In total, it captures five main outages on 16-Nov-2014 13:16:48, 20-Nov-2014 18:55:14, 15-Dec-2014 20:56:08, 12-Jan-2015 20:11:16, and 26-Jan-2015 07:07:53 and in some cases, days earlier than the results based on the unclean training data, which is critical in the predictive maintenance framework, enabling it to provide more time for better decision making before the machine shuts down. SPE obtains similar results to T-squared statistics, where five outages occur, warning the system at 16-Nov-2014 13:16:48, 20-Nov-2014 22:58:37, 15-Dec-2014 20:13:19, 12-Jan- 2015 20:13:49, and 26-Jan-2015 08:46:54, which is earlier than the noise training data. These better results confirm the theoretical expectations that the clean data-based model is closer to the normal condition with less tolerance for abnormal patterns. When an anomaly appears, the reconstruction error becomes larger and exceeds the threshold earlier.

Table 6-5: Detected alarm logs using SPE statistics

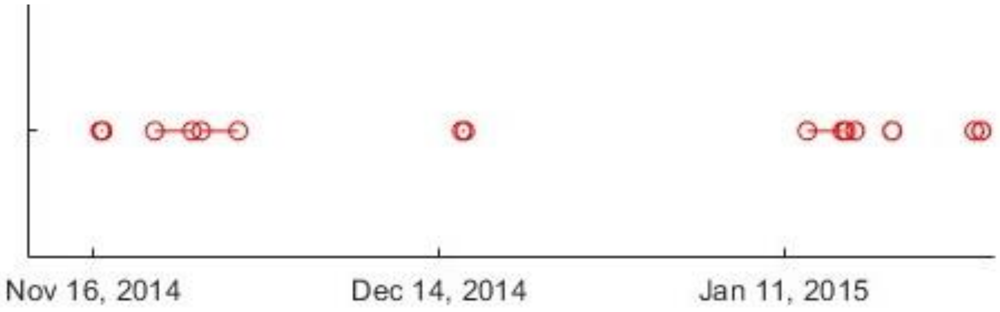
Alarm ID	Starting time	Ending time
1	16-Nov-2014 13:16:48	16-Nov-2014 15:01:42
2	16-Nov-2014 15:01:44	16-Nov-2014 15:06:57
3	16-Nov-2014 15:07:04	16-Nov-2014 15:33:20
4	16-Nov-2014 15:33:22	16-Nov-2014 15:41:04
5	16-Nov-2014 15:41:27	16-Nov-2014 16:18:37
6	16-Nov-2014 16:19:16	16-Nov-2014 16:38:15
7	16-Nov-2014 16:38:17	16-Nov-2014 17:33:58
8	16-Nov-2014 17:38:33	16-Nov-2014 17:44:12
9	20-Nov-2014 22:58:37	23-Nov-2014 23:08:39
10	24-Nov-2014 17:19:42	27-Nov-2014 17:15:16
11	15-Dec-2014 20:13:19	16-Dec-2014 02:17:56
12	12-Jan-2015 20:13:49	15-Jan-2015 16:49:02
13	15-Jan-2015 16:49:16	15-Jan-2015 17:09:59
14	15-Jan-2015 17:10:13	15-Jan-2015 17:26:04
15	15-Jan-2015 17:26:08	15-Jan-2015 17:37:49
16	15-Jan-2015 23:27:42	15-Jan-2015 23:32:50
17	16-Jan-2015 16:32:25	16-Jan-2015 17:11:17
18	19-Jan-2015 16:27:45	19-Jan-2015 16:41:26
19	19-Jan-2015 17:56:04	19-Jan-2015 18:06:03
20	26-Jan-2015 08:46:54	26-Jan-2015 09:02:47
21	26-Jan-2015 22:38:31	26-Jan-2015 22:51:36
22	26-Jan-2015 23:27:49	26-Jan-2015 23:36:22

Table 6-6: Summary of model parameters before and after noisy data cleaning

Data	Exp_variance	Num_comp	T^2	SPE
Before cleaning	0.9001	5	8.1152	5.2798
After cleaning	0.8923	6	9.4461	5.5875



(a) T-squared



(b) SPE

Fig. 6-17: Alarm sequences based on the clean data

6.3 Conclusion

This chapter evaluates the proposed IoT-based big data ecosystem. We utilized two different case studies utilizing real data from a manufacturing plant at IPL to carry out the evaluation. The experiment results demonstrate the effectiveness of both the architectural framework and techniques proposed in this thesis for fault detection and diagnosis in predictive maintenance, which is capable of alarming the system in a timely fashion, several days before the fault happens. In addition, the feature selection and noise detection techniques also show the ability to subsequently improve the performance of the fault detection model, which indicates that data preprocessing is critical to enhance the quality of the source data.

Chapter 7

Optimized Architectural Framework

This chapter presents an optimized architectural framework by the embedding edge computing technique and distributed stacked sparse autoencoder algorithm to further improve the robustness of our ecosystem proposed in Chapter 5 leading to an advanced framework for IIoT.

7.1 Introduction

Most proposed IoT application architectures constantly gather and process all data in a centralized cloud including the architecture proposed in Chapter 5 of this thesis. The benefits of integrating cloud computing into the IIoT are self-evident. However, cloud computing involves transferring data to a cloud server for analysis and returning the results back to the users, which is more likely to cause high latency and mobility-related issues [292,293]. The current cloud-based IoT systems suffer latency and cost issues for industrial manufacturing, as sensor data are collected, stored and processed in the cloud and then sent back to the applications. Industrial manufacturing urgently requires a real-time intelligent system especially with low latency to be able to respond to alarms in a timely manner. In addition, when it comes to human safety scenarios, latency becomes a more critical issue and the analysis model must be performed in strict real-time. Furthermore, many manufacturing plants are located in remote areas without a proper and stable network bandwidth, which will increase the processing time and increase the latency between the data sources and the centralized cloud cluster.

Edge computing can be the solution to this unresolved urgent challenge. In contrast to cloud computing, edge computing refers to decentralized data processing at the edge of the network, thus reducing the heavy load on networks and offloading the processing tasks from the cloud. Intuitively, it is more efficient to compute the data close to where it is produced instead of struggling with the transfer and then processing the data in a faraway cluster. Additionally, edge computing also brings extra advantages,

- Edge computing relieves network pressure as there is less data to be transferred, it allows for low-latency data processing and gives the monitoring system the ability to make (near) real-time, autonomous alarm decisions.
- The data owned by private industries are mostly confidential and data security has always been a hot topic in IIoT. As reported in [294], research organizations estimate that over 90% of data are stored and processed locally. The edge server is able to process sensitive private data without the need to transmit them to the remote cloud, and it is easier to better protect user privacy by processing and storing data where it belongs, at the edge of the network [295].
- Pushing, storing, and processing all data in the cloud is associated with high cloud costs, and cloud costs and resources can be optimized by enabling edge devices with a computing ability.
- Most data stored in the cloud are of no value to the company and are never used. After aggregating and filtering raw data locally to suppress superfluous noise and remove unnecessary data, only the intermediate processed data will be transferred to the cloud, saving significant cloud space, minimizing cost and increasing the quality of the data sent to the cloud.

7.2 Review of Related Work on Edge Computing

Edge computing is performed near the IoT device layer with a limited computing capability but provides fast processing and quick response services. Characterized by these properties, edge computing has been leveraged for applications with special requirements focusing on minimizing latency, obtaining real-time data insights, maximizing privacy, strengthening security, optimizing resource utilization, etc. [296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318]. Wang et al. [319] proposed an edge-based IoT-cloud architecture with a trust evaluation mechanism to address security issues. The architecture that is integrated with a balanced dynamic based on cloud and edge computing is designed to improve the efficiency of IoT services, reducing resource consumption and network delay. In the work in [320], Wang et al. proposed a trust evaluation method for efficient data collection based on edge computing, and the obtained node trust value is utilized to design a trustworthy data collection path of the mobile fog nodes.

As edge computing and cloud computing are functionally complementary in the aspects of storage and computing capabilities, recent studies focus on jointly deploying cloud computing and edge computing in IoT applications. An IIoT data processing framework based on both the fog and cloud computing is proposed in [321], integrating data preprocessing, storage and retrieval.

Typically, AI analytic models need to be compressed or reduced on edge servers, in which case the network traffic can be greatly reduced. But in turn, the degree to which the models should be distributed into edge servers is limited by its constricted computing power, which is a common issue in most application scenarios. With the limited computation resources in edge servers and the low bandwidth of the network in IoT systems, most related works aim at finding the best trade-off between them. Li et al. [322] introduced deep learning in IIoT under the edge computing environment. In their proposed architecture, the deep learning networks are first trained in the cloud server and task offloading only happens in the testing phase, where the pretrained networks are divided into two parts. The lower layers are deployed at the edge servers close to the input data while the higher layers are arranged at the cloud. They designed a scheduling algorithm to solve the problem of maximizing the number of deep learning tasks with the limited network bandwidth and service capability of the edge nodes. Xu et al. [323] proposed a computation offloading method by formulating the problem into a multiple objective programming problem and solving the optimization problems using NSGA-III (non-dominate sorting genetic algorithm III), aiming at achieving trade-offs between optimizing the execution time and the energy consumption of the mobile devices in the cloud-edge computing environment.

However, when the training phase of some machine learning algorithms, especially deep learning, is also conducted on both edge servers and the cloud, it imposes a heavy burden on both the computational capability of edge servers and the communication resources in the network since it requires the model parameters to be iteratively updated using a backpropagation learning algorithm. Some researchers try to reshape the learning process to find a trade-off between computation, communication and accuracy. With the accuracy factor being introduced into the offloading tasks, the strategy needs to consider more, including the delay and accuracy requirements of a certain machine learning task, the computing capability of edge servers, and the network topology.

In the work in [324], the authors address the problem of how to efficiently utilize the limited computation and communication resources at the edge for high-accuracy model learning. A gradient-descent-based distributed learning algorithm that includes local update and global aggregation steps is designed and utilized to train machine learning models with the support of both edge nodes and the cloud. Based on this learning algorithm, a control algorithm is proposed to determine the trade-off between the frequency of local updates and global aggregation to minimize the loss function to achieve an optimal learning performance under a given resource budget. The work in [325] proposes an optimal offloading framework for accuracy maximization offloading with latency constraints. The work in [326] formulates a computation offloading game to model the competition between IoT users and allocate the limited processing power of fog nodes to IoT users in a hierarchical computing paradigm including fog and remote cloud computing services. Lai et al. [327] install the edge servers near different data sources to execute decentralized data preprocessing from the cloud, including appliance selection, data purging, and feature selection. Sun et al. [328] propose an efficient IoT architecture, edgeIoT, by leveraging fog computing and software-defined networking to analyse the IoT data streams at the mobile edge. However, they only discuss the logic of the collaboration between edge computing and the involved machine learning techniques, without special attention being given to deployment in real industrial manufacturing.

7.3 Contribution

Predictive maintenance automates the detection of potential equipment failures to maximize machine parts and equipment uptime and deploy maintenance more cost effectively. The IIoT-based predictive maintenance that involves real-time monitoring requires fast processing and quick response services. In a smart manufacturing factory, edge devices can be installed close to the sensors for collecting and analysing raw IIoT data to evaluate the equipment running condition locally, rather than directly pushing them to the cloud. Therefore, once an alarm condition is reached, an actuator can be triggered immediately, which enables engineers to react more quickly. When the real-time monitoring requirement is guaranteed, the processed data can be transferred to the centralized cloud at a slower pace with less restrictions in relation to timeliness.

Based on these considerations, this chapter aims to significantly reduce the detection system response time and consistently achieve real-time processing by providing a complete solution for predictive maintenance ecosystem embedding with edge computing in IIoT-based smart manufacturing. Thus, in this chapter, we introduce edge computing into the IoT-based fault detection ecosystem in manufacturing, effectively precluding the time delay resulting from the heavy transfer of mass data and potentially speeding up the IoT big data process. This provides more possibilities of achieving a real-time processing ability. Instead of focusing on elaborating the optimized computing of machine learning methods when incorporated with edge computing [327, 322], we introduce an integrated architecture, which demonstrates the overall working principle of edge computing facilitating fault detection in an IoT paradigm. The seamless integration between the edge computing layer and the currently proposed cloud computing-based IoT ecosystem is also a critical issue in deployment in real industrial settings. The edge computing layer needs to be embedded properly to be compatible with other heterogeneous layers. For practical purposes, we also provide application programming interface (API)-oriented implementation guidelines. To the best of our knowledge, this is innovative work focusing on fault detection for IoT-enabled smart manufacturing with edge computing. Specifically,

- We propose an optimized IoT-based big data ecosystem embedded with edge computing for real-time fault detection processing. It is integrated, involving the whole process of the data flow from the data source through the edge layer and cloud layer, then reaching the application layer for service-specific tasks. This is a completely new concept of the IoT-based framework, hence there are major transitions in the working principles of data ingestion, preprocessing and analytics due to the incorporation of edge computing.
- Unlike previous research which is limited to the presentation of the edge computing structure, we additionally introduce the proposed optimized ecosystem through a new perspective. Behind the ecosystem structure, various advanced software techniques are presented and utilized that enable it to be successfully launched and deployed in industrial settings. This is the first work from the business application perspective, detailing the implementations of edge computing to assist the IoT to store and process increasing growing scalable big data.
- A distributed stacked sparse autoencoder model is proposed for fault detection in predictive maintenance on the Apache Spark Platform in the edge layer of our optimized IoT-based big data ecosystem, addressing the complicated non-linear relationships between multiple sensors. It is an upgraded version of DPCA, which is only feasible when the sensors have linear correlations. The prediction model is built and trained in the cloud and the pre-trained model is deployed and located in the edge layer, running on-premises. If an issue is detected, IoT edge triggers an alert and processes the data locally with a high priority and sends it to the cloud for further analysis.

7.4 Real-time Framework for IoT Fault Analytics: Architecture, Implementation and Techniques

As edge computing can bring various benefits to an IIoT-based intelligent manufacturing system, an optimized big data analytics system is highly desirable for global manufacturing industries. In this chapter, we propose an optimized architecture for embedding edge computing into a big data ecosystem, which involves big data ingestion, processing, storage and analytics in a real-time environment. The proposed architecture consists of three main layers as depicted in *Fig. 7-1*, namely the edge layer, the cloud layer and the application layer. Most of the computation is performed on the edge layer instead of the cloud layer by moving certain workloads to the edge cluster.

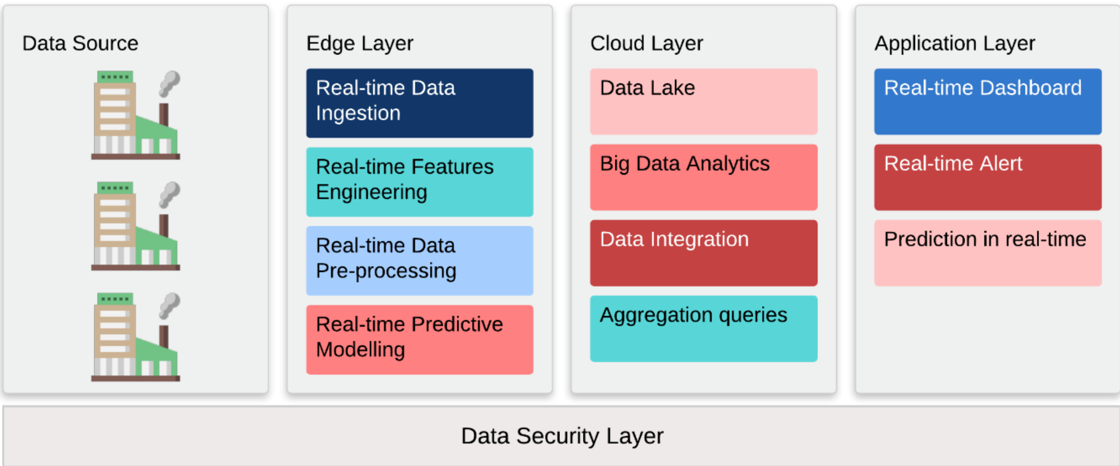


Fig. 7-1: The optimized architecture of the big data ecosystem in the Industrial Internet of Things

7.4.1 Optimized Architecture of the Proposed Framework

Compared to previous solutions [329] and other IoT data analytics architectures, our optimized system completely offloads the real-time equipment fault detection functions from the cloud. Therefore, the proposed architecture not only preserves the original advantages of edge computing, which improves the computing power near the IoT devices, it also decreases the latency issue when the cloud is involved in the process of detection. As shown in *Fig. 7-2*, there are three types of data analytic frameworks. Without the incorporation of edge computing, all the sensor data is transferred into the cloud for its powerful computation, resulting in heavy network traffic. The communication performance will be the bottleneck with improved processing efficiency. Gradually, some studies started employing edge computing to alleviate traffic issues. In the edge computing environment, raw data are sent to the edge layer first, and only the preprocessed data need to be transferred to the central cloud cluster. In this case, the pressure on the network is reduced due to less data being transferred. However, all the analytic results associated with the cloud server need to be send back to the IoT-related factory, providing the control room operators with technical support for smart decision making. The remote cloud architectures suffer from high response time and scalability issues.

