Computers & Industrial Engineering 160 (2021) 107580

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A big data-driven root cause analysis system: Application of Machine Learning in quality problem solving

Qiuping Ma *, Hongyan Li, Anders Thorstenson

CORAL — Cluster for Operations Research, Analytics, and Logistics, Department of Economics and Business Economics, Aarhus BSS, Aarhus University, Denmark

ARTICLE INFO

Keywords:
Quality management
Data mining
Machine Learning
Multi-class classification
Neural Network

ABSTRACT

Root cause analysis for quality problem solving is critical to improve product quality performance and reduce the quality risk for manufacturers. Subjective conventional methods have been applied frequently in past decades. However, due to increasingly complex product and supply chain structures, diverse working conditions, and massive amounts of components, accuracy and efficiency of root cause analysis are progressively challenged in practice. Therefore, data-driven root cause analysis methods have attracted attention lately. In this paper, taking advantage of the availability of big operations data and the rapid development of data science, we design a big data-driven root cause analysis system utilizing Machine Learning techniques to improve the performance of root cause analysis. More specifically, we first propose a conceptual framework of the big data-driven root cause analysis system including three modules of Problem Identification, Root Cause Identification, and Permanent Corrective Action. Furthermore, in the Problem Identification Module, we construct a unified feature-based approach to describe multiple and different types of quality problems by applying a data mining method. In the Root Cause Identification Module, we use supervised Machine Learning (classification) methods to automatically predict the root causes of multiple quality problems. Finally, we illustrate the accuracy and efficiency of the proposed system and algorithms based on actual quality data from a case company. This study contributes to the literature from the following aspects: (i) the integrated system and algorithms can be used directly to develop a computer application to manage and solve quality problems with high concurrences and complexities in any manufacturing process; (ii) a general procedure and method are provided to formulate and describe a large quantity and different types of quality problems; (iii) compared with traditional methods, it is demonstrated using real case data that manufacturing companies can save significant time and cost with our proposed data-driven root cause analysis system; (iv) this study not only aims at improving the quality problem solving practices for a complex manufacturing process but also bridges a gap between the theoretical development of Machining Learning methods and their application in the operations management domain.

1. Introduction

One of the most critical factors to increase revenue and retain brand reputation is high product quality (Hanaysha, Hilman, & Abdul-Ghani, 2014; Xu, Dang, & Munro, 2018). Quality problems occur almost every day in almost all manufacturing processes. They cause negative outcomes, e.g., product recalls, customer churns, and production disruptions. Product recalls not only incur immediate costs but also directly damage the product reputation. Djurdjanovic, Lee, and Ni (2003) show that one-minute downtime costs up to 20,000 US dollars in the automotive industry. Thus, quality management and control is one of the most important operation management tasks in the manufacturing industry. A primary concern of this task is quality problem solving (Xu et al., 2018) because quality problems need to be solved from the root, otherwise, they may occur again.

Moreover, decades of industrial practices have proved that a structured and effective quality problem solving process can be a systematic solution for sustained quality improvement (Liang & Zhang, 2010). With efficient quality problem detection and problem solving, it is possible to reduce warranty spending, which accounts for between 2 and 6 percent of total sales in the automobile industry (WarrantyWeek, 2016). At the core of quality problem solving, root cause analysis (RCA) identifies the underlying root causes of quality deviations, ensures the right directions of further actions, and prevents quality problem recurrence. According to Mohammadnazar, Pulkkinen, and Ghanbari

* Correspondence to: Department of Economics and Business Economics, Aarhus BSS, Aarhus University, Fuglesangs Allé 4, Building 2621, B112, DK-8210 Aarhus V, Denmark.
E-mail address: qiuping.ma@econ.au.dk (Q. Ma).

https://doi.org/10.1016/j.cie.2021.107580
Received 11 November 2020; Received in revised form 3 May 2021; Accepted 23 July 2021
Available online 3 August 2021
0360-8352/

(2019), RCA is a structured investigation to identify the underlying causes of recurring faults.

RCA of quality failures and quality deviations usually rely on existing on-site quality experts using traditional RCA methods (Lokrantz, Gustavsson, & Jirstrand, 2018; Mueller, Greipel, Weber, & Schmitt, 2018). Empirical evidence shows that traditional RCA methods such as brainstorming, 5-Why, Ishikawa diagram, are still very popular and widely used in practice, due to their structured concepts and easy-to-use. However, the quality problems are often very complicated because of the high complexity of the products and supply chains in high-tech manufacturing industries. Additionally, a large number of quality problems cause heavy workloads to experts and prevent them from spending plenty of time on individual problems. Therefore, new RCA methods and systems are required in order to solve the quality problems more efficiently and effectively.

With a wide range of applications of advanced information technologies and the internet of things, a vast amount of data has been produced and accumulated since the early 21st century in companies. Huge amounts of data have been generated and collected along the supply chain and over the whole product life cycle. The big data era has arrived (Brown, Chui, & Manyika, 2011; Zhang, Ren, Liu, Sakao, & Huisingh, 2017; Zhou, Fu, & Yang, 2016). Essentially, big data have the characteristics of "4V", i.e., volume, velocity, variety and value. Big data analytics and advances in Machine Learning (ML) make big data-driven analysis regarding RCA possible (Lokrantz et al., 2018). However, on the one hand, research on data-driven quality management is scarce. Nguyen, Zhou, Spiegler, Ieromonachou, and Lin (2018) conduct a survey on big data applications within supply chain management context and point out that the use of big data analytics and ML for quality management during manufacturing process is rather limited in recent years. Especially, big data-driven RCA has not been investigated in depth before. On the other hand, there are still large research gaps between existing studies and practical requests. For instance, many studies apply ML methods based on simplified business processes and simulation data. Several studies have attempted to consider real business processes and data, but they all focus on a single data source (type) such as machining process data (quantitative), expert knowledge (qualitative), or historical quality report data (textual). The complexity of real production processes and quality problems cannot be captured. In addition, most extant literature concentrates on a specific type of quality problem and analyzes specific data. The efficiency will be an issue when dealing with multiple quality problems occurring in a short time window.

Motivated by real-world business requests in a world-class luxury automotive company, the aim of this paper is to propose a big data-driven RCA methodology and system for a complex high-tech manufacturing process so that the accuracy and efficiency of RCA are improved. To achieve the research objective, we need to investigate three main research questions: (i) what are the necessary components in a big data-driven RCA system, and how to establish an infrastructure to facilitate automated RCA? (ii) How to define, construct, and describe multiple and different types of quality problems in a systematic and structured way by using multiple types of historical data from multiple sources in the manufacturing process? (iii) How to efficiently find the root causes of multiple and different types of quality problems based on a big empirical quality dataset?

With regard to the research questions, we first construct a conceptual model for the big data-driven RCA system including three modules: Problem Identification (PI), Root Cause Identification (RCI), and Permanent Corrective Action (PCA). Then, we apply ML methods to define quality problems and identify the root causes of the quality problems. In more details, in the PI Module, we establish feature libraries from multiple data sources through three ways: directly collecting part data and supply chain data (procedure data and people data) as features from the ERP system; gathering quality experts' knowledge and experiences in the form of features through interviews; and using TF–IDF (term frequency–inverse document frequency) to extract features from textual quality reports. As a result, we can systematically define, construct, and describe multiple and different types of quality problems by using feature vectors including different features from the standard feature libraries. In the RCI Module, we further employ ML classifiers to detect the possible root causes of defined quality problems. Finally, the case example shows that our method can identify the root causes of more than 12,000 quality problems within seconds with the trained model and the accuracy rate is up to 90%.

The main contributions of this study include: (i) we propose a comprehensive system design for big data-driven RCA from raw data transmission to root cause detection of quality problems in a complex manufacturing process. The system includes a generalized conceptual framework, big data analysis methods, and machine learning algorithms. The system can be used directly to develop a computer application and completely replace traditional subjective RCA methods. Moreover, the system provides a knowledge sharing platform for quality management teams. (ii) We develop a feature-based standard model by applying a data mining method to identify and describe the complex and large quantity of quality problems. The standard feature structure not only accomplishes the systematic description of multiple and different types of quality problems in batches but also enhances a common understanding of quality problems among different users of the system in a company. (iii) The big data-driven RCA methods and ML algorithms improve the accuracy and efficiency of RCA significantly compared with conventional methods. Large quantities and multiple types of quality problems can be identified and analyzed simultaneously in a common procedure. (iv) A case study with real operations data shows that the proposed system is effective to solve the RCA problem with high frequencies and level of complexity. To the best of our knowledge, this is also the first study aiming at solving complicated RCA problems by applying advanced ML techniques.

The rest of the paper is structured as follows. Section 2 provides an overview of the literature and illustrates the research gaps. Section 3 describes the common quality problem solving process and challenges in practice. In Section 4, we design the framework of a big data-driven RCA system with application of ML techniques. Section 5 presents the methodology in the PI and RCI Modules. A case example will be used to illustrate the system and methodology in Section 6. Section 6.3 shows the result and discussion. Finally, the paper ends with a conclusion and future research directions in Section 7.

2. Literature review

Exploring the root causes of observed symptoms in complicated systems has been a major concern for decades (Solé, Muntés-Mulero, Rana, & Estrada, 2017). Traditionally, quality experts manually handle RCA using their knowledge through some methods. A wide variety of conventional RCA methods have been applied, for example, 5-Why (Latino, Latino, & Latino, 2019), Cause Mapping (York, Jin, Song, & Li, 2014), Failure Modes and Effect Analysis (FMEA) (Aurisicchio, Bracewell, & Hooey, 2016; Medina-Oliva, Iung, Barberá, Viveros, & Ruin, 2012), Fault Tree Analysis (FTA) (Leveson, 2004), etc. These approaches are still broadly applied by practitioners because of their simplicity (Wieczerniak et al., 2017). However, with the increasing complexity of quality problems and high requirements on accuracy and efficiency in the industrial 4.0 age, the traditional RCA methods are criticized because of the following drawbacks: (i) tree-like causal structures of the traditional methods confine the main focus on linear relations, resulting in limitations for non-linear interactions in quality problems (Aurisicchio et al., 2016; Yuniarto, 2012); (ii) traditional methods cannot distinguish the strengths of possible root causes (Chemweno et al., 2016); (iii) the results of RCA largely depend on a high level of expert experience and knowledge which are difficult to transfer and preserve given frequent turnover in manufacturing companies (Mueller et al., 2018); (iv) the heavy manual analysis makes the RCA a time-consuming and labor-intensive task (Xu & Dang, 2020).

In order to handle the quality problems with high complexity and quantity in modern high-tech manufacturers, data-driven RCA methods have captured the attention of both practitioners and researchers. For example, Du, Lv, and Xi (2012) derive off-line match relationships between fixture fault patterns and part variation motion patterns in the machining process of the cylinder head of the engine, and then identify the root cause of component variation in the light of on-line dimensional measurements based on Neural Network (NN). He, Zhu, He, Gu, and Cui (2017) propose a big data-oriented approach based on Axiomative domain mapping and weighted association rule mining for product infant failure. These two studies only focus on one specific type of quality problem such as fault of vibration and noise of the washing machine body. It means that the off-line match relationship and the association tree based on Axiomative domain mapping need to be reconstructed for each type of quality problem, and cannot be generalized to other quality problems. Therefore, the methods are unsuitable for large scale manufacturing processes, in which thousands of components and various quality problems are involved simultaneously.

In addition, some data-driven research is based on simulation processes or simulation data. For instance, Lokrantz et al. (2018) utilize a probabilistic graphical model with Bayesian networks to learn casual relationships. The network structure is deduced based on a simulated manufacturing process and prior knowledge. The network parameters are trained by historical data. The root cause is then inferred according to defect types and measurements. Mueller et al. (2018) introduce a decision tree algorithm on RCA within a small batch production scenario. However, the decision tree model is built on a simple linear production-related regression model and trained by a Monte Carlo simulation. Both simulated processes and simulation data simplify real business. Clear guidance and comprehensive design of application systems on how to apply the method to solve real-world problems remain unexplained.

Our study is particularly related to two recent papers that utilize ML methods on textual data from historical quality problems. Xu et al. (2018) use TF–IDF (term frequency–inverse document frequency) algorithm to mine and artificially prune to obtain critical and informative features to describe quality problems and extract relationships between quality problems and root causes. Xu and Dang (2020) adopt K-means clustering to cluster problems and causes based on textual data, and construct the cause-and-effect relationships between problem classes and cause classes. They use a single source of data: manual input text of quality reports. It is not sufficient to fully describe quality problems and detect root causes since the data from the ERP system and expert knowledge are neglected. In addition, the ML methods applied in the study are oversimplified and may cause information loss, and thus high accuracy cannot be guaranteed.

In summary, there exist clear research gaps between existing data-driven RCA studies and real-world application requests in terms of efficiency and accuracy. Most extant literature discretely defines quality problems and analyze root causes, making it difficult to realize batch RCA given a large number of quality problems, leading to low efficiency. What is more, simulation-based methods simplify the real business and thus low accuracy are reasonable to expect when applying in practice. Finally, single data sources such as process parameters, expert knowledge, or textual data limit the application and resulting in low flexibility and also low accuracy. Our study is different from the studies mentioned above in four aspects. First, we provide a conceptual design of the RCA system including three modules. Second, we consider quality-related empirical data from multiple sources: the ERP database, quality audit database, and expert knowledge, which leads to a more complete and accurate quality problem description. Third, the root causes of multiple quality problems are identified automatically by using ML techniques. At last, the system supports knowledge learning and transfer. The method and system can be easily applied to quality problem solving for a new product.

3. Problem description

Quality problem solving is an essential part of daily operations and normally complicated since there are thousands of components, hundreds of working units and workers involved in an assembly line of complex advanced products, e.g., automobiles, machine tools, equipment, and instruments. It is very difficult, if not impossible, to reach the target of "zero-defect". To retain and improve product quality, factories invest a huge amount of capital and other resources in quality problem solving.

3.1. The quantity of quality problems

The quality problem solving process is generally complex and time consuming. We use the case data to illustrate the high quantity and diversity of the quality problems. We focus on one classical car model and collect the historical quality data through its whole life cycle with over half a million vehicles produced. In Fig. 1, we first show an overview of the quantity of the quality problems over time from two statistical perspectives: the number of defect component types and the number of defects. The number of quality problems in both perspectives shows an overall growing trend over a 7-year period with 25% and 12% increase, respectively, while production volume increases by 17% roughly. Furthermore, the number of defects is much higher than the number of defect component types, since one component may have multiple defects. More importantly, the graphs show drastic fluctuations on a weekly basis. The average and peak values of the defect component types are 211 (4% of the sample data) and 423 (8.5% of the sample data). Note that the component types are not the same at different times. According to the data analysis, about 48% of the total number of components incurred quality problems over the 7 years. Therefore, the quality problems are diverse and difficult to handle manually. From the perspective of the number of defects, average and peak values are 3,400 and 8,068, respectively. It shows that the quality problems might cause significant extra costs.

Additionally, in Fig. 2 we take three months of data out from the sixth year and illustrate the daily workload. On average, 119 defect component types with 1,276 defects are found every working day while daily numbers fluctuate heavily, especially for the defects. That means a large and variable number of tasks needs to be solved in the quality department on a continuous basis. As mentioned earlier, these data only represent the quality problems of one specific product model. However, it is very common for manufacturers to assemble multiple types of product models at the same time. For example, in our case company, a single factory assembles five automobile models in the same assembly line. Thus quality problem solving is often a quantity intensive task in the manufacturing industry, and the workload shows large fluctuations. Again, if all the problems are handled by the traditional RCA approaches, a great amount of time and man power is consistently required and the accuracy of the solutions cannot be guaranteed. Therefore, a more effective and efficient quality problem solving process is practically needed.

3.2. Quality problem solving and RCA

As a result of the high quantity and diversity of quality problems, in the real production situation, the quality problem solving process is complicated and time-consuming. However, the quality problems incurred during the production periods need to be resolved as quickly as possible so that the loss and risk are minimized. In Fig. 3, we illustrate the quality problem solving process in practice with the *8D* concept, which has been proved to be a common and highly efficient problem solving methodology (Wieczerniak et al., 2017), and also applied by our case company.

First, a quality problem is detected in an assembly line or logistics handling area. The quality specialist checks whether the component

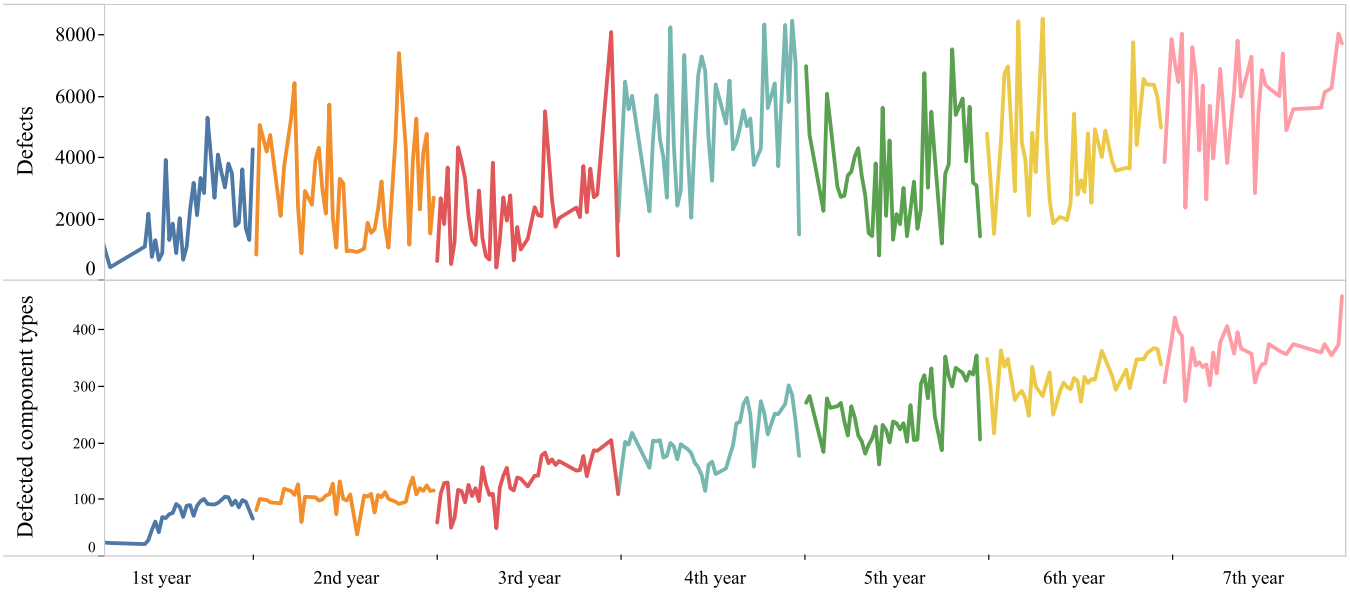


Fig. 1. 7-year overview of weekly quality problem quantities.

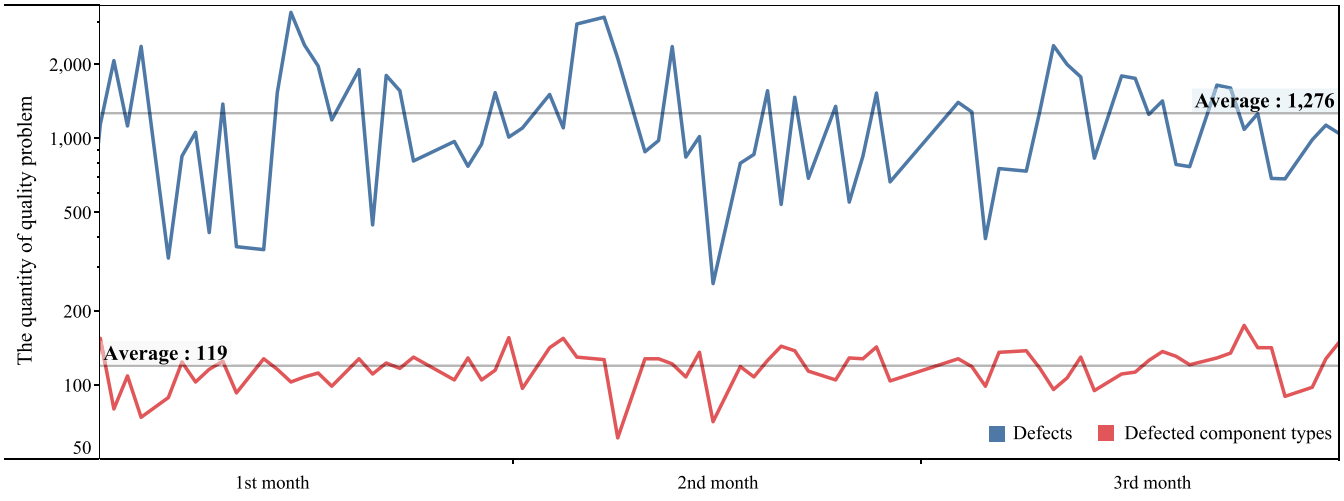


Fig. 2. Daily numbers of quality problems.

can be used or not. A component with tiny quality defects, such as inconspicuous scratches, can be used directly and the process is closed. Second, if the component cannot be used, the reporter immediately creates a quality defect incident in an embedded IT system and hands it over to the problem owner (the department that should lead the problem solving process). The problem owner checks whether it is a single case or not. A single case is not worth the time and effort to solve it, and then the process is closed. Third, problem solving becomes indispensable for a non-single case. To start with *D1*, it is necessary to invite experts from different backgrounds to build up a team, especially for complicated quality problems. Then, the experts work together in *D2* to identify what is the exact problem. In *D3*, interim containment actions or remedial measures are taken to ensure the production line is running smoothly. When the production line is safe, the expert team goes back to the problem and identifies the root cause in *D4*. On the basis of the root cause identified, in *D5* permanent corrective action is generated. Then, in *D6* the team implements and validates the action and in *D7* extends the results to similar components or similar quality problems to prevent a recurrence. Finally, in *D8* the expert team closes the problem and shares success.

The critical steps of the quality problem solving process are *D2*, *D4*, and *D5* because they are complex and time consuming. Other steps are either simple (*D1*, *D3*) or only related to executive ability (*D6*, *D7*, *D8*). Moreover, only *D2*, *D4*, and *D5* are decision making steps in which technical knowledge and intelligence are highly required. In Fig. 4, the detailed quality problem solving process for a specific quality problem is used as an example to illustrate the time consumption at each step. As we can see, the quality problem solving process lasts for about two months. A relatively long time is spent on *D4* and *D5*. The main reason is that they are the prerequisite and core of the quality problem solving process. Unskillful RCA results in problem solving delays or improper solutions (Wieczerniak et al., 2017). Experts are careful to make the final decisions since further actions will be wasted with the wrong direction. In fact, *D2*, *D4*, and *D5* are also the three steps in the broad definition of RCA suggested by Lehtinen, Mäntylä, and Vanhanen (2011), Okes (2005), and York et al. (2014).