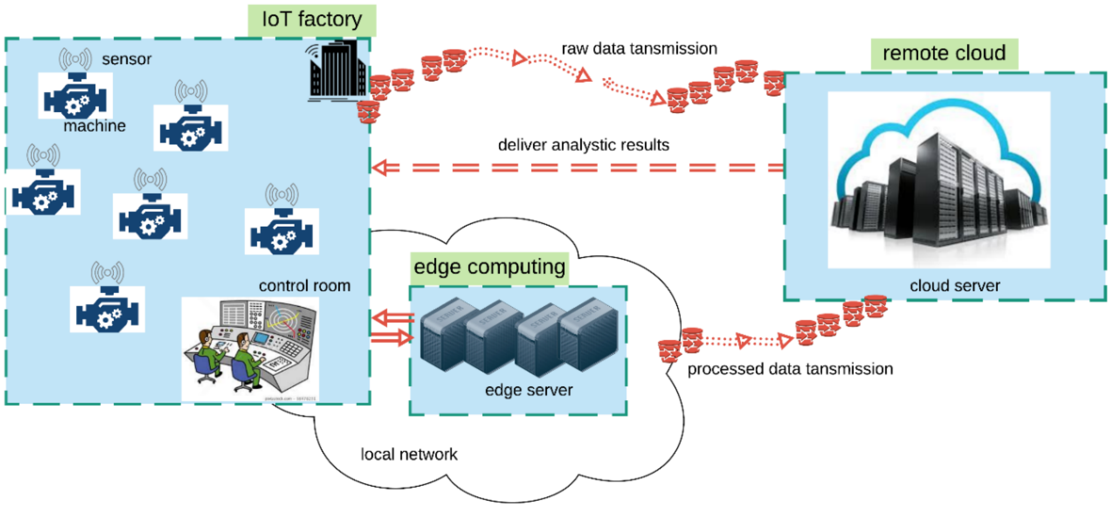


Fig. 7-2: Types of data analytic frameworks

Data analytics which is involved in remote cloud computing unavoidably leads to a high response time, but this might not be tolerated by delay-sensitive applications and even real-time applications, such as fault detection in manufacturing equipment monitoring. To address the delayed response problem, our proposed architecture implements the whole process of fault detection at the edge layer, while big data management and data analytics are conducted on the cloud layer. This short-range communication makes real-time services feasible due to its short transmission distance and powerful local area network.

7.4.2 Edge Layer

In our architecture, the edge layer is responsible for big data collection, features engineering, data pre-processing and predictive analytics to improve system response time, reduce operational costs and increase data security. *Fig. 7-3* demonstrates the architecture of the edge cluster in our framework from a high-level perspective, which provides more detailed guidelines for framework implementation and all the involved API and techniques are clearly labelled along the process line.

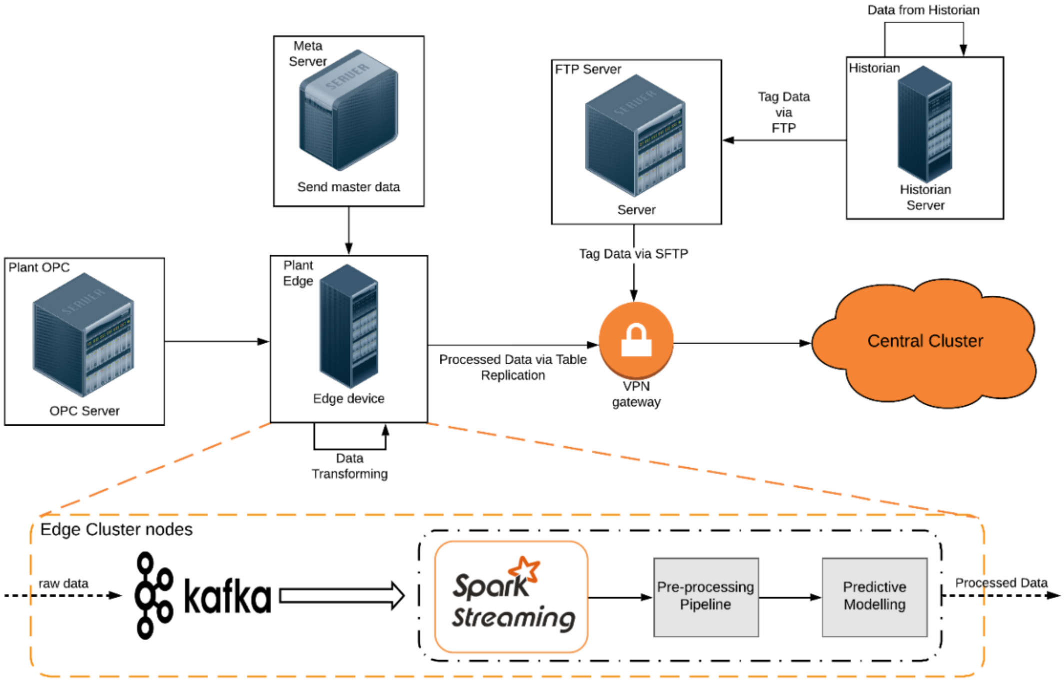


Fig. 7-3: Details in the single edge cluster for the API-oriented edge layer

---

Algorithm 4 Ingest the sensor data into the edge layer

Environment1: Deploy OPC Servers, Meta Servers, OPC Collector and edge devices;

Environment2: LAN/WAN network configuration.

Step1: Retrieve metadata of sensor tags that users want to collect;

Step2: Match and monitor the collected tags from OPC Server to generate new tag value data by: if collected sensor names in Metadata Server exist in OPC Servers;

Step3: Write the tag values data into a text file;

Step4: Publish the text file as a message $M_{key,value}$ into a particular topic to Kafka Cluster.

Our system stores the measurement logs with structured data in the metadata server, the numerical sensor data of all the equipment in the OPC servers in real time and the historical sensor data in the historical servers in case they need to be extracted in future. Algorithm 4 shows the proposed big data ingestion method. Firstly, the OPC Collector, which is a platform-based Windows Service Application and is installed on the edge device, retrieves the metadata of sensor tags that will be collected from a database, and then monitors these tags from the OPC server. All the tag values received from the OPC server are written in a text file. Then, each text file is published as a message to a particular topic and saved into the Kafka Cluster. Apache Kafka is a distributed messaging system that is used for publishing and subscribing to a stream of records. Compared to traditional messaging systems, it provides a high-throughput with partitions, it is fault-tolerant with replication, it has a low latency and horizontally scalable platform, which has been used in production in thousands of companies.

In the edge layer, Apache Kafka is applied as a streaming data platform and Apache Spark Structured Streaming is applied as a streaming processing platform. Spark Structured Streaming is a scalable fault-tolerant streaming processing engine and is a more optimized streaming platform in comparison to Spark Streaming. Our system integrates Apache Spark with Kafka using the Spark Structured Streaming API to consume the sensor data from Kafka. The message is forward by the Kafka broker to Spark Structured Streaming, which accepts data in real-time and processes the data parallel on a cluster. The workflow of our streaming data pipelines includes data transformation, data cleaning and predictive modelling. Firstly, all raw data are transformed into the DataFrame format which is a distributed collection of data into named columns to define the schema in advance. Then the new structured data will be filtered by the noise filter node proposed in our paper [289] to remove undesirable sensor noise and improve data quality. Finally, a deployed predictive model will use the pre-processed data as input to detect whether the equipment is running in normal conditions.

7.4.3 Predictive Model

A stacked autoencoder as a type of unsupervised deep learning is introduced as the key technique in our edge layer due to complicated sensor nonlinear relationships. Using the nonlinear map function, a stacked autoencoder can typically extract better features than linear transformation methods such as PCA and the correlations between sensors can be well described and formulated. The stacked autoencoder attempts to approximate a map function in an unsupervised manner by minimizing the reconstruction errors between inputs and outputs.

A stacked sparse autoencoder is a variant of the traditional autoencoder that improves robustness against noisy perturbations during the training stage. To obtain robust encoded features, the sparse penalty term is introduced in the hidden layer to control the number of active neurons to ensure the neurons are mostly inactive. Specifically, for the input samples $x \in R^n$, the weight matrix and bias vector in the q-th layer are $W_{i,j}^{(q)}$ and $b_i^{(q)}$. The cost function of the stacked sparse autoencoder can be expressed as,

$$\hat{C} = \sum_{x \in X_n} \left[C\left(x, W_{i,j}^{(q)}, b_i^{(q)}\right) + \lambda P\left(x, W_{i,j}^{(q)}, b_i^{(q)}\right) \right].$$

In the cost function, the first term is the cost function of the traditional autoencoder, while the second is a sparsity constraint term. As shown in *Table 7-1*, there are several ways to add the constraint as we can employ different types of sparsity on different parts of autoencoder networks. ρ is the sparsity parameter in the Kullback-Leibler divergence, which is the target average activation of hidden units, this generally being a small value nearing zero. $\hat{\rho}_j$ is the average of h_j which is the activation of hidden node j.

Table 7-1: Different ways to introduce sparsity

sparsity methods	sparsity parts
$\|\cdot\|_1$	$W_{i,j}^{(q)}$
$\|\cdot\|_2$	$b_i^{(q)}$
$\sum_{j=1} KL(\rho\|\hat{\rho}_j)$ $KL(\rho\|\hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho)\frac{1-\rho}{1-\hat{\rho}_j}$	h_j

Once the model is constructed, without the concern of delay issues, the offline training stage can proceed on the cloud in order to benefit from strong computation power. For real-time fault detection in the process of monitoring, the model is deployed in the edge layer under the support of the Apache Spark processing engine in a distributed manner as depicted in *Fig. 7-4*.

Initially, we distribute the parameters of each layer in separate work nodes. The work node1 implements the first layer of the network with parameter $W_{1,1}^{(1)}, \cdots, W_{n,m}^{(1)}, b_n^{(1)}, \cdots, b_n^{(1)}$ and transforms the raw data, which is monitoring data in our case, into a vector of activation values $H^{(1)} = \left[h_1^{(1)}, h_2^{(1)}, \cdots, h_m^{(1)}\right]$, where m is the number of hidden nodes in the first layer. Then H is used as the input for the second layer and is performed in work node2 with parameters $W_{1,1}^{(2)}, \cdots, W_{m,k}^{(2)}, b_m^{(2)}, \cdots, b_n^{(1)}$, delivering the output of the second layer $H^{(2)}$, where k is the number of hidden nodes of the second layer. For the subsequent layers, the same strategy is applied and the same number of work nodes are launched as the number of layers. The various work nodes cooperate with each other by communicating with the Spark driver (master node). When the monitoring data is arriving continuously, even if the last data stream has not been processed completely, as long as the task in work node1 is finished, the edge layer can start processing the newly arriving data. In this way, all the nodes are actually working in a parallel way. Algorithm 5 summarizes the parallel processing of the stacked sparse autoencoder.

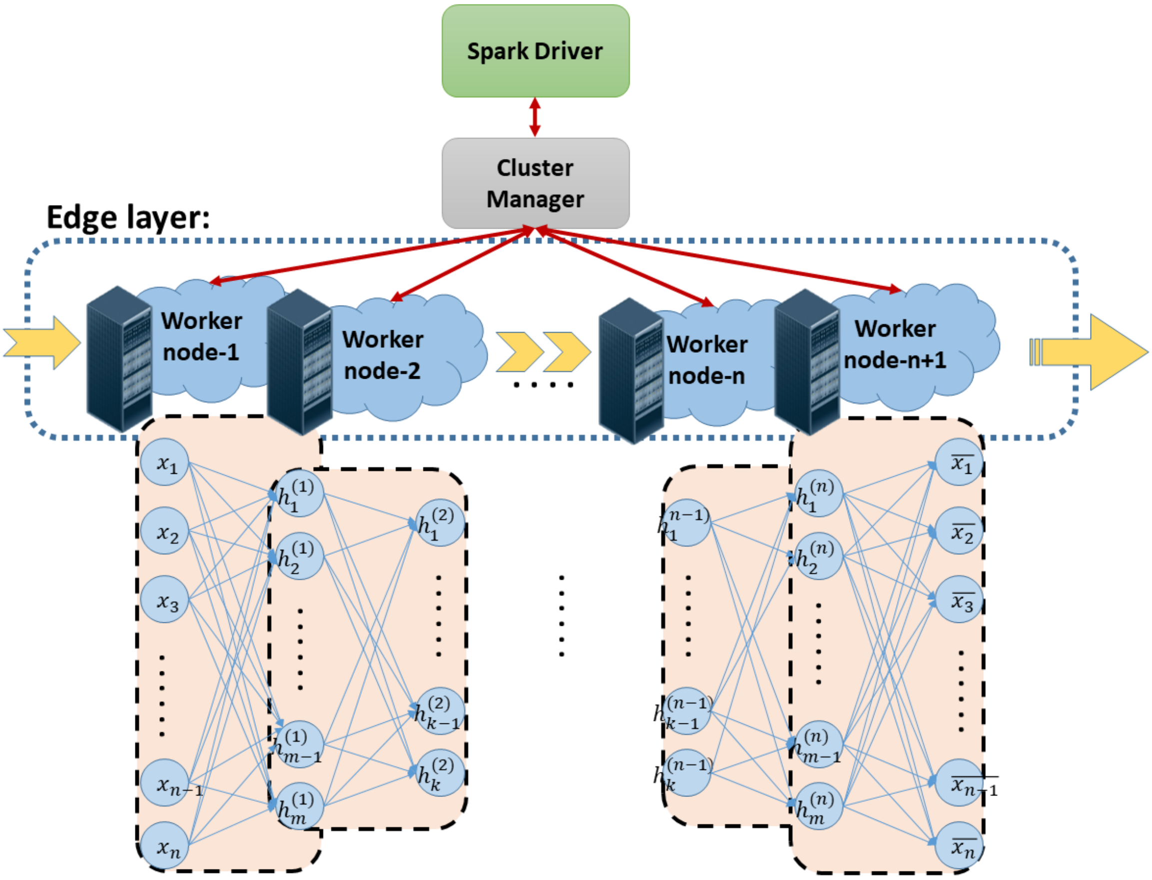


Fig. 7-4: Distributed stacked sparse autoencoder on Apache Spark

---

Algorithm 5 parallel processing of stacked sparse autoencoder

---

1. Distribute the parameters of different layers into different work nodes;
2. **for** each node (q): **do**
3. Compute the output values $H^{(q)}$ of hidden nodes in autoencoder;
4. Send the results to the Spark driver;
5. Driver node broadcasts the results from the current node to the next node as inputs and updates the autoencoder processing.
6. **end for**

7.4.4 Cloud Layer

The predicted results generated by the predictive model are delivered to the control room for monitoring the equipment health state by local engineers. In addition, all the processed data and the predicted results are not only monitored in the control room by local engineers, they are also sent to the boundary node of the AWS cluster through the Secure File Transfer Protocol (SFTP) to provide reliable connection-oriented data transfer over the Transmission Control Protocol (TCP) with the capability to resume in the face of a potentially unstable or slow network connection. *Fig. 7-5* depicts in detail the API-oriented cloud and application layer and how the procedures are implemented for deployment in actual production. The boundary node is the interface between the Hadoop cluster and the outside network. It is used to run applications and cluster administration techniques in our system. Moreover, a VPN gateway is set up between the boundary node and multiple edge clusters to secure the entire network.

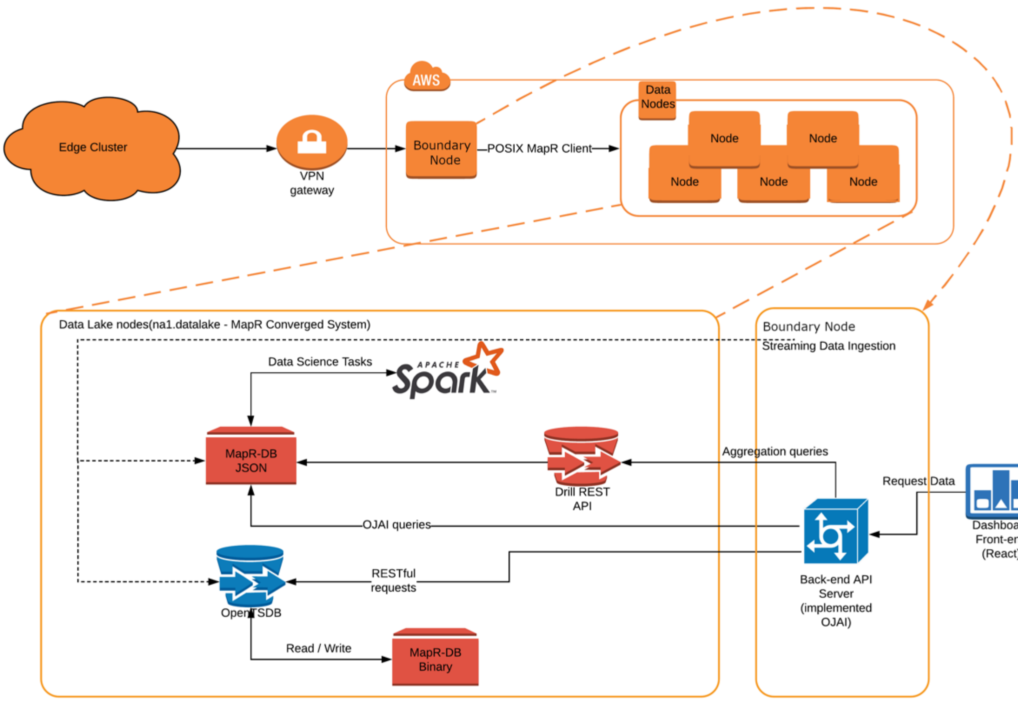


Fig. 7-5: Illustration of the API-oriented cloud and application layer

The streaming processed data and predicted results from the boundary node are replicated three times and randomly distributed on the cloud nodes through an optimized HDFS, which can preserve high volumes of large text files. For such huge data, the files are stored across multiple nodes in a redundant fashion to avoid possible data losses in the case of system failure or error, and provides the availability of parallel processing. This type of storage is called a data lake, which is a new type of cloud-based enterprise architecture that structures data in a more scalable way. Once data is collected on the cluster, preliminary data analysis and processing are used to structure the raw data and define the schema in advance. All data are transformed into the DataFrame format which is a distributed collection of data into named columns to provide an optimized data structure by utilizing the Apache Spark processing engine. In addition, the cluster applies Access Control Expressions which is a powerful model to control the access to data using Boolean logic expressions. All the data transmissions between the cloud nodes in this secure cluster are encrypted to prevent any potential attacks by accessing communication to obtain the contents of the transmission. For instance, the Secure Sockets Layer and the Transport Layer Security protocol are deployed in our system to secure several channels of HTTP traffic for communication security. Encryption also restricts the ability of an external party to read or modify data.

It is essential to choose an effective data storage facility and format for each use case to satisfy application requirements including response time, throughput and the number of concurrent accesses, etc. Therefore, in the proposed framework, all the transformed, processed data and data analysis results are stored in the MapR database with different table formats: JSON and Binary. The MapR Database is an extremely scalable, reliable, globally distributed NoSQL database for building powerful, intelligent, and mission-critical applications. It stores structured data as a nested series of maps. With the MapR database, a table is automatically partitioned across a cluster by a key range, and each server is the source for a subset of a table. The MapR JSON table supports query, read and write for a billion records within a few seconds, while the MapR Binary table is used for monitoring all the collected tag values in real time through OpenTSDB which is an API for time-series data.

7.4.5 Application Layer

The application layer directly serves the users through any self-developed and 3rd party software. It is common to include dashboarding requirements in the building of an IoT-based big data ecosystem, such as monitoring, reporting and interactive analytics. For a global manufacturing industry, all equipment running conditions from multiple plants are also needed to be monitored and managed in a central office by global engineers. Therefore, in this thesis, a self-built dashboard running on the boundary node which is shown in *Fig. 7-6* is designed and implemented to:

- visualise various types of data by utilising software Qlik [330];

- monitor sensors and predictive values of different equipment in real time;
- trigger alarms to warn the global engineers when equipment is running in abnormal conditions;
- summarise and display reports in terms of data quality, alarms, and equipment running conditions when users make requests.

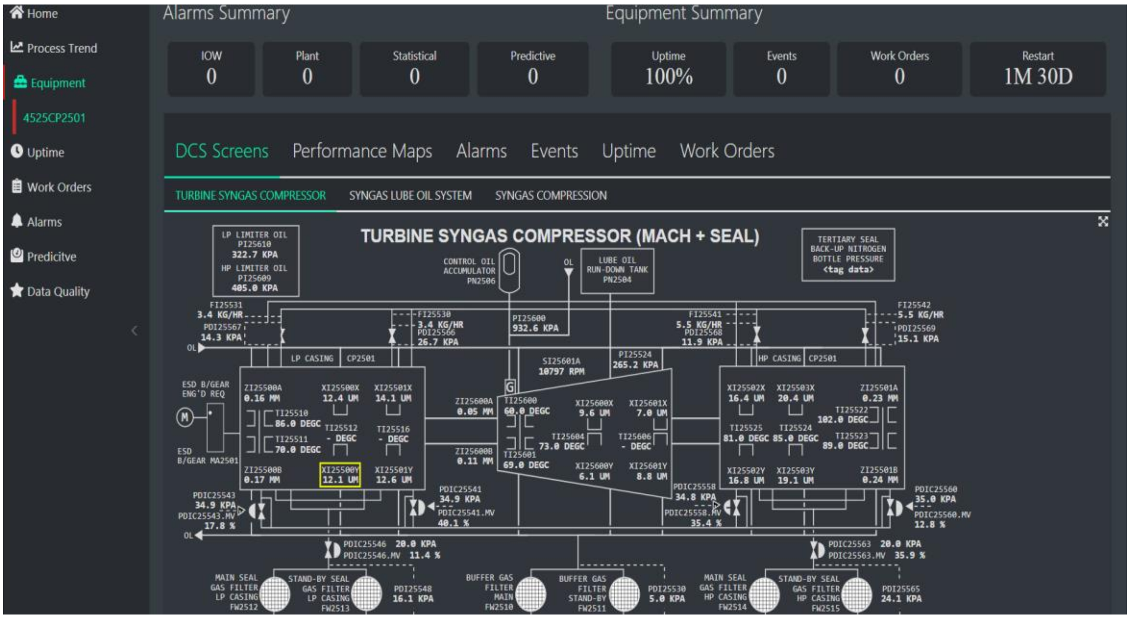


Fig. 7-6: Self-developed dashboard in real production

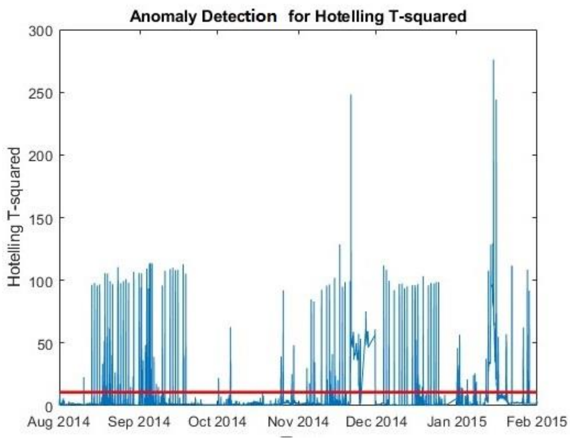
7.5 Experiment Results

To evaluate the fault prediction performance of the proposed optimized architectural framework, in this case study, we also use the six-month data (from August 2014 to February 2015) with 43 features collected from the installed sensors every second in a four stage compressor at the Phosphate Hill site. The features include temperature, pressure, vibration, speed and so on, as shown in *Fig. 6-3*. The raw data is read by the sensor and then transformed to column-based tag data in the DataFrame format after a series of processing. According to the engineers' feedback, the syngas compressor was running in normal condition without the occurrence of any faults for the first three months, so the data collected during this period are used for training the model, and the data collected in the last three months are used for testing the model. In addition, *Table 7-2* presents four main functional failures of the syngas compressor provided by the engineers during testing case time, which occurs at 29/12/2014 15:54, 20/01/2015 08:57, 26/01/2015 23:46 and 27/01/2015 14:39.

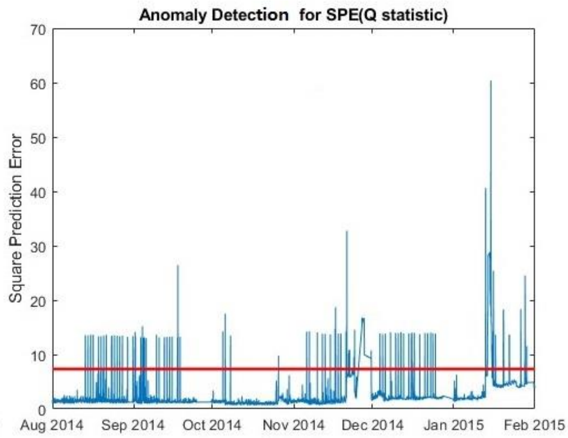
Table 7-2: Main functional failures of the syngas compressor

Fault ID	Category	Time of trip
(1)	Equipment functional failure	29/12/2014 15:54
(2)	Equipment functional failure	20/01/2015 08:57
(3)	Equipment functional failure	26/01/2015 23:46
(4)	Rotating equipment functional failure	27/01/2015 14:39

Simplicity is one of the critical challenges for building an IoT project architecture. Reducing system architecture complexity is key to the success of IoT applications. Therefore, this experiment will evaluate the fault detection performance of the distributed stacked sparse autoencoder without using noise removal and feature selection techniques. *Fig. 7-7* shows the PCA-based T-squared and SPE detected results without preprocessing the data. It can be seen that a large number of false alarms are generated due to the existence of noisy values and features and non-linear relationships in the 43 features. An autoencoder is an upgraded version of PCA to address the non-linear relationship issue among different sensors. However, *Fig. 7-8* presents the detected results by utilising a traditional autoencoder and many false alarms are also generated. *Fig. 7-9* shows the detected results by utilizing the proposed distributed stacked sparse autoencoder. The number of false alarms is reduced significantly, and all the events can be successfully detected. It can be concluded that the proposed distributed stacked sparse autoencoder demonstrates better fault detection performance compared to the PCA-based model and traditional autoencoder by improving robustness against noisy perturbations during the model training stage.



(a)



(b)

Fig. 7-7: PCA-based fault detection results. (a) and (b) show the results based on the T-squared and SPE thresholds

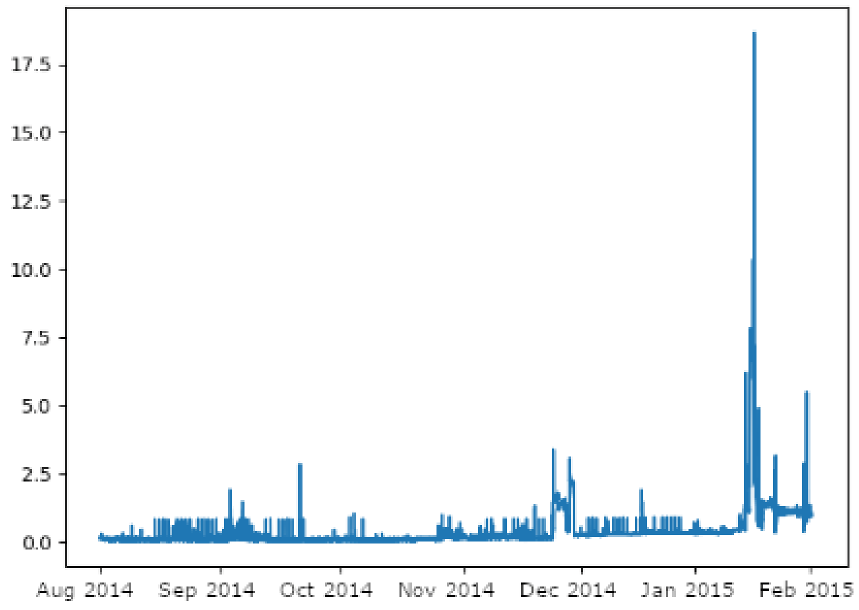


Fig. 7-8: Detected result of the traditional autoencoder

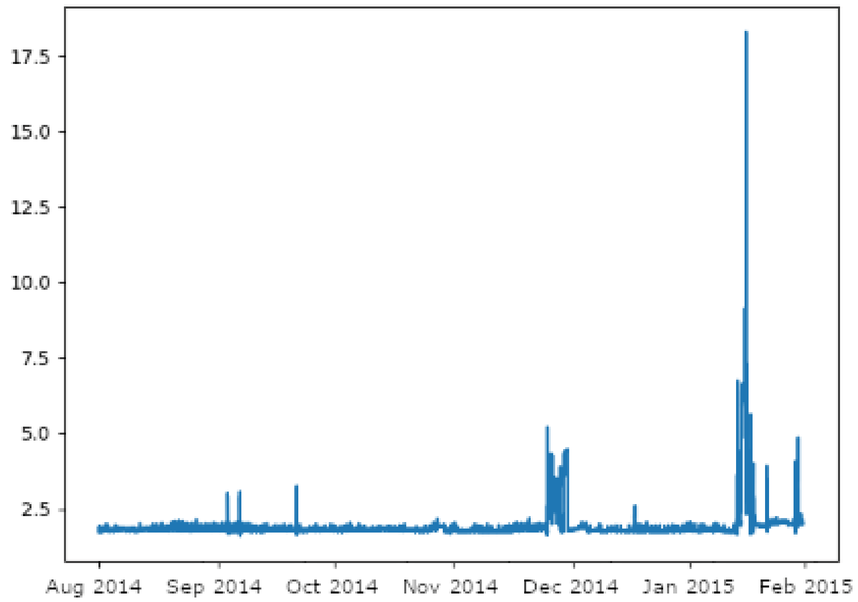


Fig. 7-9: Detected results of the distributed stacked sparse autoencoder

Table 7-3 presents the test results for (1), (2), (3) and (4) equipment functional failure times of the syngas compressor in 2014 and 2015. It can be seen that equipment functional failure time at 29/12/2014 15:54 is detected successfully by the distributed stacked sparse autoencoder at 29/12/2014 11:08 (five hours in advance). For the equipment functional failure at 20/01/2015 08:57, the algorithm delivers warnings eight days in advance at 12/01/2015 04:06, which provides the engineer sufficient time to identify the issues. Outages (3) and (4) are also successfully detected. The model starts to alarm the system at 26/01/2015 07:18 and continues to alarm until end of February. Therefore, the testing case results demonstrate that the distributed stacked sparse autoencoder is able to trigger alarms a few hours or days before the occurrence of the actual event.

Table 7-3: Test results containing four outages

Testing Case (Outage 1, 2, 3, 4)		
Pre-condition	Use specific range to filter dirty data based on engineer feedback (such as speed over 8000)	
Test Outage	Test Result	Test Comment
(1)	Success	Detect anomalies start at 29/12/2014 11:08
(2)	Success	Detect anomalies at 12/01/2015 04:06
(3) and (4)	Success	Detect anomalies at 26/01/2015 07:18

7.6 Conclusion

In this chapter, an optimized architectural framework for predictive maintenance is proposed to address the issues of increasing cloud cost, real-time monitoring and response requirements, and the existence of a nonlinear relationship among different types of sensors. The proposed optimized ecosystem consists of three new layers, namely the edge layer, cloud layer and application layer. It completely offloads real-time equipment fault detection and diagnosis functions from the cloud and only the processed data need to be transferred to the central cloud cluster. In addition, a distributed stacked sparse autoencoder is utilized as the fault detection model, which demonstrates a prominent performance especially when dealing with a large number of sensors which contains high noise values and features.

Chapter 8

Recapitulation and Future Work

This chapter concludes this thesis and discusses a possible research direction regarding a complementary study in advanced data preprocessing, the security issues in an IoT-based system, logistics support for predictive maintenance and model evolution.

8.1 Recapitulation of this Research

Since the advent of industrial systems, there has been an increasing requirement to ensure the reliability and safety of manufacturing machines and systems to reduce equipment unscheduled downtime, performance degradation and safety hazards. Traditional time-driven methods provide a maintenance guideline according to normal machine lifespans. In other words, the decision as to when to repair or replace a component of a machine is made mainly based on the domain knowledge or personal experience of the maintenance manager. However, a good maintenance management guideline should be capable of scheduling both preventive and corrective maintenance tasks on an as-needed basis in a more flexible way, instead of only depending on the machine average life statistics, such as Mean Time to Failure (MTTF).

A more effective approach is building a predictive maintenance system by monitoring the machine health condition and evaluating the need for maintenance through an anomaly detection technique and alarm mechanism which should be capable of giving an alert for maintenance sufficiently in advance of possible breakdown for personnel to react to prevent such a breakdown. It is crucial to capture any subtle underlying changes or abnormal behaviours to identify potential developing fault events at the earliest possible time.

Initially, computerized control and information techniques promoted the feasibility of fault detection in predictive maintenance, realizing the detection of potential failures as early as possible before equipment breakdowns. Traditional fault detection and diagnosis methods were classified into three categories in [331]: model-based methods, signal-based methods, and knowledge-based methods. However, the traditional condition monitoring techniques were focused on putting sensors on a specific piece of equipment with a computer control system associated with it. This required the expert engineers to able to be present at the manufacturing plant to check potential alarms.

The next stage of evolution used traditional networks to bring information to a single central monitoring point within the Manufacturing plant to allow the experts to monitor the information at the central system. These were known as SCADA systems. However, these systems were still relatively inflexible in that an addition of a new piece of equipment, or new measurements or change in the process structure of the plant would often require changes to the network hardware configuration and the processing software.

The Internet of Things carried a promise of overcoming these disadvantages. With the prosperity of the Internet of Things (IoT), industrial manufacturing has started to invest in the Industrial Internet of Things (IIoT) to achieve the evolutional transition from traditional manufacturing facilities into highly optimised smart manufacturing modes, which can significantly improve safety, operational efficiency, connectivity, scalability and cost savings. However, IIoT not only provides more insights into manufacturing systems, it also brings new challenges.

When faced with a high volume of IIoT data collected from a massive number of sensors located in hierarchical and multidimensional sources, the development of an IIoT-based intelligent manufacturing system brings several new issues in terms of:

- efficient big data ingestion and management, which is the most fundamental step in building an IIoT-based intelligent system;
- accurate analysis and prediction of equipment conditions by dealing with partially labelled or unlabelled sensor data due to the impracticality of manually labelling all conditions of various equipment [13];
- tremendous computation pressure, as conventional processing methods rely on local computers and simple parallel operations to improve computation power [332];
- achieving real-time streaming data processing and responding.

Several studies [61, 254] provide an excellent discussion on the challenges and opportunities of IIoT. To address these issues, IoT technologies, big data techniques, CPS and cloud computing are collaborated and integrated into a service-oriented architecture (SoA), which enables a big data processing framework and IoT-based fault detection ecosystem to be implemented [217, 71, 329, 333].