3.3. Research motivation

Based on the above discussions about the quantity and diversity of quality problems and the quality problem solving process, we explore

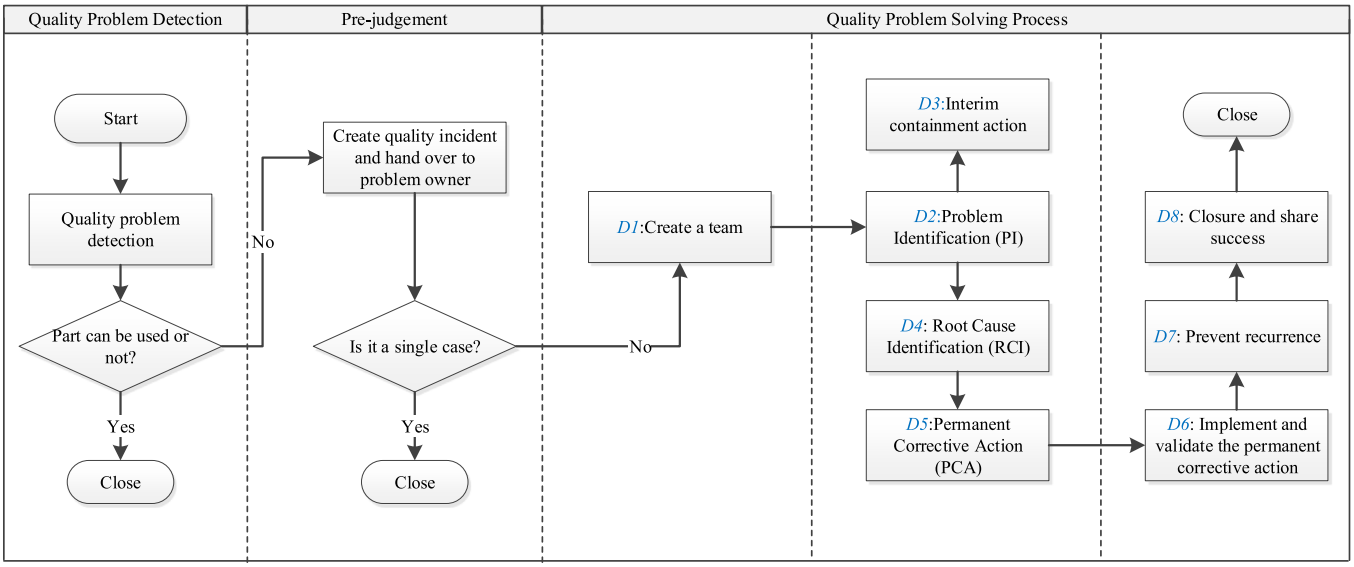


Fig. 3. The quality problem solving process in the case company.

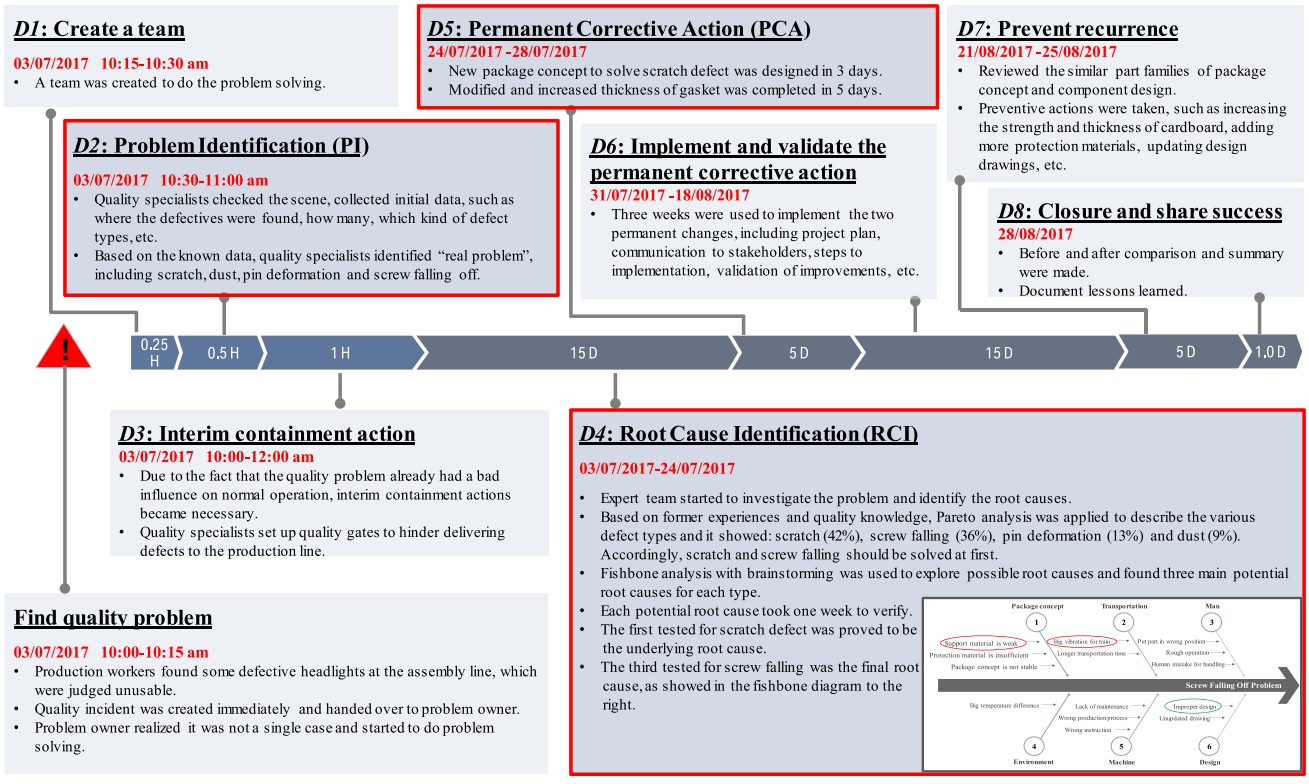


Fig. 4. An example to illustrate the current quality problem solving process in the case company.

the challenges of RCA in practice. First, most of the quality problems are manually processed by applying traditional RCA tools. Quality specialists handle the RCA according to personal expertise, experience, or intuition, making it difficult to guarantee efficiency and effectiveness. Second, different persons may obtain different results for the same quality problems. It is most likely that the personal knowledge and bias affect the correctness of the decisions significantly. Third, although big data are currently created and collected in the whole supply chain process, they are typically not used directly for the quality problem solving in practice, because of the lack of conceptual approaches and application tools. Fourth, insightful information and knowledge sharing platforms with historical quality problem solving cases are missing, not to mention their successful applications. The existing academic research studies about big data and business analytics have not been applied in daily quality problem solving processes for real operations. Finally, even if root causes can be found efficiently, selecting a final solution can still be a challenge. This is due to the fact that several departments (e.g. financial department, quality department, production department,

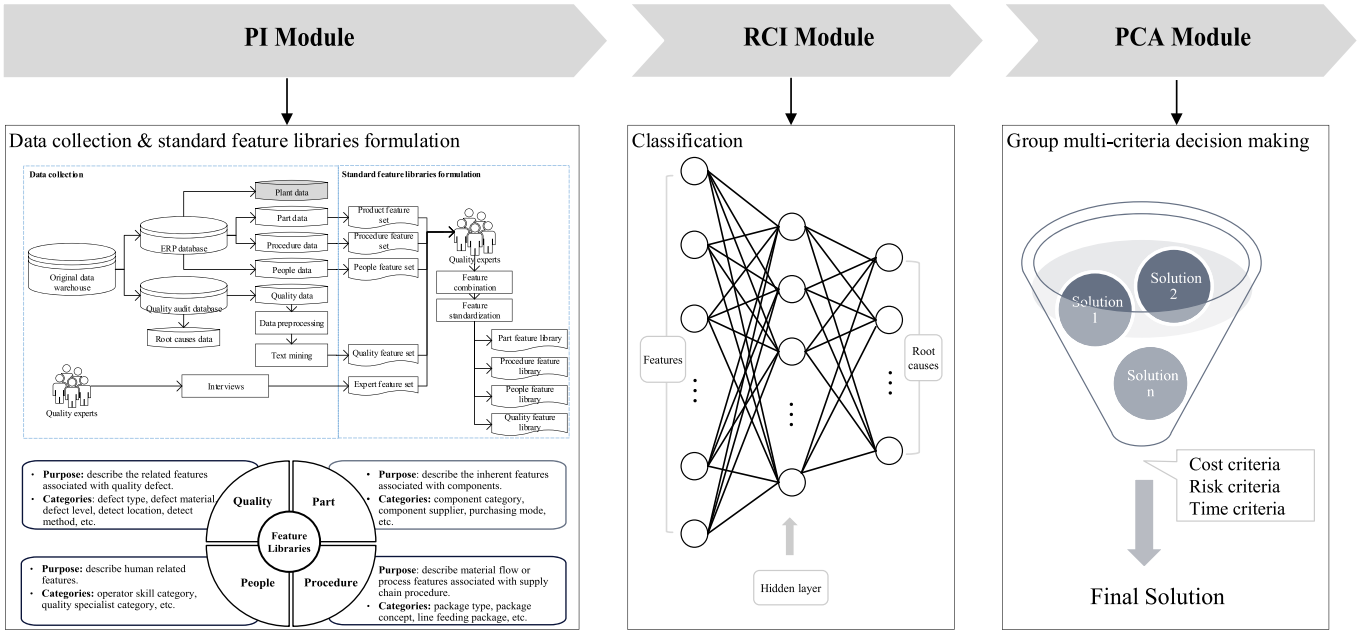


Fig. 5. Framework of the big data-driven RCA system.

root cause department) need to be involved concerning solution selection, and each makes the final decisions based on their own evaluation criteria. Therefore, it is necessary to develop an improved method for the quality problem solving process.

4. The framework of big data-driven RCA system

As previously mentioned in Section 3.2, the bottleneck of the quality problem solving process is the RCA process. However, as we mentioned in the literature review, both conventional and existing data-driven RCA methods face challenges in accuracy and efficiency. They cannot fully meet the needs of implementing RCA in a complex manufacturing environment. Therefore, we propose a big data-driven RCA system utilizing advanced ML methods.

In this section, we provide a conceptual design of the big data-driven RCA system. The broader sense of RCA includes three steps: (i) Problem Identification (PI), (ii) Root Cause Identification (RCI) and (iii) Permanent Corrective Action (PCA) (Lehtinen et al., 2011; Okes, 2005; York et al., 2014). Our big data-driven RCA system is constructed with three modules accordingly: (i) PI Module, (ii) RCI Module, and (iii) PCA Module. The structure of the system is shown in Fig. 5. Each module is then introduced in more detail.

PI Module As a starting point, this module collects relevant data and extracts features from the data to formulate standard feature libraries, so that quality problems can be described in a unified systematic manner. As the foundation of the system, we collect data from the ERP database and quality audit database. Furthermore, expert knowledge and experience can be collected through interviews. As the result, multiple initial feature sets are obtained. Then, feature combination and standardization tasks are conducted by quality management authorities in the company. Finally, four standard feature libraries from the perspectives of part, procedure, and people, quality are established. Thereby, the quality problem can always be described by a vector including various features from standard libraries.