The emerging IoT has resulted in a new generation of intelligent digital sensors, which is gradually being implemented in smart manufacturing. IoT technology bridges the gap between physical equipment and cyber systems, enabling continuous data gathering, comprehensive data analysis, and real-time monitoring. Such capabilities lead to big data, foster Industry 4.0, and make IoT a fertile ground for the development of cyber physical systems, wireless communication, cloud computing, and so on. IoT is the prerequisite for the advancements of other technologies, and all of them in combination lead the evolution of modern smart manufacturing. With the availability of online monitoring data thanks to the IoT deployed in smart manufacturing, IoT-enabled predictive maintenance is able to provide the health condition assessment of equipment or systems across the entire factory in a real-time manner. It is noted that for now, IoT-enabled predictive maintenance is not like traditional predictive maintenance anymore which mainly concerns fault detection algorithms. IoT-enabled predictive maintenance involves a whole ecosystem, from data ingestion, transformation, management, analytics to data visualization, where fault-detection-related machine learning is only a part of the ecosystem. The design and deployment of such an ecosystem is based on the integration of IoT, CPS, wireless communication, cloud computing, big data analytics and artificial intelligence, thus posing diverse challenges including the ones that are inherited from each technology and the challenges in terms of how to seamlessly and securely integrate several technologies into one ecosystem.

To address the above problems, Chapter 5 proposes an integrated framework for manufacturing equipment health state monitoring and diagnostics in predictive maintenance using IoT big data in a real industrial setting, which can be easily used for smart manufacturing and can be implemented in Industry 4.0 systems. The intelligent sensors play a pivotal role in the success of IoT and the connectivity across networks results in massive volumes of streaming data. Big data brings many challenges and equipment health state monitoring for IoT smart manufacturing requires evolution. The proposed framework provides an integrated solution to this issue, comprising a big data ingestion phase, big data management phase, data preparation phase, and predictive modelling phase. In the data ingestion and management phase, data lake techniques are employed for flexible data storage and faster data retrieval. The big data flows between the collectors, servers, and clouds across the connected ingestion system. Sensor selection and noisy detection approaches are incorporated in the data preparation phase for efficient modelling and analytics. Given the large volume of rapidly arriving real-time data, based on the proposed big data ingestion system, unsupervised distributed PCA is embedded in the predictive modelling phase for event prediction and diagnosis due to its capability for online processing, its efficiency, and ease of deployment. The overall framework is implemented in the Apache Spark platform, which provides an environment to make it available for distribution and cloud computing to process the big data.

In order to evaluate the proposed big data ecosystem, we utilize the data collection from a remote site on the Ammonia plant at Phosphate Hill. Two case studies are discussed in Chapter 6 to validate whether the proposed ecosystem can predict and prevent equipment breakdowns. The experiment results of the first case study demonstrated that the proposed big data ecosystem was capable of generating alarms on the system in a real-time fashion, several days before an actual event, which enables the engineer to analyse the problem before any failures occur. The second case study results showed that the feature selection and noise detection techniques can significantly improve predictive model performance by addressing IoT data quality issues. During the test period in the actual production, our proposed big data ecosystem continually stores and processes all collected data in a centralized cloud, which leads to latency and cost issues. It is necessary to design an improved architectural framework to improve system response time and optimize cloud costs and resources.

Edge computing saves time and money by streamlining IoT communication, reducing system and network architecture complexity, and decreasing the number of potential failure points in an IoT application. Reducing system architecture complexity is key to the success of IoT applications. Chapter 7 proposes an optimized architectural framework with embedding edge computing into the IoT-based big data ecosystem proposed in Chapter 5. The proposed new architecture consists of three layers: the edge layer, the cloud layer and the application layer. Most of the computation is performed on the edge layer instead of the cloud layer by moving the whole process of data ingestion, data preprocessing and fault detection and diagnosis to the edge cluster, while big data management and data analytics are conducted on the cloud layer. Moreover, a distributed stacked sparse autoencoder model is proposed for fault detection instead of utilizing the DPCA algorithm in the edge layer to address the complicated non-linear relationships between multiple sensors. The experiment evaluation shows the superior capability of the proposed distributed stacked sparse autoencoder against noisy perturbations when training the fault detection model.

8.1.1 Summary of Contributions Made in this Thesis

In this thesis, we make the following contributions in two parts, namely:

1) An IoT-based global manufacturing big data ecosystem for predictive maintenance
2) An optimized architectural framework

For an IoT-based global manufacturing big data ecosystem for predictive maintenance:

- An integrated architecture is developed for fault detection and diagnosis in predictive maintenance in Industry 4.0, which consists of four layers, namely big data ingestion, management, preprocessing, and predictive modelling layers, achieving functionalities include IoT data ingestion, data management, sensor selection, noisy data cleaning, fault detection, contribution analysis and alarm sequence modelling, hence arriving at a comprehensive health condition report. Each functionality is equipped with effective and efficient techniques. To the best of our knowledge, this is the first work that includes all the necessary steps and details to construct an integrated manufacturing intelligent system.
- The developed architectural framework combines and integrates different methods and tools in heterogeneous layers in an appropriate way, which enables various computing and communication functions to be supported. This thesis provides a detailed API-oriented guideline for implementation.
- The developed architectural framework is generic, so it is not limited to one piece of equipment or human-specified critical components, rather, it can be applied to a wide array of machines or systems. The developed ecosystem can monitor the health states of all the equipment from various manufacturing plants to predict and prevent any failures in a central place, which brings enormous benefits to the expert engineers.
- By combining IoT big data and cloud computing, this thesis introduces the new challenges of sensor selection and noise detection problems for IoT big data and provide novel solutions in the data preprocessing layer. In order to overcome the big data difficulties related to the generated large volume and high velocity of data, we developed the distributed PCA (DPCA) model in this thesis, which was utilized during the predictive phase and avoided the need for a complex model structure and resource-hungry algorithms. The advantages of distributed PCA lies in its scalability to cope with the large volume of data which arrives at a high velocity to provide real-time responses as the data arrives. We achieve the DPCA utilizing a singular value decomposition of the matrix before determining the eigenvectors and

eigenvalues in a distributed fashion. To provide effective accessing of the large volume of different types of data, we employ a MapReduce in conjunction with DPCA to give us a MapReduce-based DPCA. The implementation of this MapReduce-based DPCA is carried out on the Apache Spark processing engine with its real-time analytics ability with cloud computing, which is critical for achieving the required real-time response in the presence of big data with its high volume, high velocity and different types of data. It also takes advantage of unsupervised learning without the need for labels, as it is impossible to collect all kinds of failure data in large-scale monitored factories.

- The developed architectural framework is practical and applicable to Industry 4.0. It was implemented successfully in a real industrial environment in a cooperating company, tested over several years, and the project won the Best Industry Application of IoT at the BigInsights Data and AI Innovation Awards [227].

For an optimized architectural framework:

- An optimized IoT-based big data ecosystem embedded with edging computing is developed for real-time fault detection and diagnosis processing in predictive maintenance, which consists of three layers, namely edge layer, cloud layer and application layer. We provide a seamless integration between the edge computing layer and the currently proposed cloud computing-based IoT ecosystem. Our new framework shifts most of the computation on the edge layer instead of the central cloud cluster by moving certain workloads to the edge cluster.
- Unlike previous research which is limited to the presentation of the edge computing structure, we additionally introduce the proposed optimized ecosystem through a new perspective. Behind the ecosystem structure, various advanced software techniques are presented and integrated that enable it to be successfully launched and deployed in industrial settings. To the best of our knowledge, this is the first work from the business application perspective, detailing the implementations of edge computing to assist the IoT to store and process increasing growing scalable big data.
- A distributed stacked sparse autoencoder model was developed for fault detection in predictive maintenance on the Apache Spark Platform in the edge layer of our optimized IoT-based big data ecosystem to address the complicated non-linear relationships between multiple sensors. It is an upgraded version of DPCA, which is only feasible when the sensors have linear correlations. The prediction model is

built and trained in the cloud and the pre-trained model is deployed and located in the edge layer, running on-premises. Once an issue is detected, IoT edge triggers an alert and processes the data locally with a high priority and sends it to the cloud for further analysis.

8.2 Complementary Studies and Future Research

The work in this section includes two different sets of discussions, namely:

- Complementary work which was beyond the scope of the thesis and was being done at the same time as the thesis by another PhD scholar.
- Future research which discusses topics that arose from the work of this thesis that would be valuable to investigate in the future.

8.2.1 Advanced Data Preprocessing

The material described in this section constitutes complementary research and is important for the whole IIoT-based big data ecosystem but for reasons of time was considered to be outside the scope of this thesis. It was addressed by a complementary thesis by Yue Hua Liu which was done partly in parallel with my thesis. It consists of three parts and these are :

a) Noise Removal
b) Data Imputation
c) Feature Selection

Below we briefly describe the motivations and challenges associated with doing this research.

Noise Removal: Unlike the general time series data which can only have normal data and outliers, IIoT monitoring time series data have normal data, anomalies and noise. But anomalies are meaningful and indicate when the fault occurs. It is the main information that can be used for fault detection in predictive maintenance. Noise, on the other hand, is spurious and erroneous information and would lead to the model mis-specification. So, distinguishing the noise from anomalies then removing it can improve the data quality and facilitate more accurate fault detection results. From the viewpoint of patterns, anomalies and noise both belong to outliers, which brings a critical challenge on noise removal in the presence of significant anomalies. This is beyond the work of traditional work on noise removal which often relies on filtering outliers as these would remove both noise and significant anomalies. Specifically, it is important when removing noise in the IIoT ecosystem, that the noise is removed but not significant anomalies which constitutes vital information which must be retained.

Data Imputation: Missing data is a common concern in the manufacturing industry especially in IIoT systems. Since it is a system to collect, transfer, and analyse IoT data, any failures of the sensors, edge nodes and communication links during this process would result in data errors and loss of information. How to impute or replace the missing data with the most reasonable values to improve the data quality is another important issue. Several approaches utilized the existing values from other sensors to evaluate the missing values in the current sensor. However, in the IoT systems, common-mode failures always occur in a large number of sensors, as a result, the values from these sensors are missing at the same time, in which case methods based on the existing values from other related sensors are not feasible anymore and new approaches need to be developed.

Advanced Feature Selection: For an IoT system deployed in a plant-wide manufacturing, a wide array of equipment is monitored. If all features collected from sensors across the whole manufacturing plant are incorporated into the fault detection model, the features which are responsible for different pieces of equipment might negatively affect the detection of faults that occur in each other. This is especially the case in IoT systems as one-to-one labels are inaccessible. Without feature selection, all input features are given equal weight in the fault detection models. Other features which are not responsible for this specific equipment would bring disruptive information. Therefore, it is crucial to develop a feature selection method to effectively detect multiple faults which occurs in different pieces of the equipment across the whole manufacturing plant. In the work in Chapter 5 we used a method of feature selection which removed high variance features to achieve feature selection. This produced some improvement in the results. However, in addition to the high variance characteristic, we also need to develop a more advanced feature selection approach which determines the relevance of a particular input feature when carrying out health state determination or fault diagnosis.

8.2.2 Security Issues of IoT-based System

As the enabling technologies of Industry 4.0-driven predictive maintenance ecosystems, IoT, CPS and cloud computing are integrated to be capable of dealing with IoT streaming big data and supporting IoT applications with real-time services. Smart manufacturing is born from a convergence of such enabling technologies. Based on a real industry project in which an Australia company invested, we proposed one integrated ecosystem and an upgraded version on top of that. Our work provides a reference for researchers in academia and industry who want a better understanding in terms of both techniques and implementations. Since the solutions we proposed are based on a project with less stringent requirements in security, we equip the proposed ecosystem with some basic techniques such as encryption, decryption, access control and so on, but did not conduct in-depth analysis on it. However, this does not mean that privacy and security are not important as these issues are key concerns in several specific scenarios related to human safety or where sensitive data is involved. With IoT widely spreading around the world in the future, trust is essential to realise the full potential of the IoT.

A general IoT-based architecture inherits all the security problems specific to each layer (i.e. physical device layer, network layer, and application layer) and also contains security threats specific to the gateways that connect these layers [334]. The work in [335, 336] specifically focus on the physical layer security for IoT. The work in [337] provides a detailed categorization of security issues that might occur for a wide variety of devices and equipment ranging from small embedded processing chips to large high-end servers. In the work in [334], specific security and privacy issues related to different IoT applications and from different IoT layers are discussed with countermeasures. The involved IoT applications include smart cities, smart environment, smart metering and smart grids, smart retail, smart agriculture and animal farming, and smart homes. Four layers, the sensing layer, network layer, middle-ware layer and application layer, and the gateway that connects different layers are also considered for the identification of possible security issues. The work in [338] presents another classification of IoT attacks and analyses the security issues in different layers of an architecture. Khattak et al. [339] describe the IoT security requirements and classify various attacks at different layers with a focus on the perception layer of IoT, discussing security issues, the countermeasures, and the research challenges. IoT-based ecosystems are complex. A digital security risk is present at every step along the IoT journey, and there are a large number of hackers that will take advantage of a system's vulnerability.

Researchers also point to a number of security concerns mainly coming from aspects of IoT devices, wireless communication, and cloud computing. Since IoT devices and networking appliances are currently being pervasively used and are relatively new, security has not always been considered in product design. Huge potential risks exist pertaining to a large number of unsecured devices connecting to the Internet. A secure wireless communication system involves authentication and secure transmission [340]. Although the cloud plays a central role in storing, processing and distributing data and has contributed greatly to IoT applications, the strong dependencies on centralized servers pose significant trust issues [341]. The work in [342] comprehensively surveys security issues in terms of IoT and cloud computing. With regard to the IoT, decode-and-forward and amplify-and-forward are mainly discussed, and AES and RSA are listed as solutions for the secure integration of IoT and cloud technologies. Yang et al. [343] proposed a privacy-preserving IoT-based storage system for healthcare big data, where self-adaptive access control is designed and embedded to ensure the security of patients' healthcare data. The work in [344] proposes a Hadoop-based system architecture of the distributed file system, targeting the security of IoT data storage in cloud computing. In the system, an enhanced ant colony algorithm is designed to optimize task-resource scheduling in the cloud. Zhou et al. [345] focus on the security threats for cloud-based IoT, and list a taxonomy of main security and privacy requirements which encompasses identity privacy, location privacy, node compromise attack, layer removing/adding attack, forward and backward security, semi-trusted/malicious cloud security, and so on and provide a review of the corresponding countermeasures. Then a new method is proposed especially for secure packet forwarding.

It can be seen from the above discussions that data security issues have become a serious concern in IoT systems. To address this, various studies and investigations have been conducted on many factors, including the physical security of facilities such as IoT devices, and cyber security for networking and the cloud. Although security issues have already gained much attention, IoT and other technologies that have been developed based on IoT are still evolving, thus the security that is associated with all these technologies also requires advancements. For example, manufactures roll out new products more quickly and they continue to be connected to the Internet but security is often given a low priority as the focus is on time-to-market and return-on-investment metrics. The diversity of IoT means there is no 'one size fits all' security solution that can protect any IoT deployment. More importantly, there is a lack of comprehensive studies on security that are specifically related or apply to IoT-enabled predictive maintenance. Therefore, the problems of security and privacy for IoT-based smart manufacturing needed further study.

8.2.3 Logistics Support for Predictive Maintenance

Predictive maintenance in manufacturing can aid in determining the right moment to replace a part. In the proposed ecosystem, we focus on detecting when faults are going to occur but have not discussed the logistical support which is responsible for providing the equipment that needs to be replaced in advance. An optimized inventory management for logistics support will not only reduce the investments and inventory of the parts that are not actually needed, but it is also able to provide the off-the-shelf part that is needed as soon as possible. If the equipment that we need is not available in the inventory, the replacement of one piece of equipment might cause the long shut-down of the whole manufacturing plant as well because some of the equipment in a manufacturing plant is very specialized and may need to be purchased from overseas. Moreover, the time it takes to order and receive spare parts and their stocking quantities should be planned so that holding costs are minimized while also avoiding stock-outs.

8.2.4 Model Evolution

The model for fault detection in predictive maintenance is trained offline and is then used to detect possible faults in an online mode. As time goes by, the characteristics of the input data stream may vary, so our proposed solution is to update the detection model periodically. How to adjust the model based on the current data through online/streaming learning for better fault detection for future input data is also the research direction that we can explore in the future. In an online mode, the model is trained and evolves continuously on new incoming data and adapts to the changing data characteristics. However, we still need to consider the trade-off between accuracy and efficiency, since online learning can improve performance but definitely requires more computation.

8.2.5 Migration Process

Another common issue that needs to be addressed in the future for companies is the migration from the legacy systems to the new systems proposed in this thesis. Currently, an increasingly number of organizations are immersed in digital transformation and digital solutions. The key aspect of the migration process is to clearly delineate “the system as is before migration” and “the system to be after the migration”. The next important aspect is to define the strategy for going from the system as is to the system to be. The last step in the process is verification of the proposed system in situ in the manufacturing plant followed by the process of going live with the new system which involves switching from the old system to the new system and monitoring the new system over a period of time after the cutover to ensure that it is functioning correctly.

For industrial manufacturing, in determining the as is system the several factors are important, such as:

- whether the company has a monitoring system for individual machines;
- whether the company has a large cloud service;
- whether the company has an extensive database with software capability for handling big data.

In our research project, after carefully analysing the business value and requirements, IPL started the study and constructed the digital ecosystem from the scratch which was deployed as a parallel system to the existing “as is system”. Several characteristics had been carefully analysed, including the reliability of the upgraded sensors and data transformation, the understanding of speed of processing that was needed to achieve the different services, the size of data we wanted to store and how quickly we wanted to refresh the values in our dashboard. This experience can be utilized as a reference for other organizations for designing digital solutions. The migration process is one of the greatest challenges in digital transformation in terms of the analysis of business requirements and the consideration of cost, time and risk. Therefore, this can be considered as a research project in its own right in the future.

List of References

[1] A. Davies, *Handbook of condition monitoring: techniques and methodology*. Springer Science & Business Media, 2012.

[2] Y. He, C. Gu, Z. Chen, and X. Han, "Integrated predictive maintenance strategy for manufacturing systems by combining quality control and mission reliability analysis," *International Journal of Production Research*, vol. 55, no. 19, pp. 5841–5862, 2017.

[3] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[4] D. P. Filev, R. B. Chinnam, F. Tseng, and P. Baruah, "An industrial strength novelty detection framework for autonomous equipment monitoring and diagnostics," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 4, pp. 767–779, 2010.

[5] Y. Lei, F. Jia, J. Lin, S. Xing, and S. X. Ding, "An intelligent fault diagnosis method using unsupervised feature learning towards mechanical big data," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 5, pp. 3137–3147, 2016.

[6] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi, "Machine learning for predictive maintenance: A multiple classifier approach," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 812–820, 2014.

[7] L. Wang, Z. Zhang, H. Long, J. Xu, and R. Liu, "Wind turbine gearbox failure identification with deep neural networks," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1360–1368, 2016.

[8] P. Bangalore and L. B. Tjernberg, "An artificial neural network approach for early fault detection of gearbox bearings," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 980–987, 2015.

[9] K. Pichler, M. Pichler, T. Buchegger, E. Lughofer, E. P. Klement, and M. Huschenbett, "On the robustness of fault detection in reciprocating compressor valves," in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2014, pp. 2733–2738.

[10] N. K. Verma, R. K. Sevakula, S. Dixit, and A. Salour, "Intelligent condition based monitoring using acoustic signals for air compressors," *IEEE Transactions on Reliability*, vol. 65, no. 1, pp. 291–309, 2015.

[11] L. Tan and N. Wang, "Future internet: The internet of things," in *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, vol. 5. IEEE, 2010, pp. V5–376.

[12] C. Duan, L. Yang, Q. Lin, Y. Liu, and L. Xie, "Robust spinning sensing with dual-RFID-tags in noisy settings," *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, pp. 2647–2659, 2018.

[13] Y. He, J. Guo, and X. Zheng, "From surveillance to digital twin: Challenges and recent advances of signal processing for industrial internet of things," *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 120–129, 2018.

[14] J. Wang, F. Adib, R. Knepper, D. Katabi, and D. Rus, "Rf-compass: Robot object manipulation using rfids," in *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking*, 2013, pp. 3–14.

[15] T. Wei and X. Zhang, "Gyro in the air: tracking 3d orientation of batteryless Internet-of-Things," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, 2016, pp. 55–68.

[16] L. Shangguan and K. Jamieson, "Leveraging electromagnetic polarization in a two-antenna whiteboard in the air," in *Proceedings of the 12th International on Conference on Emerging Networking Experiments and Technologies*, 2016, pp. 443–456.

[17] J. Liu, M. Chen, S. Chen, Q. Pan, and L. Chen, "Tag-compass: Determining the spatial direction of an object with small dimensions," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[18] S. Manzari, C. Occhiuzzi, S. Nawale, A. Catini, C. Di Natale, and G. Marrocco, "Polymer-doped uhf rfid tag for wireless-sensing of humidity," in *2012 IEEE International Conference on RFID (RFID)*. IEEE, 2012, pp. 124–129.

[19] P. Yang, Y. Feng, J. Xiong, Z. Chen, and X.-Y. Li, "Rf-ear: Contactless multidevice vibration sensing and identification using cots rfid," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 297–306.

[20] L. Yang, Y. Li, Q. Lin, X.-Y. Li, and Y. Liu, "Making sense of mechanical vibration period with sub-millisecond accuracy using backscatter signals," in *Proceedings of*

the 22nd Annual International Conference on Mobile Computing and Networking, 2016, pp. 16–28.

[21] J. Han, C. Qian, X. Wang, D. Ma, J. Zhao, W. Xi, Z. Jiang, and Z. Wang, "Twins: Device-free object tracking using passive tags," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1605–1617, 2015.

[22] Y. Zheng, Y. He, M. Jin, X. Zheng, and Y. Liu, "Red: Rfid-based eccentricity detection for high-speed rotating machinery," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 1565–1573.

[23] J. Guo, T. Wang, M. Jin, S. Yang, C. Jiang, L. Liu, and Y. He, "Tagleak: Nonintrusive and battery-free liquid leakage detection with backscattered signals," in *2018 IEEE International Conference on Industrial Internet (ICII)*. IEEE, 2018, pp. 40–48.

[24] X. Wang, J. Zhang, Z. Yu, S. Mao, S. C. Periaswamy, and J. Patton, "On remote temperature sensing using commercial uhf rfid tags," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10715–10727, 2019.

[25] J. Wang, J. Xiong, X. Chen, H. Jiang, R. K. Balan, and D. Fang, "Tagscan: Simultaneous target imaging and material identification with commodity rfid devices," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, 2017, pp. 288–300.

[26] C. Gao, Y. Li, and X. Zhang, "Livetag: Sensing human-object interaction through passive chipless wifi tags," in *15th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 18)*, 2018, pp. 533–546.

[27] C. Jiang, Y. He, X. Zheng, and Y. Liu, "Orientation-aware rfid tracking with centimeter-level accuracy," in *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2018, pp. 290– 301.

[28] F. Xiao, Z. Wang, N. Ye, R. Wang, and X.-Y. Li, "One more tag enables fine-grained rfid localization and tracking," *IEEE/ACM Transactions on Networking*, vol. 26, no. 1, pp. 161–174, 2017.

[29] Y. Ma, N. Selby, and F. Adib, "Minding the billions: Ultra-wideband localization for deployed rfid tags," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, 2017, pp. 248–260.

[30] W. Liu, L. Zheng, D. Sun, X. Liao, M. Zhao, J. Su, and Y. Liu, "Rfid-enabled real-time production management system for loncin motorcycle assembly line," *International Journal of Computer Integrated Manufacturing*, vol. 25, no. 1, pp. 86–99, 2012.

[31] Q.Dai, R.Zhong, G.Q. Huang, T.Qu, T.Zhang, and T.Luo, "Radio frequency identification-enabled real-time manufacturing execution system: a case study in an automotive part manufacturer," *International Journal of Computer Integrated Manufacturing*, vol. 25, no. 1, pp. 51–65, 2012.

[32] T. Qu, H. Yang, G. Q. Huang, Y. Zhang, H. Luo, and W. Qin, "A case of implementing rfid-based real-time shop-floor material management for household electrical appliance manufacturers," *Journal of Intelligent Manufacturing*, vol. 23, no. 6, pp. 2343–2356, 2012.

[33] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for iot big data and streaming analytics: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2923–2960, 2018.

[34] F. Shrouf, J. Ordieres, and G. Miragliotta, "Smart factories in industry 4.0: A review of the concept and of energy management approached in production based on the internet of things paradigm," in *2014 IEEE International Conference on Industrial Engineering and Engineering Management*. IEEE, 2014, pp. 697–701.

[35] K. Zhou, T. Liu, and L. Zhou, "Industry 4.0: Towards future industrial opportunities and challenges," in *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*. IEEE, 2015, pp. 2147–2152.

[36] Y.Lu, "Industry 4.0: A survey on technologies, applications and open research issues," *Journal of Industrial Information Integration*, vol. 6, pp. 1–10, 2017.

[37] R. Baheti and H. Gill, "Cyber-physical systems," *The Impact of Control Technology*, vol. 12, no. 1, pp. 161–166, 2011.

[38] E. A. Lee, "Cyber physical systems: Design challenges," in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*. IEEE, 2008, pp. 363–369.

[39] M. Hilbert, “Big data for development: A review of promises and challenges,” *Development Policy Review*, vol. 34, no. 1, pp. 135–174, 2016.

[40] W. Fan and A. Bifet, “Mining big data: current status, and forecast to the future,” *ACM SIGKDD explorations newsletter*, vol. 14, no. 2, pp. 1–5, 2013.

[41] H. Hu, Y. Wen, T.-S. Chua, and X. Li, “Toward scalable systems for big data analytics: A technology tutorial,” *IEEE access*, vol. 2, pp. 652–687, 2014.

[42] Y. Demchenko, P. Grosso, C. De Laat, and P. Membrey, “Addressing big data issues in scientific data infrastructure,” in *2013 International Conference on Collaboration Technologies and Systems (CTS)*. IEEE, 2013, pp. 48–55.

[43] D. Barton and D. Court, “Making advanced analytics work for you,” *Harvard business review*, vol. 90, no. 10, pp. 78–83, 2012.

[44] M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiqa, and I. Yaqoob, “Big iot data analytics: architecture, opportunities, and open research challenges,” *IEEE Access*, vol. 5, pp. 5247–5261, 2017.

[45] J. Dean and S. Ghemawat, “Mapreduce: simplified data processing on large clusters,” *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[46] T. White, *Hadoop: The definitive guide*. ” O’Reilly Media, Inc.”, 2012.

[47] “Apache spark, ”http://spark.apache.org/”.”

[48] D. Singh and C. K. Reddy, “A survey on platforms for big data analytics,” *Journal of big data*, vol. 2, no. 1, p. 8, 2015.

[49] V. Morabito, “Managing change for big data driven innovation,” in *Big Data and Analytics*. Springer, 2015, pp. 125–153.

[50] A. Bhardwaj, S. Bhattacherjee, A. Chavan, A. Deshpande, A. J. Elmore, S. Madden, and A. G. Parameswaran, “Datahub: Collaborative data science & dataset version management at scale,” *arXiv preprint arXiv:1409.0798*, 2014.

[51] F. F¨arber, S. K. Cha, J. Primsch, C. Bornh¨ovd, S. Sigg, and W. Lehner, “Sap hana database: data management for modern business applications,” *ACM Sigmod Record*, vol. 40, no. 4, pp. 45–51, 2012.

[52] S. Burke, “Hp haven big data platform is gaining partner momentum,” *CRN [online] http://www. crn. com/news/applications-os*, vol. 240161649, 2013.

[53] “Hortonworks, 2017[online]. available: https://hortonworks.com/.”

[54] Y. Zhuang, Y. Wang, J. Shao, L. Chen, W. Lu, J. Sun, B. Wei, and J. Wu, “D-ocean: an unstructured data management system for data ocean environment,” *Frontiers of Computer Science*, vol. 10, no. 2, pp. 353–369, 2016.

[55] D. Sl´ zak, P. Synak, J. Wr´oblewski, and G. Toppin, “Infobright analytic database engine using rough sets and granular computing,” in *2010 IEEE International Conference on Granular Computing*. IEEE, 2010, pp. 432–437.

[56] M. H. ur Rehman, I. Yaqoob, K. Salah, M. Imran, P. P. Jayaraman, and C. Perera, “The role of big data analytics in industrial internet of things,” *Future Generation Computer Systems*, vol. 99, pp. 247–259, 2019.

[57] J. Lee, E. Lapira, B. Bagheri, and H.-a. Kao, “Recent advances and trends in predictive manufacturing systems in big data environment,” *Manufacturing letters*, vol. 1, no. 1, pp. 38–41, 2013.

[58] R.Y. Zhong, X.Xu, E.Klotz, and S.T. Newman, “Intelligent manufacturing in the context of industry 4.0: a review,” *Engineering*, vol. 3, no. 5, pp. 616–630, 2017.

[59] M. Strohbach, H. Ziekow, V. Gazis, and N. Akiva, “Towards a big data analytics framework for iot and smart city applications,” in *Modeling and processing for next-generation big-data technologies*. Springer, 2015, pp. 257–282.

[60] S. Landset, T. M. Khoshgoftaar, A. N. Richter, and T. Hasanin, “A survey of open source tools for machine learning with big data in the hadoop ecosystem,” *Journal of Big Data*, vol. 2, no. 1, p. 24, 2015.

[61] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, “Industrial internet of things: Challenges, opportunities, and directions,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

[62] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, “The rise of “big data” on cloud computing: Review and open research issues,” *Information systems*, vol. 47, pp. 98–115, 2015.

[63] S. Sharma, "Expanded cloud plumes hiding big data ecosystem," *Future Generation Computer Systems*, vol. 59, pp. 63–92, 2016.

[64] G.Manogaran, R.Varatharajan, D.Lopez, P.M. Kumar, R.Sundarasekar, and C. Thota, "A new architecture of internet of things and big data ecosystem for secured smart healthcare monitoring and alerting system," *Future Generation Computer Systems*, vol. 82, pp. 375–387, 2018.

[65] Z.Xu, L.Chen, C.Chen, and X.Guan, "Joint clustering and routing design for reliable and efficient data collection in large-scale wireless sensor networks," *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 520–532, 2015.

[66] M. Dong, K. Ota, and A. Liu, "Rmer: Reliable and energy-efficient data collection for large-scale wireless sensor networks," *IEEE Internet of Things Journal* vol. 3, no. 4, pp. 511–519, 2016.

[67] A. P. Plageras, K. E. Psannis, C. Stergiou, H. Wang, and B. B. Gupta, "Efficient IoT-based sensor big data collection–processing and analysis in smart buildings," *Future Generation Computer Systems*, vol. 82, pp. 349–357, 2018.

[68] L. Jiang, L. Da Xu, H. Cai, Z. Jiang, F. Bu, and B. Xu, "An IoT-oriented data storage framework in cloud computing platform," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1443–1451, 2014.

[69] F. Tao, Y. Cheng, L. Da Xu, L. Zhang, and B. H. Li, "Cciot-cmfg: cloud computing and internet of things-based cloud manufacturing service system," *IEEE transactions on industrial informatics*, vol. 10, no. 2, pp. 1435–1442, 2014.

[70] C. Wang, Z. Bi, and L. Da Xu, "IoT and cloud computing in automation of assembly modeling systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1426–1434, 2014.

[71] J. Wan, S. Tang, D. Li, S. Wang, C. Liu, H. Abbas, and A. V. Vasilakos, "A manufacturing big data solution for active preventive maintenance," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2039–2047, 2017.

[72] G. A. Susto, A. Beghi, and C. De Luca, "A predictive maintenance system for epitaxy processes based on filtering and prediction techniques," *IEEE Transactions on Semiconductor Manufacturing*, vol. 25, no. 4, pp. 638–649, 2012.

[73] H. Wang, “A survey of maintenance policies of deteriorating systems,” *European journal of operational research*, vol. 139, no. 3, pp. 469–489, 2002.

[74] J. J. McCall, “Maintenance policies for stochastically failing equipment: a survey,” *Management science*, vol. 11, no. 5, pp. 493–524, 1965.

[75] R.E. Barlow and F.Proschan, *Mathematical theory of reliability*. SIAM, 1996.

[76] R. Barlow and L. Hunter, “Optimum preventive maintenance policies,” *Operations research*, vol. 8, no. 1, pp. 90–100, 1960.

[77] K.-S. Park, “Optimal wear-limit replacement with wear-dependent failures,” *IEEE Transactions on Reliability*, vol. 37, no. 3, pp. 293–294, 1988.

[78] K. S. Park, “Optimal continuous-wear limit replacement under periodic inspections,” *IEEE Transactions on reliability*, vol. 37, no. 1, pp. 97–102, 1988.

[79] A. Christer and W. Wang, “A model of condition monitoring of a production plant,” *The International Journal Of Production Research*, vol. 30, no. 9, pp. 2199–2211, 1992.

[80] A. Christer and W. Wang, “A simple condition monitoring model for a direct monitoring process,” *European journal of operational research*, vol. 82, no. 2, pp. 258–269, 1995.

[81] W. Feller, *An introduction to probability theory and its applications, vol 2*. John Wiley & Sons, 2008.

[82] W.-b. Wang, “A model to determine the optimal critical level and the monitoring intervals in condition-based maintenance,” *International Journal of Production Research*, vol. 38, no. 6, pp. 1425–1436, 2000.

[83] A. Grall, L. Dieulle, C. B´erenguer, and M. Roussignol, “Continuous-time predictive-maintenance scheduling for a deteriorating system,” *IEEE transactions on reliability*, vol. 51, no. 2, pp. 141–150, 2002.

[84] J. Wang, L. Zhang, L. Duan, and R. X. Gao, “A new paradigm of cloud-based predictive maintenance for intelligent manufacturing,” *Journal of Intelligent Manufacturing*, vol. 28, no. 5, pp. 1125–1137, 2017.

[85] L. Krishnamurthy, R. Adler, P. Buonadonna, J. Chhabra, M. Flanigan, N. Kushalnagar, L. Nachman, and M. Yarvis, “Design and deployment of industrial

sensor networks: experiences from a semiconductor plant and the north sea," in *Proceedings of the 3rd international conference on Embedded networked sensor systems*, 2005, pp. 64–75.

[86] R. Langone, C. Alzate, B. De Ketelaere, and J. A. Suykens, "Kernel spectral clustering for predicting maintenance of industrial machines," in *2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*. IEEE, 2013, pp. 39–45.