RCI Module This module explores the relationships between features and root causes based on the advanced ML algorithms. According to the PI Module, all quality problems can be described by feature vectors in a standard way. Thus, the feature vectors will be the predictor or input of the ML models. The extracted root causes of quality problems from the historical quality database will be the target or response variables of the ML model. In the language of ML, we solve a multi-class classification problem by using the combination of features to directly predict potential root causes of quality problems. KNN and NN classifiers are suggested and tested in this paper.

PCA Module This module chooses the final quality problem solution in terms of various criteria. Multiple criteria will be considered, such as the cost criterion, risk criterion, and the time criterion. A Multi-criteria Decision Making (MCDM) model can be applied in this module to make trade-offs among various criteria. In addition, the solution selection step normally involves several functional departments in a company. It can be considered as a group decision making problem. Group Multi-criteria Decision Making (GMCDM) methods have been developed and studied for a long time. Many approaches can be directly applied to this problem environment. Related literature is provided by, e.g., Chang, Yeh, and Chang (2013), Pamučar, Mihajlović, Obradović, and Atanasković (2017), and Safarzadeh, Khansefid, and Rasti-Barzoki (2018). Given the limits of space and scope of this paper, this module will not be pursued further.

5. Methodology

Based on the framework of our big data-driven RCA system presented in Section 4, we further elaborate the methodologies used in the PI Module and RCI Module in details in this section.

5.1. PI module

The aim of the PI Module is to describe quality problems in a systematic and structured way based on the historical quality-related data so that the quality problems can be batch analyzed later in the RCI Module. The PI Module is the key factor that makes a data-driven RCA system feasible and differentiates it from the existing literature. It is an essential prerequisite for the whole study. To the best of our knowledge, our study is the first attempt to systematically describe

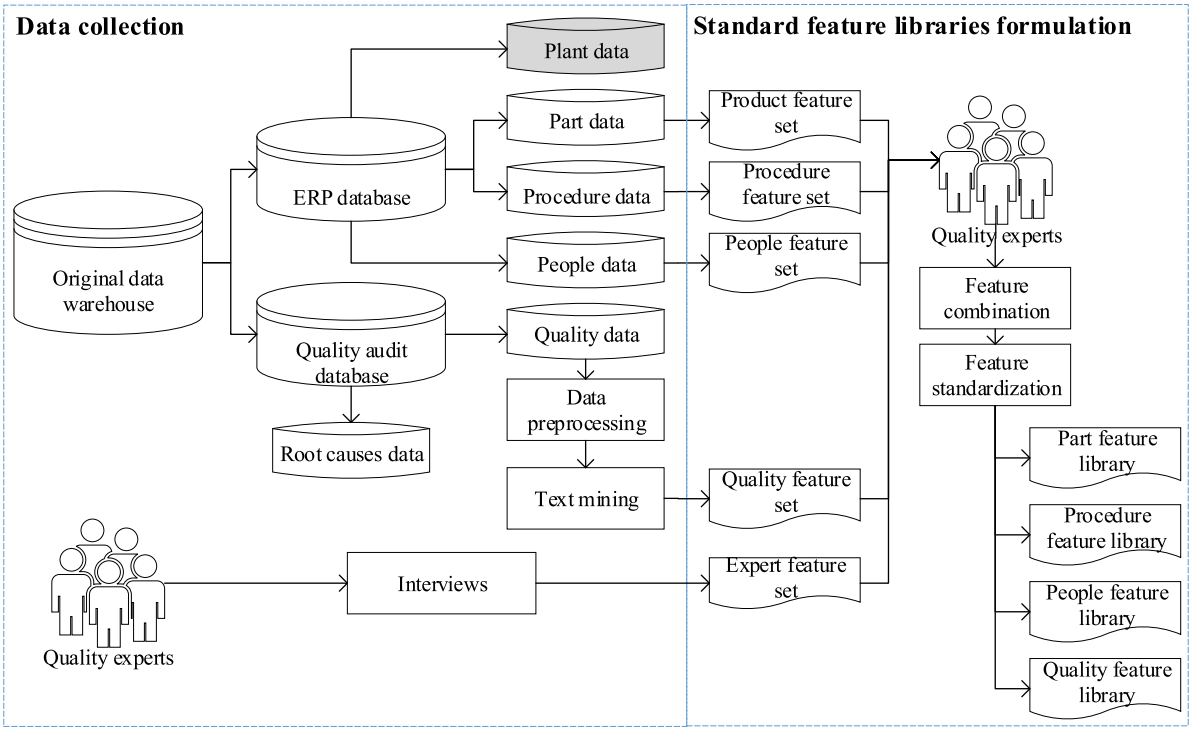


Fig. 6. Data collection and standard feature libraries formulation.

multiple and different types of quality problems based on unstructured raw big data. In this regard, the main challenge lies in the transmission from the unstructured raw big data to the structured descriptions of quality problems. In the field of ML, features of events are often extracted from big data to achieve better prediction. Inspired by that, we first transform raw big data into related features and then use the obtained features to describe quality problems. Quality problem may be characterized by multiple features. Therefore, we define a feature vector including multiple feature elements for each quality problem. In order to extract and standardize all the features of quality problems, the PI Module involves two tasks: data collection and standard feature libraries formulation shown in Fig. 6.

In the data collection stage, relevant data are collected from the original data warehouse in which the ERP database and quality audit database are involved. In the ERP database, four types of data are often relevant to the quality problems: plant (machine) data, part (raw material) data, procedure (logistics) data, and people data, called also 4Ps. The concept of 4Ps is originated from Ishikawa analysis and has been used to predefine categories of root causes (Mirsu, 2013). Due to the low level of automation in the automotive assembly workshop in the case company, the role of the equipment in quality problems is negligible. Therefore, we only focus on the part, procedure, and people data from the ERP database. Several studies have investigated quality problems for equipment or machining processes as mentioned in Section 2. If the equipment plays an important role in some quality problems, we can easily add the corresponding parameters to the feature libraries. The quality audit database provides quality data and root causes data. In addition, quality expert knowledge and experience are also considered and added to our data sources. We conduct interviews to collect the features which are not covered in the former databases.

Based on the data collected, we construct standard feature libraries. As for the data from the ERP database, they are normally formatted and structured data associated with a specific name, type, and value. In this case, the data structure forms the part feature set, procedure feature set, and people feature set. However, the data from the quality audit database are textual and semi-structured because the data are often manually entered by quality management staff. Thus, data preprocessing is required to handle incomplete, noisy, and inconsistent, redundant, and imbalanced, outlying, and duplicate data in advance. Then, we adopt the TF–IDF (term frequency–inverse document frequency) algorithm (Xu et al., 2018) to mine and artificially prune to obtain critical and informative features. As a result, features extracted from the quality audit database constitute a quality feature set. Eventually, the five feature sets mentioned above are evaluated by quality experts so that the overlapping features are removed or combined. Moreover, the remaining features from the expert feature set are divided into other four feature sets according to their attributes. At last, four standard feature libraries are formulated as shown in Fig. 7.

Furthermore, each feature library contains multiple categories and each category includes multiple features. For instance, the part feature library is used to describe the inherent features associated with the components. There are various categories in this library, such as component category, component supplier, purchasing mode. In the component category, multiple features are included, such as the exterior, interior, chassis, drivetrain, and electronics. An example of the feature library is from our case example shown in Appendix.

Once the feature libraries are constructed, a quality problem can be described by a feature vector with multiple indexed elements representing all features. The value of each element corresponding to a specific feature will be non-zero when it occurs in a quality problem. Based on the unique component ID, most feature values can be captured automatically from an ERP system or quality reports. However, a limited number of features extracted from expert knowledge may still need manual judgment given changes with respect to new components or processes. Previously, we had to use some texts to describe a quality problem as shown in Fig. 4, such as where the defects were found, how many defects, and which kind of defect types. Now we can use a vector $(0, 1, 0, 0, 5, 0, 0, 1....)$ to express the quality problem by combining features from the four feature libraries. The first "1" indicates that defects were found at an assembly line, the "5" indicates that 5 defects had been found, and the second "1" indicates the defect type was a scratch, etc.

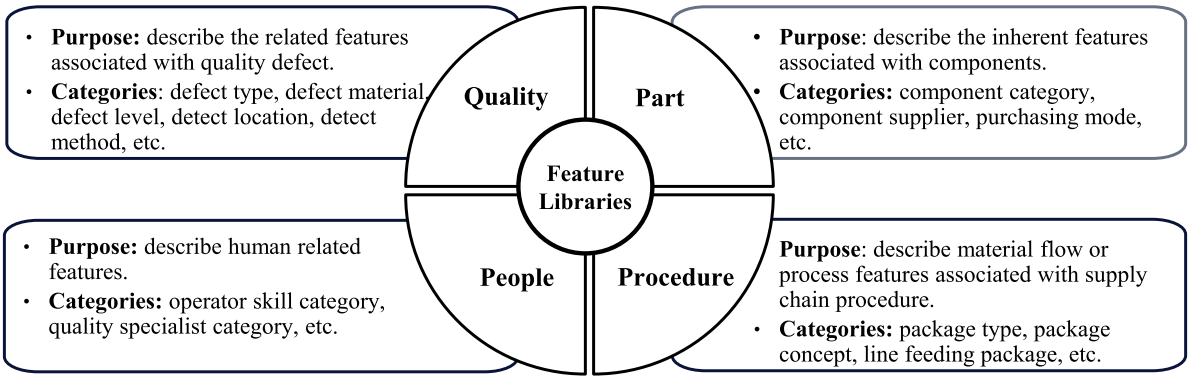


Fig. 7. Feature libraries in PI Module.

The vector described is suitable for all historical or similar quality problems. Moreover, a vector with large amounts of common and standardized features can also be applied to new quality problems. It is true that some special features of new quality problems may not be covered and need manual interference, but the rate of new features will presumably be low and not be considered at the technical level in this paper. Updating the feature libraries regularly on the basis of practical experiences can be a proper way to solve this problem.

5.2. RCI module

Based on the PI Module, we are able to describe multiple and different types of quality problems by feature vectors with standard structure. Moreover, we can extract the root causes of historical quality problems from the quality audit database. The root causes are then standardized based on the location of supply chain partners (supplier, logistics service provider, and the manufacturer) and root cause categories (machine, material, man, design, and logistics). Table 1 shows the root cause classes. We extract and standardize 41 classes in total. Based on the location of the root causes, the manufacturer has the most classes with 28, followed by the supplier with 9 and the logistics service provider with 4. According to the category of the root causes, there are 23 classes for manhandling, 5 classes for material and design, and 4 classes for machine and logistics transportation. The production and logistics area in the assembly workshop contain a lot of working units. Thus, we split the class of production handling and logistics handling into details on the basis of the unit blocks. As a result, all historical quality problems can be classified by the classes of the root causes. Therefore, it is suitable to apply the supervised ML method in the RCI Module, i.e., the classification. The classification model aims at finding a mathematical relationship between a set of input features and categorical output responses (James, Witten, Hastie, & Tibshirani, 2013). In our approach, the standard feature vectors corresponding to different quality problems are the input features and the root cause classes of the quality problems are the categorical output responses.

As shown in Section 3.1, multiple quality problems may incur at the same time or within a short time period, and therefore, we aim to predict multiple root causes regarding multiple quality problems. It means that we need to solve a multi-class classification problem. Furthermore, we propose the ML (classification) methods to predict the root causes. To facilitate the description, the notations in Table 2 are used.