[87] M. Compare, P. Baraldi, and E. Zio, "Challenges to IoT-enabled predictive maintenance for industry 4.0," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4585–4597, 2019.

[88] A. Kusiak, "Smart manufacturing must embrace big data," *Nature*, vol. 544, no. 7648, pp. 23–25, 2017.

[89] R. Sahal, J. G. Breslin, and M. I. Ali, "Big data and stream processing platforms for industry 4.0 requirements mapping for a predictive maintenance use case," *Journal of Manufacturing Systems*, vol. 54, pp. 138–151, 2020.

[90] L. Wang, M. T¨orngren, and M. Onori, "Current status and advancement of cyber-physical systems in manufacturing," *Journal of Manufacturing Systems* vol. 37, pp. 517–527, 2015.

[91] P. Wang, R. X. Gao, and Z. Fan, "Cloud computing for cloud manufacturing: benefits and limitations," *Journal of Manufacturing Science and Engineering*, vol. 137, no. 4, 2015.

[92] Y. Lu, X. Xu, and J. Xu, "Development of a hybrid manufacturing cloud," *Journal of Manufacturing Systems*, vol. 33, no. 4, pp. 551–566, 2014.

[93] D. Wu, D. W. Rosen, and D. Schaefer, "Cloud-based design and manufacturing: status and promise," in *Cloud-based design and manufacturing (CBDM)*. Springer, 2014, pp. 1–24.

[94] S. Chr´etien, N. Herr, J.-M. Nicod, and C. Varnier, "Post-prognostics decision for optimizing the commitment of fuel cell systems," *IFAC-PapersOnLine*, vol. 49, no. 28, pp. 168–173, 2016.

[95] H. Skima, C. Varnier, E. Dedu, K. Medjaher, and J. Bourgeois, "Postprognostics decision making in distributed mems-based systems," *Journal of Intelligent Manufacturing*, vol. 30, no. 3, pp. 1125–1136, 2019.

[96] J. M. Sim˜oes, C. F. Gomes, and M. M. Yasin, "A literature review of maintenance performance measurement," *Journal of Quality in Maintenance Engineering*, 2011.

[97] A. K. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical systems and signal processing*, vol. 20, no. 7, pp. 1483–1510, 2006.

[98] O. E. Dragomir, R. Gouriveau, F. Dragomir, E. Minca, and N. Zerhouni, "Review of prognostic problem in condition-based maintenance," in *2009 European Control Conference (ECC)*. IEEE, 2009, pp. 1587–1592.

[99] E. Zio, "Prognostics and health management of industrial equipment," in *Diagnostics and prognostics of engineering systems: methods and techniques*. IGI Global, 2013, pp. 333–356.

[100] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel, "Prognostics and health management design for rotary machinery systems—reviews, methodology and applications," *Mechanical systems and signal processing*, vol. 42, no. 1-2, pp. 314–334, 2014.

[101] 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*, vol. 213, no. 1, pp. 1–14, 2011.

[102] W. Zhang, D. Yang, and H. Wang, "Data-driven methods for predictive maintenance of industrial equipment: A survey," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2213–2227, 2019.

[103] S. Yin, X. Li, H. Gao, and O. Kaynak, "Data-based techniques focused on modern industry: An overview," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 657–667, 2014.

[104] S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 11, pp. 6418–6428, 2014.

[105] T. Benkedjouh, K. Medjaher, N. Zerhouni, and S. Rechak, "Health assessment and life prediction of cutting tools based on support vector regression," *Journal of Intelligent Manufacturing*, vol. 26, no. 2, pp. 213–223, 2015.

[106] M. A. Mazzoletti, G. R. Bossio, C. H. De Angelo, and D. R. Espinoza-Trejo, "A model-based strategy for interturn short-circuit fault diagnosis in pmsm," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 9, pp. 7218–7228, 2017.

[107] D. Siegel, C. Ly, and J. Lee, "Methodology and framework for predicting helicopter rolling element bearing failure," *IEEE Transactions on Reliability*, vol. 61, no. 4, pp. 846–857, 2012.

[108] T. W. Chow and H.-Z. Tan, "Hos-based nonparametric and parametric methodologies for machine fault detection," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 5, pp. 1051–1059, 2000.

[109] L. Zhen, H. Zhengjia, Z. Yanyang, and C. Xuefeng, "Bearing condition monitoring based on shock pulse method and improved redundant lifting scheme," *Mathematics and computers in simulation*, vol. 79, no. 3, pp. 318–338, 2008.

[110] R.B. Randall and J.Antoni, "Rolling element bearing diagnostics—a tutorial," *Mechanical systems and signal processing*, vol. 25, no. 2, pp. 485–520, 2011.

[111] P. McFadden and J. Smith, "Model for the vibration produced by a single point defect in a rolling element bearing," *Journal of sound and vibration*, vol. 96, no. 1, pp. 69–82, 1984.

[112] J. K. Sinha and K. Elbhbah, "A future possibility of vibration based condition monitoring of rotating machines," *Mechanical Systems and Signal Processing* vol. 34, no. 1-2, pp. 231–240, 2013.

[113] E. Cabal-Yepez, A. G. Garcia-Ramirez, R. J. Romero-Troncoso, A. GarciaPerez, and R. A. Osornio-Rios, "Reconfigurable monitoring system for timefrequency analysis on industrial equipment through stft and dwt," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 2, pp. 760–771, 2012.

[114] Q. Miao and V. Makis, "Condition monitoring and classification of rotating machinery using wavelets and hidden markov models," *Mechanical systems and signal processing*, vol. 21, no. 2, pp. 840–855, 2007.

[115] E. C. Lau and H. Ngan, “Detection of motor bearing outer raceway defect by wavelet packet transformed motor current signature analysis,” *IEEE Transactions on Instrumentation and measurement,* vol. 59, no. 10, pp. 2683–2690, 2010.

[116] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, “The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis,” *Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences,* vol. 454, no. 1971, pp. 903–995, 1998.

[117] R. Yan and R. X. Gao, “Hilbert–huang transform-based vibration signal analysis for machine health monitoring,” *IEEE Transactions on Instrumentation and measurement*, vol. 55, no. 6, pp. 2320–2329, 2006.

[118] J. Tian, C. Morillo, M. H. Azarian, and M. Pecht, “Motor bearing fault detection using spectral kurtosis-based feature extraction coupled with k-nearest neighbor distance analysis,” *IEEE Transactions on Industrial Electronics,* vol. 63, no. 3, pp. 1793–1803, 2015.

[119] S. Saponara, L. Fanucci, F. Bernardo, and A. Falciani, “Predictive diagnosis of high-power transformer faults by networking vibration measuring nodes with integrated signal processing,” *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 8, pp. 1749–1760, 2016.

[120] M. D. Prieto, G. Cirrincione, A. G. Espinosa, J. A. Ortega, and H. Henao, “Bearing fault detection by a novel condition-monitoring scheme based on statistical-time features and neural networks,” *IEEE Transactions on Industrial Electronics,* vol. 60, no. 8, pp. 3398–3407, 2012.

[121] X. Yang, R. Yan, and R. X. Gao, “Induction motor fault diagnosis based on ensemble classifiers,” in *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*. IEEE, 2016, pp. 1–5.

[122] D. He, R. Li, and J. Zhu, “Plastic bearing fault diagnosis based on a two-step data mining approach,” *IEEE Transactions on Industrial Electronics,* vol. 60, no. 8, pp. 3429–3440, 2012.

[123] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[124] A. Malhi and R. X. Gao, "PCA-based feature selection scheme for machine defect classification," *IEEE Transactions on Instrumentation and Measurement*, vol. 53, no. 6, pp. 1517–1525, 2004.

[125] J. Wang, J. Xie, R. Zhao, L. Zhang, and L. Duan, "Multisensory fusion based virtual tool wear sensing for ubiquitous manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 45, pp. 47–58, 2017.

[126] M. Fuente, G. Garcia, and G. Sainz, "Fault diagnosis in a plant using fisher discriminant analysis," in *2008 16th Mediterranean Conference on Control and Automation*. IEEE, 2008, pp. 53–58.

[127] Y. Lei, Z. He, Y. Zi, and X. Chen, "New clustering algorithm-based fault diagnosis using compensation distance evaluation technique," *Mechanical Systems and Signal Processing*, vol. 22, no. 2, pp. 419–435, 2008.

[128] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[129] N. Le Roux and Y. Bengio, "Representational power of restricted boltzmann machines and deep belief networks," *Neural computation*, vol. 20, no. 6, pp. 1631–1649, 2008.

[130] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[131] O. Janssens, V. Slavkovikj, B. Vervisch, K. Stockman, M. Loccufier, S. Verstockt, R. Van de Walle, and S. Van Hoecke, "Convolutional neural network based fault detection for rotating machinery," *Journal of Sound and Vibration*, vol. 377, pp. 331–345, 2016.

[132] C. Lu, Z. Wang, and B. Zhou, "Intelligent fault diagnosis of rolling bearing using hierarchical convolutional network based health state classification," *Advanced Engineering Informatics*, vol. 32, pp. 139–151, 2017.

[133] X. Guo, L. Chen, and C. Shen, "Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis," *Measurement*, vol. 93, pp. 490–502, 2016.

[134] D. Verstraete, A. Ferrada, E. L. Droguett, V. Meruane, and M. Modarres, "Deep learning enabled fault diagnosis using time-frequency image analysis of rolling element bearings," *Shock and Vibration*, vol. 2017, 2017.

[135] Z. Chen, C. Li, and R.-V. Sanchez, "Gearbox fault identification and classification with convolutional neural networks," *Shock and Vibration*, vol. 2015, 2015.

[136] P. Wang, R. Yan, R. X. Gao *et al.*, "Virtualization and deep recognition for system fault classification," *Journal of Manufacturing Systems*, vol. 44, pp. 310–316, 2017.

[137] H. Dong, L. Yang, and H. Li, "Small fault diagnosis of front-end speed controlled wind generator based on deep learning," *WSEAS Trans. Circuits Syst*, vol. 15, no. 9, p. 64, 2016.

[138] J.Wang, J.Zhuang, L.Duan, and W.Cheng, "A multi-scale convolution neural network for featureless fault diagnosis," in *2016 International Symposium on Flexible Automation (ISFA)*. IEEE, 2016, pp. 65–70.

[139] P.Tamilselvan and P.Wang, "Failure diagnosis using deep belief learning based health state classification," *Reliability Engineering & System Safety*, vol. 115, pp. 124–135, 2013.

[140] F. AlThobiani, A. Ball *et al.*, "An approach to fault diagnosis of reciprocating compressor valves using teager–kaiser energy operator and deep belief networks," *Expert Systems with Applications*, vol. 41, no. 9, pp. 4113–4122, 2014.

[141] H.Shao, H.Jiang, X.Zhang, and M.Niu, "Rolling bearing fault diagnosis using an optimization deep belief network," *Measurement Science and Technology*, vol. 26, no. 11, p. 115002, 2015.

[142] M. Gan, C. Wang *et al.*, "Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings," *Mechanical Systems and Signal Processing*, vol. 72, pp. 92– 104, 2016.

[143] J. Yin and W. Zhao, "Fault diagnosis network design for vehicle on-board equipments of high-speed railway: A deep learning approach," *Engineering Applications of Artificial Intelligence*, vol. 56, pp. 250–259, 2016.

[144] H. Xie, Y. Yang, H. Wang, T. Li, and W. Jin, “Fault diagnosis in high-speed train running gears with improved deep belief networks,” *J Comput Inf Syst*, vol. 11, no. 24, pp. 7723–30, 2015.

[145] C. Li, R.-V. Sanchez, G. Zurita, M. Cerrada, D. Cabrera, and R. E. V´asquez, “Gearbox fault diagnosis based on deep random forest fusion of acoustic and vibratory signals,” *Mechanical Systems and Signal Processing*, vol. 76, pp. 283–293, 2016.

[146] Z.-X. Yang, X.-B. Wang, and J.-H. Zhong, “Representational learning for fault diagnosis of wind turbine equipment: A multi-layered extreme learning machines approach,” *Energies*, vol. 9, no. 6, p. 379, 2016.

[147] S. Haidong, J. Hongkai, L. Xingqiu, and W. Shuaipeng, “Intelligent fault diagnosis of rolling bearing using deep wavelet auto-encoder with extreme learning machine,” *Knowledge-Based Systems*, vol. 140, pp. 1–14, 2018.

[148] H. Shao, H. Jiang, Y. Lin, and X. Li, “A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders,” *Mechanical Systems and Signal Processing*, vol. 102, pp. 278–297, 2018.

[149] J. Guo, X. Xie, R. Bie, and L. Sun, “Structural health monitoring by using a sparse coding-based deep learning algorithm with wireless sensor networks,” *Personal and ubiquitous computing*, vol. 18, no. 8, pp. 1977–1987, 2014.

[150] W. Sun, S. Shao, R. Zhao, R. Yan, X. Zhang, and X. Chen, “A sparse autoencoder-based deep neural network approach for induction motor faults classification,” *Measurement*, vol. 89, pp. 171–178, 2016.

[151] C. Lu, Z.-Y. Wang, W.-L. Qin, and J. Ma, “Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification,” *Signal Processing*, vol. 130, pp. 377–388, 2017.

[152] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.” *Journal of machine learning research*, vol. 11, no. 12, 2010.

[153] H.Shao, H.Jiang, F.Wang, and H.Zhao, “An enhancement deep feature fusion method for rotating machinery fault diagnosis,” *Knowledge-Based Systems*, vol. 119, pp. 200–220, 2017.

[154] C. Li, R.-V. Sanchez, G. Zurita, M. Cerrada, D. Cabrera, and R. E. V´asquez, “Multimodal deep support vector classification with homologous features and its application to gearbox fault diagnosis,” *Neurocomputing*, vol. 168, pp. 119– 127, 2015.

[155] H. Hu, B. Tang, X. Gong, W. Wei, and H. Wang, “Intelligent fault diagnosis of the high-speed train with big data based on deep neural networks,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2106–2116, 2017.

[156] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, “Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data,” *Mechanical Systems and Signal Processing*, vol.72, pp. 303–315, 2016.

[157] M. Ma, C. Sun, and X. Chen, “Deep coupling autoencoder for fault diagnosis with multimodal sensory data,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 1137–1145, 2018.

[158] T. Han, C. Liu, W. Yang, and D. Jiang, “A novel adversarial learning framework in deep convolutional neural network for intelligent diagnosis of mechanical faults,” *Knowledge-Based Systems*, vol. 165, pp. 474–487, 2019.

[159] J. Lei, C. Liu, and D. Jiang, “Fault diagnosis of wind turbine based on long short-term memory networks,” *Renewable energy*, vol. 133, pp. 422–432, 2019.

[160] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[161] W. Lu, B. Liang, Y. Cheng, D. Meng, J. Yang, and T. Zhang, “Deep model based domain adaptation for fault diagnosis,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 3, pp. 2296–2305, 2016.

[162] L. Wen, L. Gao, and X. Li, “A new deep transfer learning based on sparse auto-encoder for fault diagnosis,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 1, pp. 136–144, 2017.

[163] X. Li, W. Zhang, and Q. Ding, "A robust intelligent fault diagnosis method for rolling element bearings based on deep distance metric learning," *Neurocomputing*, vol. 310, pp. 77–95, 2018.

[164] B. Yang, Y. Lei, F. Jia, and S. Xing, "An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings," *Mechanical Systems and Signal Processing*, vol. 122, pp. 692–706, 2019.

[165] D. Chelidze and J. P. Cusumano, "A dynamical systems approach to failure prognosis," *J. Vib. Acoust.*, vol. 126, no. 1, pp. 2–8, 2004.

[166] D. He, R. Li, and E. Bechhoefer, "Stochastic modeling of damage physics for mechanical component prognostics using condition indicators," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 48999, 2009, pp. 1259–1264.

[167] J. Luo, K. R. Pattipati, L. Qiao, and S. Chigusa, "Model-based prognostic techniques applied to a suspension system," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 38, no. 5, pp. 1156– 1168, 2008.

[168] L. Liao, "Discovering prognostic features using genetic programming in remaining useful life prediction," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 5, pp. 2464–2472, 2013.

[169] N. Z. Gebraeel, M. A. Lawley, R. Li, and J. K. Ryan, "Residual-life distributions from component degradation signals: A bayesian approach," *IiE Transactions*, vol. 37, no. 6, pp. 543–557, 2005.

[170] N. Gebraeel, "Sensory-updated residual life distributions for components with exponential degradation patterns," *IEEE Transactions on Automation Science and Engineering*, vol. 3, no. 4, pp. 382–393, 2006.

[171] A. H. Elwany and N. Z. Gebraeel, "Sensor-driven prognostic models for equipment replacement and spare parts inventory," *IIe Transactions*, vol. 40, no. 7, pp. 629–639, 2008.

[172] M.-Y. You, F. Liu, W. Wang, and G. Meng, "Statistically planned and individually improved predictive maintenance management for continuously monitored degrading systems," *IEEE Transactions on Reliability*, vol. 59, no. 4, pp. 744–753, 2010.

[173] X.-S. Si, W. Wang, M.-Y. Chen, C.-H. Hu, and D.-H. Zhou, "A degradation path-dependent approach for remaining useful life estimation with an exact and closed-form solution," *European Journal of Operational Research*, vol. 226, no. 1, pp. 53–66, 2013.

[174] N. Li, Y. Lei, J. Lin, and S. X. Ding, "An improved exponential model for predicting remaining useful life of rolling element bearings," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 12, pp. 7762–7773, 2015.

[175] D. M. D'Addona, A. S. Ullah, and D. Matarazzo, "Tool-wear prediction and pattern-recognition using artificial neural network and dna-based computing," *Journal of Intelligent Manufacturing*, vol. 28, no. 6, pp. 1285–1301, 2017.

[176] J. Yu, S. Liang, D. Tang, and H. Liu, "A weighted hidden markov model approach for continuous-state tool wear monitoring and tool life prediction," *The International Journal of Advanced Manufacturing Technology*, vol. 91, no. 1-4, pp. 201–211, 2017.

[177] D.Kong, Y.Chen, and N.Li, "Gaussian process regression for tool wear prediction," *Mechanical systems and signal processing*, vol. 104, pp. 556–574, 2018.

[178] C. Zhang and H. Zhang, "Modelling and prediction of tool wear using ls-svm in milling operation," *International Journal of Computer Integrated Manufacturing*, vol. 29, no. 1, pp. 76–91, 2016.

[179] J. Yu, "Tool condition prognostics using logistic regression with penalization and manifold regularization," *Applied Soft Computing*, vol. 64, pp. 454–467, 2018.

[180] N. Gebraeel, M. Lawley, R. Liu, and V. Parmeshwaran, "Residual life predictions from vibration-based degradation signals: a neural network approach," *IEEE Transactions on industrial electronics*, vol. 51, no. 3, pp. 694–700, 2004.

[181] R. Huang, L. Xi, X. Li, C. R. Liu, H. Qiu, and J. Lee, "Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods," *Mechanical systems and signal processing*, vol. 21, no. 1, pp. 193–207, 2007.

[182] A. Soualhi, H. Razik, G. Clerc, and D. D. Doan, "Prognosis of bearing failures using hidden markov models and the adaptive neuro-fuzzy inference system," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 6, pp. 2864–2874, 2013.

[183] C. Chen, B. Zhang, G. Vachtsevanos, and M. Orchard, “Machine condition prediction based on adaptive neuro–fuzzy and high-order particle filtering,” *IEEE Transactions on Industrial Electronics*, vol. 58, no. 9, pp. 4353–4364, 2010.

[184] J. Liu, W. Wang, and F. Golnaraghi, “A multi-step predictor with a variable input pattern for system state forecasting,” *Mechanical Systems and Signal Processing*, vol. 23, no. 5, pp. 1586–1599, 2009.

[185] Y. Wang, Y. Peng, Y. Zi, X. Jin, and K.-L. Tsui, “A two-stage data-driven based prognostic approach for bearing degradation problem,” *IEEE Transactions on Industrial Informatics*, vol. 12, no. 3, pp. 924–932, 2016.

[186] R. K. Singleton, E. G. Strangas, and S. Aviyente, “Extended Kalman filtering for remaining-useful-life estimation of bearings,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1781–1790, 2014.

[187] C. K. R. Lim and D. Mba, “Switching Kalman filter for failure prognostic,” *Mechanical Systems and Signal Processing*, vol. 52, pp. 426–435, 2015.

[188] A. Malhi, R. Yan, and R. X. Gao, “Prognosis of defect propagation based on recurrent neural networks,” *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 3, pp. 703–711, 2011.

[189] L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin, “A recurrent neural network based health indicator for remaining useful life prediction of bearings,” *Neurocomputing*, vol. 240, pp. 98–109, 2017.

[190] J.Zhang, P.Wang, R.Yan, and R.X. Gao, “Deep learning for improved system remaining life prediction,” *Procedia CIRP*, vol. 72, pp. 1033–1038, 2018.

[191] Y.Wu, M.Yuan, S.Dong, L.Lin, and Y.Liu, “Remaining useful life estimation of engineered systems using vanilla LSTM neural networks,” *Neurocomputing*, vol. 275, pp. 167–179, 2018.

[192] A.Z. Hinchi and M.Tkiouat, “Rolling element bearing remaining useful life estimation based on a convolutional long-short-term memory network,” *Procedia Computer Science*, vol. 127, pp. 123–132, 2018.

[193] R. Zhao, R. Yan, J. Wang, and K. Mao, “Learning to monitor machine health with convolutional bi-directional LSTM networks,” *Sensors*, vol. 17, no. 2, p. 273, 2017.

[194] R. Zhao, D. Wang, R. Yan, K. Mao, F. Shen, and J. Wang, “Machine health monitoring using local feature-based gated recurrent unit networks,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 2, pp. 1539–1548, 2017.

[195] S. Haidong, C. Junsheng, J. Hongkai, Y. Yu, and W. Zhantao, “Enhanced deep gated recurrent unit and complex wavelet packet energy moment entropy for early fault prognosis of bearing,” *Knowledge-Based Systems*, vol. 188, p. 105022, 2020.

[196] J. Zhu, N. Chen, and W. Peng, “Estimation of bearing remaining useful life based on multiscale convolutional neural network,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 4, pp. 3208–3216, 2018.

[197] M. Xia, T. Li, T. Shu, J. Wan, C. W. De Silva, and Z. Wang, “A two-stage approach for the remaining useful life prediction of bearings using deep neural networks,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3703–3711, 2018.

[198] L. Ren, Y. Sun, J. Cui, and L. Zhang, “Bearing remaining useful life prediction based on deep autoencoder and deep neural networks,” *Journal of Manufacturing Systems*, vol. 48, pp. 71–77, 2018.

[199] M. E. Orchard and G. J. Vachtsevanos, “A particle-filtering approach for online fault diagnosis and failure prognosis,” *Transactions of the Institute of Measurement and Control*, vol. 31, no. 3-4, pp. 221–246, 2009.

[200] F. Cheng, L. Qu, and W. Qiao, “Fault prognosis and remaining useful life prediction of wind turbine gearboxes using current signal analysis,” *IEEE Transactions on Sustainable Energy*, vol. 9, no. 1, pp. 157–167, 2017.

[201] J. Deutsch, M. He, and D. He, “Remaining useful life prediction of hybrid ceramic bearings using an integrated deep learning and particle filter approach,” *Applied Sciences*, vol. 7, no. 7, p. 649, 2017.

[202] S. Shao, S. McAleer, R. Yan, and P. Baldi, “Highly accurate machine fault diagnosis using deep transfer learning,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2446–2455, 2018.

[203] P. Cao, S. Zhang, and J. Tang, “Preprocessing-free gear fault diagnosis using small datasets with deep convolutional neural network-based transfer learning,” *Ieee Access*, vol. 6, pp. 26241–26253, 2018.

[204] C. Sun, M. Ma, Z. Zhao, S. Tian, R. Yan, and X. Chen, “Deep transfer learning based on sparse autoencoder for remaining useful life prediction of tool in manufacturing,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2416–2425, 2018.

[205] L. Wen, X. Li, and L. Gao, “A transfer convolutional neural network for fault diagnosis based on resnet-50,” *Neural Computing and Applications*, pp. 1–14, 2019.

[206] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, “Deep learning for smart manufacturing: Methods and applications,” *Journal of Manufacturing Systems*, vol. 48, pp. 144–156, 2018.

[207] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, “Smart factory of industry 4.0: Key technologies, application case, and challenges,” *IEEE Access*, vol. 6, pp. 6505–6519, 2017.

[208] L. Li, K. Ota, and M. Dong, “Deep learning for smart industry: Efficient manufacture inspection system with fog computing,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4665–4673, 2018.

[209] K.-D. Thoben, S. Wiesner, and T. Wuest, ““Industry 4.0” and smart manufacturing-a review of research issues and application examples,” *International journal of automation technology*, vol. 11, no. 1, pp. 4–16, 2017.

[210] P. Zheng, Z. Sang, R. Y. Zhong, Y. Liu, C. Liu, K. Mubarok, S. Yu, X. Xu *et al.*, “Smart manufacturing systems for industry 4.0: Conceptual framework, scenarios, and future perspectives,” *Frontiers of Mechanical Engineering*, vol. 13, no. 2, pp. 137–150, 2018.

[211] L.D. Xu, E.L. Xu, and L.Li, “Industry 4.0: state of the art and future trends,” *International Journal of Production Research*, vol. 56, no. 8, pp. 2941–2962, 2018.

[212] A. G. Frank, L. S. Dalenogare, and N. F. Ayala, “Industry 4.0 technologies: Implementation patterns in manufacturing companies,” *International Journal of Production Economics*, vol. 210, pp. 15–26, 2019.

[213] M. T. Okano, “IoT and industry 4.0: the industrial new revolution,” in *International Conference on Management and Information Systems*, vol. 25, 2017, p. 26.

[214] M. Wollschlaeger, T. Sauter, and J. Jasperneite, “The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0,” *IEEE industrial electronics magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[215] T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, and G. P. Li, “Predictive maintenance in the industry 4.0: A systematic literature review,” *Computers & Industrial Engineering*, p. 106889, 2020.

[216] T. P. Carvalho, F. A. Soares, R. Vita, R. d. P. Francisco, J. P. Basto, and S. G. Alcal´a, “A systematic literature review of machine learning methods applied to predictive maintenance,” *Computers & Industrial Engineering*, vol. 137, p. 106024, 2019.

[217] M. Canizo, A. Conde, S. Charramendieta, R. Minon, R. G. Cid-Fuentes, and E. Onieva, “Implementation of a large-scale platform for cyber-physical system real-time monitoring,” *IEEE Access*, vol. 7, pp. 52455–52466, 2019.

[218] H. Lee, “Framework and development of fault detection classification using iot device and cloud environment,” *Journal of Manufacturing Systems*, vol. 43, pp. 257–270, 2017.

[219] “Intel' Internet of Things (IoT) developer kit: IoT cloud-based analytics user guide, pp. 1–21, 2014.”

[220] P. M. Kumar, S. Lokesh, R. Varatharajan, G. C. Babu, and P. Parthasarathy, “Cloud and IoT based disease prediction and diagnosis system for healthcare using fuzzy neural classifier,” *Future Generation Computer Systems*, vol. 86, pp. 527–534, 2018.

[221] K. Wang and Y. Wang, “How ai affects the future predictive maintenance: a primer of deep learning,” in *International Workshop of Advanced Manufacturing and Automation*. Springer, 2017, pp. 1–9.

[222] M. Hermann, T. Pentek, and B. Otto, “Design principles for industrie 4.0 scenarios,” in *2016 49th Hawaii international conference on system sciences (HICSS)*. IEEE, 2016, pp. 3928–3937.

[223] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, “Industry 4.0,” *Business & information systems engineering*, vol. 6, no. 4, pp. 239–242, 2014.

[224] R. Drath and A. Horch, “Industrie 4.0: Hit or hype?[industry forum],” *IEEE industrial electronics magazine*, vol. 8, no. 2, pp. 56–58, 2014.

[225] R.Galliers, *Information systems research: Issues, methods and practical guidelines*. Blackwell Scientific, 1992.

[226] J. F. Nunamaker Jr, M. Chen, and T. D. Purdin, “Systems development in information systems research,” *Journal of management information systems*, vol. 7, no. 3, pp. 89–106, 1990.

[227] “http://dataawards.org/.”

[228] D. Han, J. Zhang, Y. Zhang, and W. Gu, “Convergence of sensor networks/internet of things and power grid information network at aggregation layer,” in *2010 International Conference on Power System Technology*. IEEE, 2010, pp. 1–6.

[229] H.-C. Hsieh and C.-H. Lai, “Internet of things architecture based on integrated plc and 3g communication networks,” in *2011 IEEE 17th international conference on parallel and distributed systems*. IEEE, 2011, pp. 853–856.

[230] Z. Qin, G. Denker, C. Giannelli, P. Bellavista, and N. Venkatasubramanian, “A software defined networking architecture for the internet-of-things,” in *2014 IEEE network operations and management symposium (NOMS)*. IEEE, 2014, pp. 1–9.

[231] H. Ning and Z. Wang, “Future internet of things architecture: like mankind neural system or social organization framework?” *IEEE Communications Letters*, vol. 15, no. 4, pp. 461–463, 2011.

[232] I. Lee, “The internet of things for enterprises: An ecosystem, architecture, and iot service business model,” *Internet of Things*, vol. 7, p. 100078, 2019.

[233] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications,” *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2347– 2376, 2015.

[234] I. Mashal, O. Alsaryrah, T.-Y. Chung, C.-Z. Yang, W.-H. Kuo, and D. P. Agrawal, “Choices for interaction with things on internet and underlying issues,” *Ad Hoc Networks*, vol. 28, pp. 68–90, 2015.

[235] D. Darwish, “Improved layered architecture for internet of things,” *International Journal of Computing Academic Research (IJCAR)*, vol. 4, no. 4, pp. 214–223, 2015.

[236] S. Madakam, V. Lake, V. Lake, V. Lake *et al.*, "Internet of things (iot): A literature review," *Journal of Computer and Communications*, vol. 3, no. 05, p. 164, 2015.

[237] P.Sethi and S.R. Sarangi, "Internet of things: architectures, protocols, and applications," *Journal of Electrical and Computer Engineering*, vol. 2017, 2017.

[238] K. Chopra, K. Gupta, and A. Lambora, "Future internet: The internet of things-a literature review," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*. IEEE, 2019, pp. 135–139.

[239] I. Yaqoob, E. Ahmed, I. A. T. Hashem, A. I. A. Ahmed, A. Gani, M. Imran, and M. Guizani, "Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges," *IEEE wireless communications*, vol. 24, no. 3, pp. 10–16, 2017.

[240] D. Mourtzis, E. Vlachou, and N. Milas, "Industrial big data as a result of iot adoption in manufacturing," *Procedia cirp*, vol. 55, pp. 290–295, 2016.

[241] M. R. Bashir and A. Q. Gill, "Towards an IoT big data analytics framework: smart buildings systems," in *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE, 2016, pp. 1325–1332.

[242] C. K. Lee, C. Yeung, and M. Cheng, "Research on IoT based cyber physical system for industrial big data analytics," in *2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. IEEE, 2015, pp. 1855–1859.

[243] Q. Zhang, X. Zhang, Q. Zhang, W. Shi, and H. Zhong, "Firework: Big data sharing and processing in collaborative edge environment," in *2016 Fourth IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*. IEEE, 2016, pp. 20–25.

[244] M. M. Rathore, A. Ahmad, and A. Paul, "IoT-based smart city development using big data analytical approach," in *2016 IEEE international conference on automatica (ICA-ACCA)*. IEEE, 2016, pp. 1–8.

[245] M. H. Berlian, T. E. R. Sahputra, B. J. W. Ardi, L. W. Dzatmika, A. R. A. Besari, R. W. Sudibyo, and S. Sukaridhoto, "Design and implementation of smart environment monitoring and analytics in real-time system framework based on internet of

underwater things and big data," in *2016 International Electronics Symposium (IES)*. IEEE, 2016, pp. 403–408.

[246] A.Yassine, S.Singh, M.S. Hossain, and G.Muhammad, "Iot big data analytics for smart homes with fog and cloud computing," *Future Generation Computer Systems*, vol. 91, pp. 563–573, 2019.

[247] K. Wang, Y. Wang, X. Hu, Y. Sun, D.-J. Deng, A. Vinel, and Y. Zhang, "Wireless big data computing in smart grid," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 58–64, 2017.

[248] Y. Zhang, S. Ren, Y. Liu, and S. Si, "A big data analytics architecture for cleaner manufacturing and maintenance processes of complex products," *Journal of Cleaner Production*, vol. 142, pp. 626–641, 2017.

[249] P. Rizwan, K. Suresh, and M. R. Babu, "Real-time smart traffic management system for smart cities by using internet of things and big data," in *2016 international conference on emerging technological trends (ICETT)*. IEEE, 2016, pp. 1–7.

[250] B. Ahlgren, M. Hidell, and E. C.-H. Ngai, "Internet of things for smart cities: Interoperability and open data," *IEEE Internet Computing*, vol. 20, no. 6, pp. 52–56, 2016.

[251] B. Cheng, A. Papageorgiou, F. Cirillo, and E. Kovacs, "Geelytics: Geodistributed edge analytics for large scale IoT systems based on dynamic topology," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE, 2015, pp. 565–570.

[252] H. Wang, O. L. Osen, G. Li, W. Li, H.-N. Dai, and W. Zeng, "Big data and industrial internet of things for the maritime industry in north-western Norway," in *TENCON 2015-2015 IEEE Region 10 Conference*. IEEE, 2015, pp. 1–5.

[253] M. Villari, A. Celesti, M. Fazio, and A. Puliafito, "Alljoyn lambda: An architecture for the management of smart environments in IoT," in *2014 International Conference on Smart Computing Workshops*. IEEE, 2014, pp. 9–14.

[254] Z. Ding, X. Gao, J. Xu, and H. Wu, "IoT-statisticdb: a general statistical database cluster mechanism for big data analysis in the internet of things," in *2013 IEEE International Conference on Green Computing and Communications and IEEE*

Internet of Things and IEEE Cyber, Physical and Social Computing. IEEE, 2013, pp. 535–543.

[255] C. Vuppalapati, A. Ilapakurti, and S. Kedari, "The role of big data in creating sense ehr, an integrated approach to create next generation mobile sensor and wearable data driven electronic health record (ehr)," in *2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService)*. IEEE, 2016, pp. 293–296.

[256] A. Ahmad, M. M. Rathore, A. Paul, and S. Rho, "Defining human behaviors using big data analytics in social internet of things," in *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 2016, pp. 1101–1107.