A general classification model can be written as $\mathbf{z} = f(\mathbf{X})$, where $f(\cdot)$ refers to a classifier. Various classifiers are available (Ballabio, Grisoni, & Todeschini, 2018). In this article, we apply K-Nearest Neighbor (KNN) and Neural Network (NN) classifiers to predict the root causes. Of course, it is possible to apply other classifiers, but our system framework and module methodology will not be intrinsically changed. Given the space limitation, we do not intend to exhaust all the classification models.

The quality problems described by the standard feature vectors and the root cause classes compose the dataset $\mathbb{D}$. To implement the classification models, the dataset $\mathbb{D}$ is divided into three subsets:

- Training set: used to train the classification models.
- Validation set: a small part of the training set which is used to estimate the performance of classification models and tune the hyperparameters accordingly.
- Test set: used to evaluate the performance of the final model.

Before we train the models, we also suggest following sequential steps of data preprocessing and feature engineering documented in Boehmke and Greenwell (2019). The steps include (i) filter out zero or near-zero variance features; (ii) perform imputation if required; (iii) normalize to resolve numerical feature skewness; (iv) standardize (center and scale) numeric features; (v) perform dimension reduction on numeric features, as well as (vi) use one-hot or dummy encoding of categorical features. The specific technical details about how to implement the classification models will be presented in the case study in Section 6. In this section, we focus on presenting the structure and concepts of the KNN and the NN models.

5.2.1. The K-Nearest Neighbors model

KNN classifies a new observation by identifying the classes of its K-Nearest Neighbors (Imandoust & Bolandraftar, 2013). In this study, we use two popular classifiers, namely KNN with KD-trees classifier

Table 1
Standardized root cause classes.

	Supplier production	Supplier logistics	Logistics service provider	Manufacturer production	Manufacturer logistics
Machine	S01.mechanical issue S02.electrical issue	—	—	M01.mechanical issue M02.electrical issue	—
Material	S03.component material	S06.package material	L01.package material	M03.component material	M15.package material
Man	S04.production handling	S07.logistics handling	L02.logistics handling	M04–M13 production handling	M16–M25 logistics handling
Design	S05.component design	S08.package concept	L03.package concept	M14.component design	M26.package concept
Logistics	—	S09.transportation	L04.transportation	—	M27.local transportation M28.inner transportation

Table 2
Notation.

Indices and parameters			
i	The index of quality problems, $i = 1, \dots, I$	$\mathbf{x}_j$	The columns of $\mathbf{X}$, a vector of length I, $\mathbf{x}_j = \{x_{1j}, \dots, x_{Ij}\}^T$
j	The index of features, $j = 1, \dots, J$	y_l	The lth class of root causes
l	The index of root cause classes, $l = 1, \dots, L$	$\mathbf{y}$	The class set of root causes, $\mathbf{y} = \{y_1, \dots, y_L\}^T$
k	The quantity of nearest neighbors of a new quality problem, $k = 1, \dots, K$	z_i	The actual root cause of the ith quality problem, $z_i \in \mathbf{y}$
x_{ij}	The value of the jth feature of the ith quality problem	z_i^k	The actual root cause of the kth nearest neighbor of quality problem i, $z_i^k \in \mathbf{y}$
$\mathbf{X}$	The feature matrix of I quality problems, $\mathbf{X} = [x_{ij}]_{I\times J}$	$\mathbf{z}$	The root cause set of I quality problems, $\mathbf{z} = \{z_1, \dots, z_I\}^T$
$\mathbf{x}_i$	The rows of $\mathbf{X}$, containing J features of the ith quality problem, $\mathbf{x}_i = \{x_{i1}, \dots, x_{iJ}\}$	$\mathbb{D}$	The entire dataset of the historical quality problems, $\mathbb{D} = [\mathbf{X}, \mathbf{z}]$
Estimated variables			
$\hat{z}_i$	The predicted root cause of the ith quality problem, $\hat{z}_i \in \mathbf{y}$	$\hat{z}_i^*$	The optimal predicted root cause of the ith quality problem, $\hat{z}_i^* \in \mathbf{y}$

and Fast KNN classifier. The KD-trees algorithm is good at providing efficient space-partitioning data structure (Verma, Kakkar, & Mehan, 2014), which can be used to speed up the computation by pruning unnecessary data points. In KNN with KD-trees classifier, a binary structure tree is established at first by splitting data into two groups recursively. Then, a forward search and a retrospective search are executed and distances between leaf nodes and the new observation are computed to find nearest neighbors (Hou, Li, Xu, Zhang, & Li, 2018). The Fast KNN classifier is an approximation algorithm, which is highly competitive on computation time for large datasets. With a shrinkage estimator (weighted voting probability estimator), it is able to provide better, i.e. more accurate, prediction performance. Let z_i^k be the actual root cause of the kth nearest neighbor of the new quality problem. d_{ik} is the distance between the new quality problem and its kth nearest neighbor. Two well-known distance measures are Euclidean and Manhattan distance metrics (Boehmke & Greenwell, 2019). The shrinkage estimator $1/d_{ik}$ indicates that closer neighbors have more influences on prediction. Thereby, the probability that the ith quality problem belongs to the lth class of root causes, $P(\hat{z}_i = y_l)$ can be obtained as in Eq. (5.1).

$$P(\hat{z}_i = y_l) = \frac{\sum_{k=1}^{K}(\frac{1}{d_{ik}}\mathbf{1}(z_i^k = y_l))}{\sum_{k=1}^{K}\frac{1}{d_{ik}}}, \quad l = 1, \dots, L \tag{5.1}$$

where $\mathbf{1}(\cdot)$ is an $0-1$ indicator function. Furthermore, we are able to predict the new quality problem to the class with the highest probability, for example, based on Eq. (5.2).

$$\hat{z}_i^* = \underset{l}{\operatorname{argmax}}\, P(\hat{z}_i = y_l) \tag{5.2}$$

5.2.2. The Neural Network model

NN has been widely used in the domain of quality management because of good noise tolerance, high recognition power, and no hypothesis requirement (Du et al., 2012). Various types of models about NN have been developed including Feedforward Neural Network (FFNN), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN) (Ghatak, 2019), etc. In this study, we apply the Multi-layer Perception (MLP) model, which is a class of FFNN and used in a large amount of applications (Lima-Junior & Carpinetti, 2019). We present the detailed NN structure in order to show how RCI can be modeled as a NN.

A MLP model consists of multiple layers (each layer is denoted as $n = 1, \dots, N$), including an input layer (layer 1), one or more hidden layers (layer $n = 2, \dots, N-1$), and an output layer (layer N). A graphical illustration of the MLP model with two hidden layers is shown in Fig. 8. The nodes in the network are called neurons. The number of neurons in each layer is denoted as M_n, $n = 1, \dots, N$. Given the ith quality problem (feature vector), each neuron in the input layer denotes an input quality feature x_{ij} of the original feature space while each neuron in the output layer represents a root cause class y_l of the root causes space. It means there are $M_1 = J$ neurons in the input layer and $M_N = L$ neurons in the output layer in our model. In the hidden layers, each neuron can be viewed as a higher quality feature that is extracted from the previous layer. That means the hidden layer can be interpreted as a mapping to a higher dimensional quality feature space, so that the classifications are more easily defined than in the original quality feature space. In general, the shallow hidden layers extract low-level features, while deeper hidden layers extract high-level features. We predict the probability that the root cause of a quality problem is in a class of root causes. Moreover, we use the back-propagation algorithm to tune the weights (e.g., $\mathbf{w}^{(2)}, \mathbf{w}^{(3)}, \dots, \mathbf{w}^{(N)}$) so that a loss function (measuring the difference between actual values and predicted values, such as the Cross-entropy, Mean Square Error (MSE), and Mean per class error) is minimized, where $\mathbf{w}^{(n)}$ is a $M_n \cdot M_{n-1}$ matrix of weights.

Let $\mathbf{INP}^{(n)}$ denote the input column vector of each layer, $n = 1, \dots, N$. Let $\mathbf{OUP}^{(n)}$ represent the output column vector of the hidden layers, $n = 2, \dots, N-1$. Finally, the output of the model is the probability that the root cause of a quality problem is in the class l, denoted as $P(\hat{z}_i = y_l)$. It can be computed by the Softmax function in Eq. (5.3).

$$P(\hat{z}_i = y_l) = \frac{e^{\mathrm{INP}_l^{(N)}}}{\sum_{t=1}^{L} e^{\mathrm{INP}_t^{(N)}}} \qquad l = 1, \dots, L \qquad i = 1, \dots, I \tag{5.3}$$

where the input of each layer is $\mathbf{INP}^{(1)} = \mathbf{x}_i^T$, $\mathbf{INP}^{(2)} = \mathbf{w}^{(2)} \cdot \mathbf{x}_i^T$, and $\mathbf{INP}^{(n)} = \mathbf{w}^{(n)} \cdot \mathbf{OUP}^{(n-1)}, n = 3, \dots, N$. The output of each hidden layer is calculated by the activation function, such as Sigmoid and ReLU, $\mathbf{OUP}^{(n)} = \sigma(\mathbf{INP}^{(n)}),\ n = 2, \dots, N-1$.

In this study, we concentrate on tuning hyperparameters in the model, such as batch size, epoch, learning rate, model capacity, in order to find the most robust MLP model. Table 3 displays the hyperparameters for better understanding.

We suggest using the following steps to tune the hyperparameters and obtain a robust model.

1. Use a grid search from TensorFlow (an open source software library backed by Google) to explore model capacity and learning rate. In this step, both width (the number of units in one hidden layer) and depth (the number of hidden layers) of the model structure are considered. Adaptive learning rate optimizers are adopted because they provide improvements over regular ones by automatically adapting over time. Each combination of width, depth, and learning rate optimizer becomes a unique model. We loop through each combination and train the model. Note that in order to make the training procedure simpler and faster, the stochastic assignment of the initial weights is commonly used in studies and results in built-in randomness of the model (Scardapane & Wang, 2017). We suggest to take advantage of the randomness and establish the models with high and stable performance.
2. Explore selected models from the former step further by tuning other hyperparameters, such as batch size and epoch quantity. In addition, model regularization is implemented to prevent overfitting problems. Furthermore, we analyze and compare model performance.
3. Select a robust model by taking the evaluation constraints into account. The constraints can be model accuracy, computation time, etc.

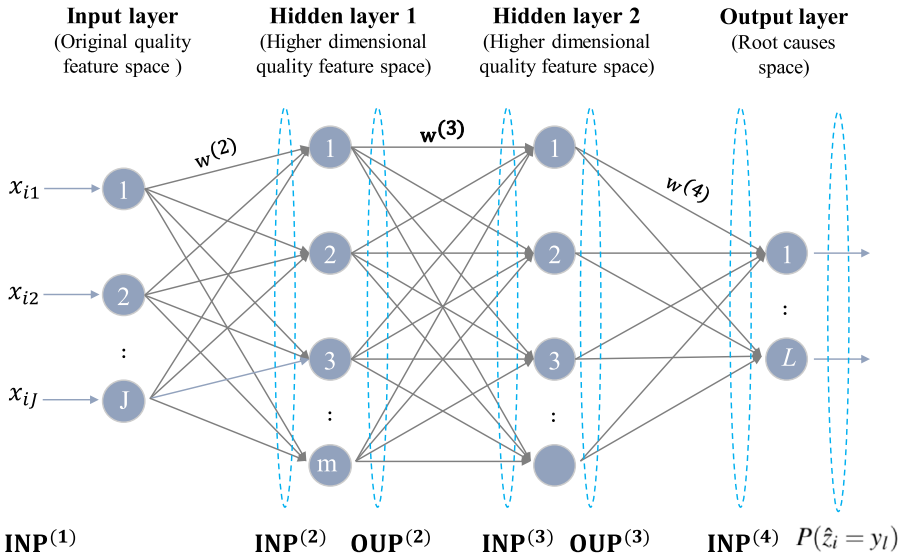


Fig. 8. A simple structure of MLP model with four layers.