[257] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[258] S. Mumtaz, A. Alsohaily, Z. Pang, A. Rayes, K. F. Tsang, and J. Rodriguez, "Massive internet of things for industrial applications: Addressing wireless IIoT connectivity challenges and ecosystem fragmentation," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 28–33, 2017.

[259] F. Al-Turjman and S. Alturjman, "Context-sensitive access in industrial internet of things (IIoT) healthcare applications," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2736–2744, 2018.

[260] R. Chaudhary, G. S. Aujla, S. Garg, N. Kumar, and J. J. Rodrigues, "Sdnenabled multi-attribute-based secure communication for smart grid in IIoT environment," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2629-2640,2018.

[261] A. Gilchrist, *Industry 4.0: the industrial internet of things*. Springer, 2016.

[262] C.-W. Tsai, C.-F. Lai, H.-C. Chao, and A. V. Vasilakos, "Big data analytics: a survey," *Journal of Big data*, vol. 2, no. 1, pp. 1–32, 2015.

[263] Y. Chen, H. Chen, A. Gorkhali, Y. Lu, Y. Ma, and L. Li, "Big data analytics and big data science: a survey," *Journal of Management Analytics*, vol. 3, no. 1, pp. 1–42, 2016.

[264] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, “Big data analytics in intelligent transportation systems: A survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383–398, 2018.

[265] N. Mehta and A. Pandit, “Concurrence of big data analytics and healthcare: A systematic review,” *International journal of medical informatics*, vol. 114, pp. 57–65, 2018.

[266] A.-R. Al-Ali, I. A. Zualkernan, M. Rashid, R. Gupta, and M. Alikarar, “A smart home energy management system using IoT and big data analytics approach,” *IEEE Transactions on Consumer Electronics*, vol. 63, no. 4, pp. 426– 434, 2017.

[267] M. M. Rathore, A. Paul, W.-H. Hong, H. Seo, I. Awan, and S. Saeed, “Exploiting IoT and big data analytics: Defining smart digital city using real-time urban data,” *Sustainable cities and society*, vol. 40, pp. 600–610, 2018.

[268] T. Saheb and L. Izadi, “Paradigm of IoT big data analytics in the healthcare industry: A review of scientific literature and mapping of research trends,” *Telematics and Informatics*, vol. 41, pp. 70–85, 2019.

[269] M. H. ur Rehman, E. Ahmed, I. Yaqoob, I. A. T. Hashem, M. Imran, and S. Ahmad, “Big data analytics in industrial IoT using a concentric computing model,” *IEEE Communications Magazine*, vol. 56, no. 2, pp. 37–43, 2018.

[270] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. Mccauley, M. Franklin, S.Shenker, and I.Stoica, “Fast and interactive analytics over hadoop data with spark,” *Usenix Login*, vol. 37, no. 4, pp. 45–51, 2012.

[271] C. Engle, A. Lupher, R. Xin, M. Zaharia, M. J. Franklin, S. Shenker, and I. Stoica, “Shark: fast data analysis using coarse-grained distributed memory,” in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, 2012, pp. 689–692.

[272] H. Cai, B. Xu, L. Jiang, and A. V. Vasilakos, “Iot-based big data storage systems in cloud computing: perspectives and challenges,” *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 75–87, 2016.

[273] A. R. Biswas and R. Giaffreda, “IoT and cloud convergence: Opportunities and challenges,” in *2014 IEEE World Forum on Internet of Things (WF-IoT)*. IEEE, 2014, pp. 375–376.

[274] M. D´ıaz, C. Mart´ın, and B. Rubio, “State-of-the-art, challenges, and open issues in the integration of internet of things and cloud computing,” *Journal of Network and Computer applications*, vol. 67, pp. 99–117, 2016.

[275] J. Zhou, T. Leppanen, E. Harjula, M. Ylianttila, T. Ojala, C. Yu, H. Jin, and L. T. Yang, “Cloudthings: A common architecture for integrating the internet of things with cloud computing,” in *Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE, 2013, pp. 651–657.

[276] M.Aazam, E.-N. Huh, M.St-Hilaire, C.-H. Lung, and I.Lambadaris, “Cloud of things: integration of IoT with cloud computing,” in *Robots and sensor clouds*. Springer, 2016, pp. 77–94.

[277] H. F. Atlam, A. Alenezi, A. Alharthi, R. J. Walters, and G. B. Wills, “Integration of cloud computing with internet of things: challenges and open issues,” in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2017, pp. 670–675.

[278] A. Botta, W. De Donato, V. Persico, and A. Pescap´e, “Integration of cloud computing and internet of things: a survey,” *Future generation computer systems*, vol. 56, pp. 684–700, 2016.

[279] C. Doukas and I. Maglogiannis, “Bringing IoT and cloud computing towards pervasive healthcare,” in *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. IEEE, 2012, pp. 922– 926.

[280] W. He, G. Yan, and L. Da Xu, “Developing vehicular data cloud services in the IoT environment,” *IEEE transactions on industrial informatics*, vol. 10, no. 2, pp. 1587–1595, 2014.

[281] J. G. Ibanez, S. Zeadally, and J. Contreras-Castillo, “Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and

internet of things technologies," *IEEE Wireless Communications*, vol. 6, no. 22, pp. 122–128, 2015.

[282] M. S. Mekala and P. Viswanathan, "A survey: Smart agriculture IoT with cloud computing," in *2017 international conference on microelectronic devices, circuits and systems (ICMDCS)*. IEEE, 2017, pp. 1–7.

[283] V. Patil, K. Al-Gaadi, D. Biradar, and M. Rangaswamy, "Internet of things (IoT) and cloud computing for agriculture: An overview," *Proceedings of agroinformatics and precision agriculture (AIPA 2012), India*, pp. 292–296, 2012.

[284] S. Tyagi, A. Agarwal, and P. Maheshwari, "A conceptual framework for IoT-based healthcare system using cloud computing," in *2016 6th International Conference-Cloud System and Big Data Engineering (Confluence)*. IEEE, 2016, pp. 503–507.

[285] L. M. Dang, M. Piran, D. Han, K. Min, H. Moon *et al.*, "A survey on internet of things and cloud computing for healthcare," *Electronics*, vol. 8, no. 7, p. 768, 2019.

[286] J. Yang, C. Wang, Q. Zhao, B. Jiang, Z. Lv, and A. K. Sangaiah, "Marine surveying and mapping system based on cloud computing and internet of things," *Future Generation Computer Systems*, vol. 85, pp. 39–50, 2018.

[287] M. Schleipen, "Opc ua supporting the automated engineering of production monitoring and control systems," in *2008 IEEE International Conference on Emerging Technologies and Factory Automation*. IEEE, 2008, pp. 640–647.

[288] Y. Lu, I. Cohen, X. S. Zhou, and Q. Tian, "Feature selection using principal feature analysis," in *Proceedings of the 15th ACM international conference on Multimedia*, 2007, pp. 301–304.

[289] Y. Liu, T. Dillon, W. Yu, W. Rahayu, and F. Mostafa, "Noise removal in the presence of significant anomalies for industrial IoT sensor data in manufacturing," *IEEE Internet of Things Journal*, 2020.

[290] N. Amani, M. Fathi, and M. Dehghan, "A case-based reasoning method for alarm filtering and correlation in telecommunication networks," in *Canadian Conference on Electrical and Computer Engineering, 2005*. IEEE, 2005, pp. 2182–2186.

[291] L. H. Chiang, E. L. Russell, and R. D. Braatz, *Fault detection and diagnosis in industrial systems*. Springer Science & Business Media, 2000.

[292] P. Pace, G. Aloi, R. Gravina, G. Caliciuri, G. Fortino, and A. Liotta, "An edge-based architecture to support efficient applications for healthcare industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 481– 489, 2018.

[293] E. Ahmed, A. Akhunzada, M. Whaiduzzaman, A. Gani, S. H. Ab Hamid, and R. Buyya, "Network-centric performance analysis of runtime application migration in mobile cloud computing," *Simulation Modelling Practice and Theory*, vol. 50, pp. 42–56, 2015.

[294] R. Kelly, "Internet of things data to top 1.6 zettabytes by 2022," *Campus Technology*, vol. 9, pp. 1536–1233, 2016.

[295] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE internet of things journal*, vol. 3, no. 5, pp. 637–646, 2016.

[296] C.-C. Lin and J.-W. Yang, "Cost-efficient deployment of fog computing systems at logistics centers in industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4603–4611, 2018.

[297] J. He, J. Wei, K. Chen, Z. Tang, Y. Zhou, and Y. Zhang, "Multitier fog computing with large-scale IoT data analytics for smart cities," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 677–686, 2017.

[298] J. Preden, J. Kaugerand, E. Suurjaak, S. Astapov, L. Motus, and R. Pahtma, "Data to decision: pushing situational information needs to the edge of the network," in *2015 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision*. IEEE, 2015, pp. 158– 164.

[299] L. Yin, J. Luo, and H. Luo, "Tasks scheduling and resource allocation in fog computing based on containers for smart manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4712–4721, 2018.

[300] B. Jia, H. Hu, Y. Zeng, T. Xu, and Y. Yang, "Double-matching resource allocation strategy in fog computing networks based on cost efficiency," *Journal of Communications and Networks*, vol. 20, no. 3, pp. 237–246, 2018.

[301] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, "Cost efficient resource management in fog computing supported medical cyber-physical system," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 1, pp. 108– 119, 2015.

[302] M. A. Hassan, M. Xiao, Q. Wei, and S. Chen, "Help your mobile applications with fog computing," in *2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking-Workshops (SECON Workshops)*. IEEE, 2015, pp. 1–6.

[303] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for internet of things and analytics," in *Big data and internet of things: A roadmap for smart environments*. Springer, 2014, pp. 169–186.

[304] C. T. Do, N. H. Tran, C. Pham, M. G. R. Alam, J. H. Son, and C. S. Hong, "A proximal algorithm for joint resource allocation and minimizing carbon footprint in geo-distributed fog computing," in *2015 International Conference on Information Networking (ICOIN)*. IEEE, 2015, pp. 324–329.

[305] T. Wang, J. Zhou, X. Chen, G. Wang, A. Liu, and Y. Liu, "A three-layer privacy preserving cloud storage scheme based on computational intelligence in fog computing," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 3–12, 2018.

[306] L. Zhang and J. Li, "Enabling robust and privacy-preserving resource allocation in fog computing," *IEEE Access*, vol. 6, pp. 50384–50393, 2018.

[307] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, and M. Palaniswami, "Ppfa: Privacy preserving fog-enabled aggregation in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3733–3744, 2018.

[308] M. Chen, W. Li, Y. Hao, Y. Qian, and I. Humar, "Edge cognitive computing based smart healthcare system," *Future Generation Computer Systems*, vol. 86, pp. 403–411, 2018.

[309] S. Wu, C. Mei, H. Jin, and D. Wang, "Android unikernel: Gearing mobile code offloading towards edge computing," *Future Generation Computer Systems*, vol. 86, pp. 694–703, 2018.

[310] Q. Fan and N. Ansari, “Application aware workload allocation for edge computing-based IoT,” *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2146–2153, 2018.

[311] Y. Liu, C. Xu, Y. Zhan, Z. Liu, J. Guan, and H. Zhang, “Incentive mechanism for computation offloading using edge computing: A stackelberg game approach,” *Computer Networks*, vol. 129, pp. 399–409, 2017.

[312] W. Chen, D. Wang, and K. Li, “Multi-user multi-task computation offloading in green mobile edge cloud computing,” *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 726–738, 2018.

[313] G. Jia, G. Han, H. Xie, and J. Du, “Hybrid-lru caching for optimizing data storage and retrieval in edge computing-based wearable sensors,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1342–1351, 2018.

[314] L. Ma, X. Liu, Q. Pei, and Y. Xiang, “Privacy-preserving reputation management for edge computing enhanced mobile crowdsensing,” *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 786–799, 2018.

[315] M. Du, K. Wang, Z. Xia, and Y. Zhang, “Differential privacy preserving of training model in wireless big data with edge computing,” *IEEE Transactions on Big Data*, 2018.

[316] R. Kozik, M. Chora´s, M. Ficco, and F. Palmieri, “A scalable distributed machine learning approach for attack detection in edge computing environments,” *Journal of Parallel and Distributed Computing*, vol. 119, pp. 18–26, 2018.

[317] R. Rapuzzi and M. Repetto, “Building situational awareness for network threats in fog/edge computing: Emerging paradigms beyond the security perimeter model,” *Future Generation Computer Systems*, vol. 85, pp. 235–249, 2018.

[318] J. Cui, L. Wei, J. Zhang, Y. Xu, and H. Zhong, “An efficient message authentication scheme based on edge computing for vehicular ad hoc networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1621–1632, 2018.

[319] T. Wang, G. Zhang, A. Liu, M. Z. A. Bhuiyan, and Q. Jin, “A secure IoT service architecture with an efficient balance dynamics based on cloud and edge computing,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4831– 4843, 2018.

[320] T. Wang, L. Qiu, A. K. Sangaiah, G. Xu, and A. Liu, "Energy-efficient and trustworthy data collection protocol based on mobile fog computing in internet of things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3531–3539, 2019.

[321] J.-S. Fu, Y. Liu, H.-C. Chao, B. K. Bhargava, and Z.-J. Zhang, "Secure data storage and searching for industrial IoT by integrating fog computing and cloud computing," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4519–4528, 2018.

[322] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: Deep learning for the internet of things with edge computing," *IEEE network*, vol. 32, no. 1, pp. 96–101, 2018.

[323] X. Xu, Q. Liu, Y. Luo, K. Peng, X. Zhang, S. Meng, and L. Qi, "A computation offloading method over big data for IoT-enabled cloud-edge computing," *Future Generation Computer Systems*, vol. 95, pp. 522–533, 2019.

[324] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 63–71.

[325] W.Sun, J.Liu, and Y.Yue, "Ai-enhanced offloading in edge computing: When machine learning meets industrial IoT," *IEEE Network*, vol. 33, no. 5, pp. 68– 74, 2019.

[326] H. Shah-Mansouri and V. W. Wong, "Hierarchical fog-cloud computing for iot systems: A computation offloading game," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3246–3257, 2018.

[327] C.-F. Lai, W.-C. Chien, L. T. Yang, and W. Qiang, "Lstm and edge computing for big data feature recognition of industrial electrical equipment," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2469–2477, 2019.

[328] X. Sun and N. Ansari, "EdgeIoT: Mobile edge computing for the internet of things," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 22–29, 2016.

[329] W. Yu, T. Dillon, F. Mostafa, W. Rahayu, and Y. Liu, "A global manufacturing big data ecosystem for fault detection in predictive maintenance," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 183–192, 2019.

[330] O. Troyansky, T. Gibson, and C. Leichtweis, *QlikView Your Business: An Expert Guide to Business Discovery with QlikView and Qlik Sense*. John Wiley & Sons, 2015.

[331] X. Dai and Z. Gao, "From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2226–2238, 2013.

[332] Y. Xu, Y. Sun, J. Wan, X. Liu, and Z. Song, "Industrial big data for fault diagnosis: Taxonomy, review, and applications," *IEEE Access*, vol. 5, pp. 17368– 17380, 2017.

[333] W. Yu, T. Dillon, F. Mostafa, W. Rahayu, and Y. Liu, "Implementation of industrial cyber physical system: Challenges and solutions," in *2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*. IEEE, 2019, pp. 173–178.

[334] V.Hassija, V.Chamola, V.Saxena, D.Jain, P.Goyal, and B.Sikdar, "A survey on IoT security: application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721–82743, 2019.

[335] N.Zhang, R.Wu, S.Yuan, C.Yuan, and D.Chen, "Rav: Relay aided vectorized secure transmission in physical layer security for internet of things under active attacks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8496–8506, 2019.

[336] J. Zhang, S. Rajendran, Z. Sun, R. Woods, and L. Hanzo, "Physical layer security for the internet of things: Authentication and key generation," *IEEE Wireless Communications*, vol. 26, no. 5, pp. 92–98, 2019.

[337] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.

[338] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in internet-of-things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1250–1258, 2017.

[339] H. A. Khattak, M. A. Shah, S. Khan, I. Ali, and M. Imran, "Perception layer security in internet of things," *Future Generation Computer Systems*, vol. 100, pp. 144–164, 2019.

[340] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1727–1765, 2016.

[341] R. Li, T. Song, B. Mei, H. Li, X. Cheng, and L. Sun, “Blockchain for largescale internet of things data storage and protection,” *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 762–771, 2018.

[342] C. Stergiou, K. E. Psannis, B.-G. Kim, and B. Gupta, “Secure integration of IoT and cloud computing,” *Future Generation Computer Systems*, vol. 78, pp. 964–975, 2018.

[343] Y. Yang, X. Zheng, W. Guo, X. Liu, and V. Chang, “Privacy-preserving smart IoT-based healthcare big data storage and self-adaptive access control system,” *Information Sciences*, vol. 479, pp. 567–592, 2019.

[344] Y. Mo, “A data security storage method for IoT under hadoop cloud computing platform,” *International Journal of Wireless Information Networks*, vol. 26, no. 3, pp. 152–157, 2019.

[345] J.Zhou, Z.Cao, X.Dong, and A.V. Vasilakos, “Security and privacy for cloud-based IoT: Challenges,” *IEEE Communications Magazine*, vol. 55, no. 1, pp. 26–33, 2017.