Table 3
Hyperparameters.

Related to model structure		Related to model training	
Hidden layers	defines the number of hidden layers in network	Loss function	used to measure the model performance
Hidden units	defines the number of hidden units in one hidden layer	Learning rate optimizer	controls how quickly an NN model learns a problem
Activation function	determines the output of nodes in network	Batch size	defines the number of training samples that run through the network
		Epoch	describes the number of times the model trains with the entire dataset

Table 4
Confusion matrix.

	Actual: Positive	Actual: Negative
Predicted: Positive	True Positive (TP)	False Positive (FP)
Predicted: Negative	False Negative (FN)	True Negative(TN)

Table 5
Performance metrics.

Precision = TP/(TP+FP)
Recall = TP/(TP+FN)
F-measure = (2 ∗ Precision ∗ Recall)/(Precision + Recall)
Accuracy = (TP + TN) / (TP + TN + FP + FN)

5.2.3. Performance evaluation

Classifier performance can be evaluated by the confusion matrix and the common performance metrics which consists of precision, recall, and F-measure, accuracy (Fawcett, 2006). Table 4 shows the confusion matrix for binary classification. The observations (quality problems) in predicted classes are presented by rows and the observations in actual classes are presented by columns. The confusion matrix collects the number of correct and incorrect predictions with count values for each class in its entries. True Positive (TP) and True Negative (TN) indicate correct predictions, while False Positive (FP) and False Negative (FN) indicate incorrect predictions. For example, FP refers to the number of observations that are negative but are predicted to be positive. Table 5 provides the specification of performance metrics based on the value of entries in confusion matrix, which has been introduced in detail by Powers (2011). It is necessary to consider various measures at the same time since each measure considers different aspects of model performance (Pizzo, Lombardo, Manganaro, & Benfenati, 2013). Thereby, we are able to provide a richer evaluation and avoid the bias caused by using a single measure.

In general, two methods are used to interpret the confusion matrix for multi-class classification: macro averaging and micro averaging (Sokolova & Lapalme, 2009). Macro averaging reduces the multi-class classification to a set of binary classifications and averages the results (Van Asch, 2013). Take precision as an example, with L classes in total, Pr_l denotes the precision of class l. Macro averaging precision Pr_{macro} is the average value of the set of binary classifications as Eq. (5.4) shows. As we can see, macro averaging treats each class equally, which is not proper for imbalanced classes. Thus, a weighted macro averaging method may make more sense, as shown in Eq. (5.5) where ω_l is the frequency of a specific class. Micro averaging, on the other hand, focuses on the entire confusion matrix and gives equal weight to each observation. In Eq. (5.6), I is the total number of observations. The above methods may provide very different results and it is hard to decide which method is better (Sebastiani, 2002). Thus, we will use all of them to evaluate performance in this study. Note that the averaging methods will not be implemented for accuracy since it extends to multi-class naturally (Kuhn, Vaughan, & RStudio, 2020).

$$Pr_{macro} = \frac{1}{L}\sum_{l=1}^{L} Pr_l \tag{5.4}$$

$$Pr_{weighted-macro} = \sum_{l=1}^{L} \omega_l Pr_l \tag{5.5}$$

$$Pr_{micro} = \frac{\sum_{i=1}^{I} TP_i}{\sum_{i=1}^{I}(TP_i + FP_i)} \tag{5.6}$$

6. Case study analysis

In this section, we use a case example to illustrate and validate the performance of our big data-driven RCA system and the methodology. The case is based on the data from an automobile factory in China. The data is regarding a specific car model and over its entire life cycle (7 years). The sample data includes about 5,000 components and over 110,000 quality problems from an assembly workshop.

Table 6
Feature libraries.

Library	Category	Feature
Part	7	917
Procedure	7	225
People	4	408
Quality	11	226
Total	29	1776

Table 7
A fragment of the part library.

Feature	Feature description or examples
Exterior	The components used in the vehicle body for various functions and decorations, e.g., headlight, taillight, badge, mirror, etc.
Interior	The components used in the vehicle internal for various functions and decorations, e.g., dashboard, airbag, trim, seat belt, etc.
Chassis	The components assembled on the load-bearing vehicle framework, e.g., steering shaft, steering column, etc.
Drivetrain	The components that delivering power to driving wheels, e.g., engine, gearbox, automatic transmission, etc.
Electronics	The electronic system in vehicles, e.g., antenna motor, semiconductor, electronic sensor, etc.

6.1. PI module

Historical data from the current SAP and quality audit system as well as the expert knowledge are taken into consideration. Hence, as proposed in Fig. 6, we use different types of data from different sources. For example, basic data about the component (component number or supplier name, etc.) are needed and we can download them directly from the system. For some qualitative data, such as the complexity of the package concept (the way that components are packed), we adopt the judgments from quality experts. In this module, feature libraries are set up with data mining, feature combination and standardization. As shown in Table 6, we set up 4 feature libraries, extract 29 categories and 1776 features in total. To indicate the contents of the feature libraries, Table 7 presents the features in the component category of the part library.

As defined in Section 5.1, a vector $\mathbf{x}_i = \{x_{i1}, \ldots, x_{iJ}\}$ can be used to represent the ith quality problem. The value of the element x_{ij} will be non-zero when the jth feature is identified in the ith quality problem. However, over 40% of the textual data in our case study cannot be operated directly. Either the text data is totally blank or it uses general and noninformative statements. In fact, this problem verifies the drawback of single textual data source used in Xu et al. (2018) and Xu and Dang (2020). For a small dataset, however, we are able to impute the missing data with ML methods such as tree-based imputation (Boehmke & Greenwell, 2019). Given a big dataset with sufficient information in our case, we delete them for computation efficiency. Therefore, after data cleaning, the feature dataset $\mathbf{X}$ contains $J = 1776$ features (multiple features) on $I = 63{,}666$ observations (multiple quality problems). Among all features, only 2 features (0.11%) are continuous measures, the other 1774 features (99.89%) are binary categorical features.

6.2. RCI module

Based on the 63,666 observations, we obtain the root cause set of all quality problems $\mathbf{z}$ and the class set of root cause $\mathbf{y}$ with $L = 41$ classes (multiple root causes). The set of detailed root cause classes is in Table 1. Now we create the whole dataset $\mathbb{D}$ and split it into a training set (80% of the observations) and a test set (20% of the observations). We set up classification models on the training set and evaluate the final model performance on the test set. All experiments are conducted by using RStudio (version 3.6.3) on a 64-bit Windows 10 PC with Intel Core i5-8250U 1.60 GHz and 8GB RAM. To begin with, we utilize the Zero Rule classifier to get the baseline performance as a standard (Choudhary & Gianey, 2017). In this naïve approach, we simply use the most recurrent root cause classification to make predictions. With the Zero Rule classifier, the accuracy of prediction in the test set is obtained as 27.43%.

Table 8
MLP model architecture parameter.

Layer type	Number of units	Activation function
Input layer	1776	–
Hidden layer (1~5)	128	Relu
Hidden layer (1~5)	256	Relu
Hidden layer (1~5)	512	Relu
Hidden layer (1~5)	1024	Relu
Hidden layer (1~5)	2048	Relu
Output layer	41	Softmax

Table 9
MLP model training parameter.

Model parameters	Value
Loss function	Cross-entropy
Optimizer	SGD with momentum / RMSprop / Adam
Optimization metrics	Accuracy
Batch size	256
Epoch	30

6.2.1. The K-Nearest Neighbors solutions

As mentioned in Section 5.2.1, we test two algorithms, the Fast KNN classifier and the KNN with KD-trees classifier, and compare the results for different numbers of nearest neighbors, k. 5-fold cross validation is leveraged to find out the best k. As Fig. 9 displays, the Fast KNN classifier outperforms the KNN with KD-trees classifier on accuracy. Moreover, the accuracy for both classifiers increase at first and then decrease with the increase of k. This shows that when adding more nearest neighbors beyond a certain level, noise is also added to the prediction. The best k for the two classifiers are 3 and 4, respectively. We fit the two classifiers with the best chosen k and compare with others in Section 6.3.

6.2.2. The Neural Network solutions

According to Section 5.2.2, we first use a grid search to explore the capacity of model architecture and learning rate. Table 8 displays the architecture of the MLP models. The range of model depth is [1, 2, 3, 4, 5] layers and the range of model width is [128, 256, 512, 1024, 2048] units. Note that the computation burden increases greatly given deeper structure, so we start from single to five hidden layers. MLP with a single hidden layer is considered as shallow NN (SNN) while MLP with several hidden layers is called Deep NN (DNN) (Li, Zhao, Huang, & Gong, 2014). Furthermore, it is a convention to set the number of units a power of 2. Table 9 shows the loss function, learning rate optimizer and other parameters. It is common to use cross-entropy as the loss function to measure model performance for multi-classification problem (Boehmke & Greenwell, 2019). Three well-known adaptive learning rate optimizers are utilized in the training procedure, including Stochastic Gradient Descent (SGD) with momentum, RMSprop, and Adam. For simplicity, we set fixed values for batch size and epoch in this step, but will explore them further in the following steps. In total, 75 ($= 5 * 5 * 3$) models are considered.

As previously mentioned, inherent randomness of the model contributes to robust training results. Thus, we train all 75 models with various random initial values of weights for multiple times, say 10 times. Our results indicate that performances of the three different adaptive learning rate optimizers show only negligible differences and wider models tend to have a better performance. For example, for the SGD with momentum, we rank the top 5 models of each training run

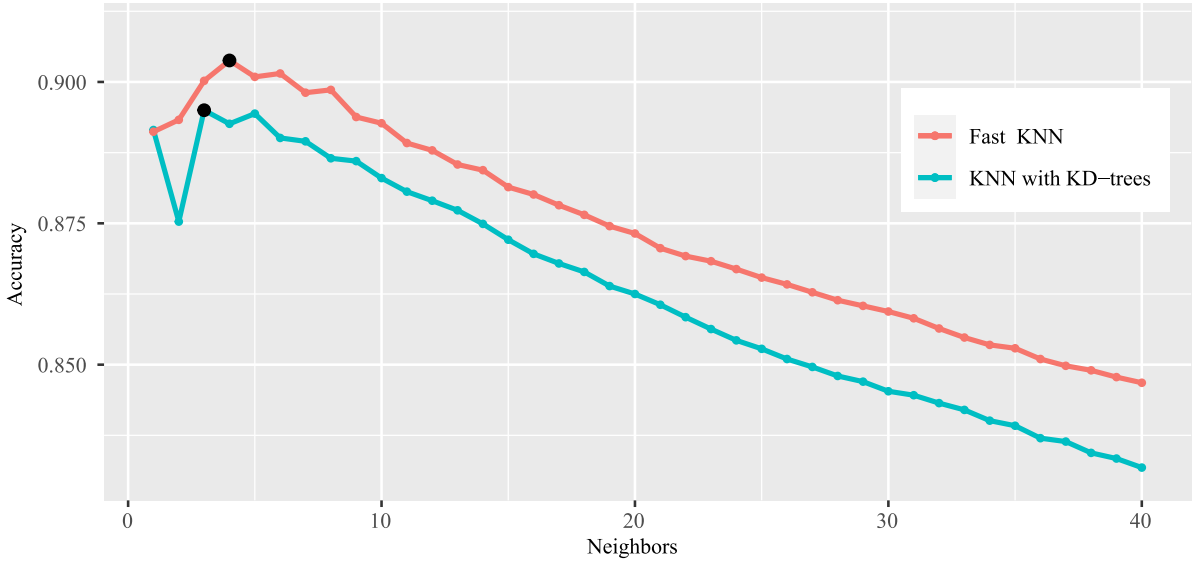
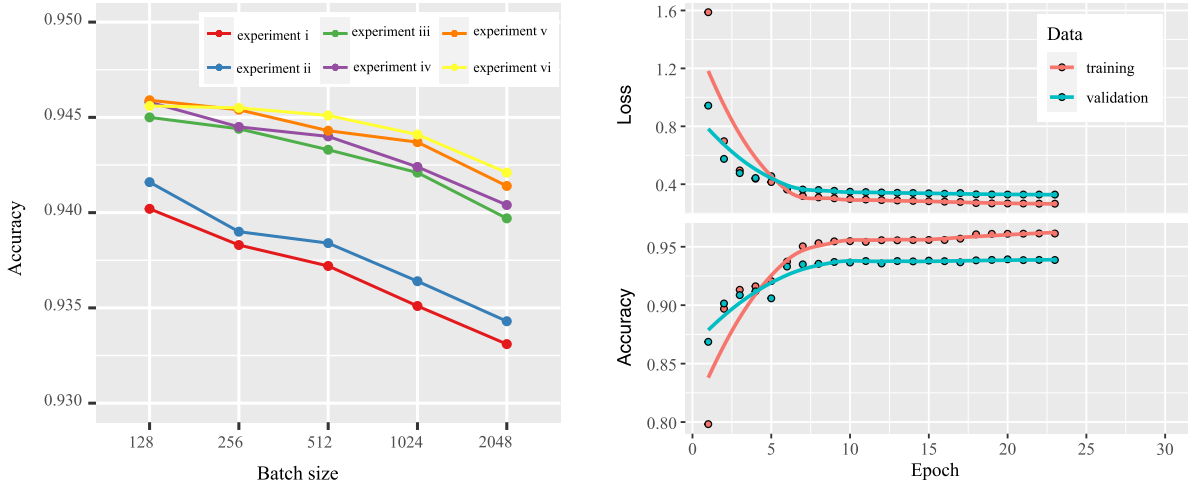


Fig. 9. The analysis results of KNN models with 5-fold cross validation.



(a) Performance comparison (b) Performance of experiment i with batch size of 128

Fig. 10. The results of six experiments of the MLP models.

and calculate the models' occurrence frequency. It turns out that the models with 1024 units and 2048 units account for 88% of the top models. However, it is unclear whether deeper models add any benefits, especially when considering the extra cost of computation time.

From here on, we concentrate only on the models with 1024 units and 2048 units, and with the SGD with momentum optimizer. One major disadvantage of deeper models is computational inefficiency. Since neural networks with two hidden layers can approximate any continuous function infinitely well (Panchal, Ganatra, Kosta, & Panchal, 2011), we only work on MLP models with 1 layer (SNN) and 2 layers (DNN). Six specific NN models are experimented: (i) one hidden layer with 1024 units; (ii) one hidden layer with 2048 units; (iii) two hidden layers with 1024 and 512 units; (iv) two hidden layers both with 1024 units; (v) two hidden layers with 2048 and 1024 units as well as (vi) two hidden layers both with 2048 units. A callback function with an adequately large epoch number enables us to stop training at the time after we get no further improvement of the loss function. Thus, when searching for different batch sizes, we are able to record model performance for further evaluation.

Fig. 10(a) illustrates the performance of the six experiments. With the increase of batch size, accuracy goes down for each experiment. In general, DNN models with 2 hidden layers show obvious advantages over SNN models with 1 hidden layer. However, the performance differences within DNN models or SNN models are small. Therefore, we use experiment i (SNN) and experiment iii (DNN) for further evaluation. Fig. 10(b) shows the training process for experiment i with a batch size of 128.

6.3. Results and discussion

We measure and compare the performance of five classifiers, including the Zero Rule, KNN with KD-trees, Fast KNN, SNN and DNN classifiers. Table 10 presents an overview of their performance. Three kinds of average methods are utilized: macro, weighted-macro, as well as micro averaging (see 5.2.3). For each classifier, the big difference between macro and micro averaging results indicates that the classes are imbalanced. This is reasonable and expected, since we have a large dataset with 41 classes. Therefore, micro averaging is more appropriate in our case.

Performance metrics are displayed from columns of Precision to Accuracy in Table 10. The results show that the SNN and DNN classifiers are more competitive than the others on performance. The prediction

Table 10
Performance measures and comparison.

Classifier	Average method	Precision	Recall	F-Measure	Accuracy	Training time
Zero rule	macro	0.2743	0.0263	0.4305	0.2743	—
	weighted-macro	0.2743	0.2743	0.4305		
	micro	0.2743	0.2743	0.2743		
KNN with KD-trees	macro	0.7404	0.7602	0.7551	0.8991	335.06 s
	weighted-macro	0.9000	0.8991	0.8984		
	micro	0.8991	0.8991	0.8991		
Fast KNN	macro	0.7973	0.7551	0.7815	0.9066	341.36 s
	weighted-macro	0.9060	0.9066	0.9055		
	micro	0.9066	0.9066	0.9066		
SNN	macro	0.9063	0.7268	0.8245	0.9394	219.43 s
	weighted-macro	0.9389	0.9394	0.9378		
	micro	0.9394	0.9394	0.9394		
DNN	macro	0.9074	0.7546	0.8490	0.9464	326.17 s
	weighted-macro	0.9461	0.9464	0.9455		
	micro	0.9464	0.9464	0.9464		

Table 11
Sample output of the DNN model.

Observation No.	Prediction class	Probability	Top class 2	Probability	Top class 3	Probability
#9714	Root cause 2	98.90%	Root cause 16	0.69%	Root cause 39	0.14%
#2824	Root cause 34	72.89%	Root cause 14	13.82%	Root cause 35	7.40%
#1113	Root cause 11	40.73%	Root cause 6	37.85%	Root cause 11	7.41%

accuracy of SNN and DNN is 93.94% and 94.64%, respectively. The SNN classifier has an obvious advantage on training time, which takes only 219.43 s. However, since accuracy is our primary target and model training is a one-time effort for a certain period, the DNN classifier can be our top choice in this case. According to the data of the case company, the time required to discover root causes varies from hours to weeks in practice, given quality problems with different levels of complexity. With our trained DNN model, we are able to predict the root causes of multiple quality problems with high accuracy in only a few seconds. In other words, we can predict the root causes effectively and efficiently by using ML classification methods.

Table 11 illustrates the output of the DNN model in the RCI Module. Regarding to the observations (quality problems) shown in the first column, prediction classes and their probabilities follow. As we can see, the probabilities of prediction classes may exhibit big differences. A threshold value can be helpful to resolve this problem. That means that multiple top classes will be taken into account until the cumulative probability reaches a specific value, say 85%. Observation #9714 is predicted to Root Cause 2 with 98.90%, so only Root Cause 2 will be considered. For observation #1113, the cumulative probability achieves 85% only when the top 3 classes are covered. In this situation, manual intervention may be needed to make the final judgment. Thus, the threshold method allows us to be efficient in observations with high accuracy and also ensure information comprehensiveness on the others.

7. Conclusions and future work

In this study, we propose a big data-driven RCA system by applying ML techniques to speed up the quality problem solving process for complex manufacturing processes. The framework of the system provides a guidance for a company to develop its applicable system in practice. Such a system will also be an information and knowledge-sharing platform for different users. In addition, to the best of our knowledge, our study is the first to utilize real production data from multiple data sources and apply multiple ML methods simultaneously. We collect real company data from multiple channels (the ERP database, the production quality audit database, and expert knowledge). Expert knowledge and experience are integrated with real production data. Furthermore, based on the features obtained, we propose a systematic method to describe the quality problems by using feature vectors. The consideration of multiple types of data helps to improve the accuracy of prediction. Then, with the application of ML methods on RCI, we explore the relationships between features and root causes of the quality problems. Our ML algorithms support RCA for both individual quality problem and multiple quality problems. It improves the efficiency of RCA for a busy and complex production process. In our case example, root causes for more than 110,000 quality problems can be predicted in seconds. The case results also show the high accuracy of the root cause identification by our method. It is worth noting that the proposed system and methodology have great applicability for the manufacturing industry due to the generalized description of quality problems and the subsequent application of ML techniques for the RCA. Thus, they can also be applied to other manufacturing contexts, such as aerospace, electronics, and mobile devices.

The empirical findings of our research have a number of important implications for industry practice and will be of interest to quality managers from various manufacturing companies. For example, features with high frequency can be identified more easily given a set of quality problem data. Quality specialists are then able to investigate these features in detail and provide continuous improvements. Moreover, a systematic RCA system reduces the knowledge and experience limitations in the field of quality management and control. However, a limitation of this study is that a large amount of historical quality data is required. For established manufacturers who do not have to create a totally new product but instead produce a successor or an updated version, quality data from former or similar products can be useful.

This research also leads to several interesting questions for further investigation. For example, other classification methods in the RCI Module, e.g., Support Vector Machines (SVM) and Random Forest (RF) should be tested. Another natural progression of this work is to extend the current system to manufacturers with multiple and diverse products such as top luxury products, mid-to-high-end and low-end products. Furthermore, it would also be interesting to consider more case studies from both automotive companies and other manufacturing companies to further consolidate and enrich our proposed system and methodology. Last but not the least, focusing on the dynamics aspects of an RCA system with respect to the product lifecycle will be a useful way to establish a greater degree of validity for the RCA system.

Table A.12
Detailed feature library.

Library	Category	Feature	Feature description or examples
Part	component category	exterior	the components used in the vehicle body for various functions and decorations, e.g., headlight, taillight, badge
		interior	the components used in the vehicle internal for various functions and decorations, e.g., dashboard, airbag, trim
		chassis	the components assembled on the load-bearing vehicle framework, e.g., steering shaft, steering column
		drivetrain	the components that delivering power to driving wheels, e.g., engine, gearbox, automatic transmission
		electronics	the electronic system in vehicles, e.g., antenna motor, semiconductor, electronic sensor
	component family	191 component families	a group of components that have similar functions, e.g., left & right sides, high & low choices, various colors
	component cluster	88 clusters	a group of components for specific business purposes
	part number	557 part numbers	unique component ID
	component supplier	71 suppliers	supplier of components
	lot size	3 lot sizes	parameters for purchasing
	purchasing mode	import part	components from foreign countries
		local part	components from local country
Procedure	package type	supplier direct package	supplier packs components with the single-use packages and sends them to the manufacturer
		LSP repacked package	supplier sends components to Logistic Service Provider (LSP) with returnable bins, LSP repacks components with the single-use packages, and then send them to the manufacturer
		manufacturer package	the manufacturer recycling package, usually used for local components
	package concept	simple	no layering, no partition, components are scattered in the package
		medium	there are layering and partition, components are neatly placed in the package
		complex	there are layering, partition and extra protection materials, components are neatly placed in the package
	line feeding package	original supplier package	no repacking action in manufacturer, but send original supplier packages for sequencing, pre-assemble or assemble
		original LSP package	no repacking action in manufacturer, but send original LSP packages for sequencing, pre-assemble or assemble
		repacked bin	components are repacked into an internal recycle bin at first and then sent to different areas for sequencing, pre-assemble or assemble
	warehouse	5 warehouses	warehouses where the components are stored in manufacturer
	unloading point	6 unloading points	locations where the components are unloaded in manufacturer
	internal material flow	8 internal material flows	the material flow of components after arriving at the manufacturer. For example, from container yard to the warehouse and then assemble line
	fitment point	197 fitment points	the location where components are assembled
People	operator category	new operator	operators without any operating experience before
		skillful operator	operators with operating experience before
	material planner	35 planners	the employees who are responsible for ordering the components
	detect worker	363 workers	the employees who identify the quality defect
	quality experts	32 experts	the employees who are responsible for quality problem solving
Quality	defect category	72 kinds of quality defects	e.g., scratch, damage, rust, leakage
	defect distribution rule	random	defect is distributed randomly on part
		concentrated	defect is concentrated on specific area of the part
	defect location material	18 kinds of materials	components are made of various materials, but the defect may only happens at specific area with specific material, e.g., plastic, metal, fabric, rubber, software
	defect quantity	the number of defects	how many defectives are detected in one quality defect incident
	detect location	26 locations	locations where the defectives are found
	detect method	manual recognition	the defect is identified by manual recognition, e.g., by naked eye, touching or smelling
		machine recognition	the defect is identified by machine recognition, e.g., by electronic test
	defect level	1	safety-relevant defect or violation of legal requirements, e.g., leakage issue of the gearbox
		2	defect with potential functional consequences that may lead to breaking down
		3	defect with potential functional consequences that annoys the customer
		4	defect with potential functional effect, or significant visual deviation, causing dissatisfaction, for example, cabin lock damage defect cause front cabin cover cannot be opened
		5	defect without functional effect, but with clear visual impairment, and repetitive or marked noise or smell. For instance, deep dent on the steering wheel or visible deformation of the airbag
		6	defect without functional effect and with slight visual impairment, for instance, soft scratch on badge or dashboard
	defect time	104 time features	when the defects are identified, including years, month, week, day

CRediT authorship contribution statement

Qiuping Ma: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing. **Hongyan Li:** Conceptualization, Methodology, Supervision, Writing – original draft, Writing – review & editing. **Anders Thorstenson:** Writing – review & editing.

Appendix. Feature libraries

Table A.12 displays the detailed feature libraries specified in the PI Module. The first column consists of the four corresponding libraries, namely part, procedure, people, and quality. The second column shows the specific categories in each library. The third column represents the features that can be combined together to describe a quality problem and be represented by a J-dimensional vector. The last column provides feature descriptions or examples for better understanding.

References

Aurisicchio, M., Bracewell, R., & Hooey, B. L. (2016). Rationale mapping and functional modelling enhanced root cause analysis. *Safety Science*, *85*, 241–257. http://dx.doi.org/10.1016/j.ssci.2015.12.022.

Ballabio, D., Grisoni, F., & Todeschini, R. (2018). Multivariate comparison of classification performance measures. *Chemometrics and Intelligent Laboratory Systems*, *174*, 33–44. http://dx.doi.org/10.1016/j.chemolab.2017.12.004.

Boehmke, B., & Greenwell, B. (2019). *Hands-on machine learning with R* (1st ed.). Chapman and Hall/CRC, http://dx.doi.org/10.1201/9780367816377.

Brown, B., Chui, M., & Manyika, J. (2011). Are you ready for the era of 'big data'. *McKinsey Quarterly*, *4*(1), 24–35.

Chang, Y.-H., Yeh, C.-H., & Chang, Y.-W. (2013). A new method selection approach for fuzzy group multicriteria decision making. *Applied Soft Computing*, *13*(4), 2179–2187. http://dx.doi.org/10.1016/j.asoc.2012.12.009.

Chemweno, P., Morag, I., Sheikhalishahi, M., Pintelon, L., Muchiri, P., & Wakiru, J. (2016). Development of a novel methodology for root cause analysis and selection of maintenance strategy for a thermal power plant: A data exploration approach. *Engineering Failure Analysis*, *66*, 19–34. http://dx.doi.org/10.1016/j.engfailanal.2016.04.001.

Choudhary, R., & Gianey, H. K. (2017). Comprehensive review on supervised machine learning algorithms. In *2017 international conference on machine learning and data science (MLDS)* (pp. 37–43). Noida: IEEE, http://dx.doi.org/10.1109/MLDS.2017.11.

Djurdjanovic, D., Lee, J., & Ni, J. (2003). Watchdog agent—An infotronics-based prognostics approach for product performance degradation assessment and prediction. *Advanced Engineering Informatics*, *17*(3–4), 109–125. http://dx.doi.org/10.1016/j.aei.2004.07.005.

Du, S., Lv, J., & Xi, L. (2012). A robust approach for root causes identification in machining processes using hybrid learning algorithm and engineering knowledge. *Journal of Intelligent Manufacturing*, *23*(5), 1833–1847. http://dx.doi.org/10.1007/s10845-010-0498-9.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874. http://dx.doi.org/10.1016/j.patrec.2005.10.010.

Ghatak, A. (2019). *Deep learning with R*. Singapore: Springer Singapore, http://dx.doi.org/10.1007/978-981-13-5850-0.

Hanaysha, J., Hilman, H., & Abdul-Ghani, N. H. (2014). Direct and indirect effects of product innovation and product quality on brand image: Empirical evidence from automotive industry. *International Journal of Scientific and Research Publications*, *4*(11), 1–7.

He, Y., Zhu, C., He, Z., Gu, C., & Cui, J. (2017). Big data oriented root cause identification approach based on axiomatic domain mapping and weighted association rule mining for product infant failure. *Computers & Industrial Engineering*, *109*, 253–265. http://dx.doi.org/10.1016/j.cie.2017.05.012.

Hou, W., Li, D., Xu, C., Zhang, H., & Li, T. (2018). An advanced k nearest neighbor classification algorithm based on KD-tree. In *2018 IEEE international conference of safety produce informatization (IICSPI)* (pp. 902–905). Chongqing, China: IEEE, http://dx.doi.org/10.1109/IICSPI.2018.8690508.

Imandoust, S. B., & Bolandraftar, M. (2013). Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background. *International Journal of Engineering Research and Applications*, *3*(5), 605–610.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *Springer texts in statistics*: *Vol. 103*, *An introduction to statistical learning*. New York, NY: Springer New York, http://dx.doi.org/10.1007/978-1-4614-7138-7.

Kuhn, M., Vaughan, D., & RStudio (2020). Tidy characterizations of model performance. R package version 0.0.6.

Latino, M. A., Latino, R. J., & Latino, K. C. (2019). *Root cause analysis: Improving performance for bottom-line results*. CRC press.

Lehtinen, T. O., Mäntylä, M. V., & Vanhanen, J. (2011). Development and evaluation of a lightweight root cause analysis method (ARCA method) – Field studies at four software companies. *Information and Software Technology*, *53*(10), 1045–1061. http://dx.doi.org/10.1016/j.infsof.2011.05.005.

Leveson, N. (2004). A new accident model for engineering safer systems. *Safety Science*, *42*(4), 237–270. http://dx.doi.org/10.1016/S0925-7535(03)00047-X.

Li, J., Zhao, R., Huang, J.-T., & Gong, Y. (2014). Learning small-size DNN with output-distribution-based criteria. In *Fifteenth annual conference of the international speech communication association*.

Liang, K., & Zhang, Q. (2010). Study on the organizational structured problem solving on total quality management. *International Journal of Business and Management*, *5*(10), 178–183. http://dx.doi.org/10.5539/ijbm.v5n10p178.

Lima-Junior, F. R., & Carpinetti, L. C. R. (2019). Predicting supply chain performance based on SCOR® metrics and multilayer perceptron neural networks. *International Journal of Production Economics*, *212*, 19–38. http://dx.doi.org/10.1016/j.ijpe.2019.02.001.

Lokrantz, A., Gustavsson, E., & Jirstrand, M. (2018). Root cause analysis of failures and quality deviations in manufacturing using machine learning. *Procedia CIRP*, *72*, 1057–1062. http://dx.doi.org/10.1016/j.procir.2018.03.229.

Medina-Oliva, G., Iung, B., Barberá, L., Viveros, P., & Ruin, T. (2012). Root cause analysis to identify physical causes. In *Proceedings of PSAM 2011 & ESREL*.

Mirsu, D. B. (2013). Monitoring help desk process using KPI. In V. E. Balas, J. Fodor, A. R. Várkonyi-Kóczy, J. Dombi, & L. C. Jain (Eds.), *Soft computing applications, Vol. 195* (pp. 637–647). Berlin, Heidelberg: Springer Berlin Heidelberg, http://dx.doi.org/10.1007/978-3-642-33941-7_56.

Mohammadnazar, H., Pulkkinen, M., & Ghanbari, H. (2019). A root cause analysis method for preventing erratic behavior in software development: PEBA. *Reliability Engineering & System Safety*, *191*, Article 106565. http://dx.doi.org/10.1016/j.ress.2019.106565.

Mueller, T., Greipel, J., Weber, T., & Schmitt, R. H. (2018). Automated root cause analysis of non-conformities with machine learning algorithms. *18*(4), 60–72. http://dx.doi.org/10.5604/01.3001.0012.7633.

Nguyen, T., Zhou, L., Spiegler, V., Ieromonachou, P., & Lin, Y. (2018). Big data analytics in supply chain management: A state-of-the-art literature review. *Computers & Operations Research*, *98*, 254–264. http://dx.doi.org/10.1016/j.cor.2017.07.004.

Okes, D. (2005). Improve your root cause analysis. *Manufacturing Engineering*, *134*(3), 171–178.

Pamučar, D., Mihajlović, M., Obradović, R., & Atanasković, P. (2017). Novel approach to group multi-criteria decision making based on interval rough numbers: hybrid DEMATEL-ANP-MAIRCA model. *Expert Systems with Applications*, *88*, 58–80. http://dx.doi.org/10.1016/j.eswa.2017.06.037.

Panchal, G., Ganatra, A., Kosta, Y. P., & Panchal, D. (2011). Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers. *International Journal of Computer Theory and Engineering*, 332–337. http://dx.doi.org/10.7763/IJCTE.2011.V3.328.

Pizzo, F., Lombardo, A., Manganaro, A., & Benfenati, E. (2013). In silico models for predicting ready biodegradability under REACH: A comparative study. *Science of the Total Environment*, *463*, 161–168. http://dx.doi.org/10.1016/j.scitotenv.2013.05.060.

Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, *2*, 37–63.

Safarzadeh, S., Khansefid, S., & Rasti-Barzoki, M. (2018). A group multi-criteria decision-making based on best-worst method. *Computers & Industrial Engineering*, *126*, 111–121. http://dx.doi.org/10.1016/j.cie.2018.09.011.

Scardapane, S., & Wang, D. (2017). Randomness in neural networks: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *7*(2), Article e1200. http://dx.doi.org/10.1002/widm.1200.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, *34*(1), 1–47. http://dx.doi.org/10.1145/505282.505283.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437. http://dx.doi.org/10.1016/j.ipm.2009.03.002.

Solé, M., Muntés-Mulero, V., Rana, A. I., & Estrada, G. (2017). Survey on models and techniques for root-cause analysis. arXiv:abs/1701.08546.

Van Asch, V. (2013). *Macro-and micro-averaged evaluation measures [[basic draft]]*. Belgium: CLiPS, 49.

Verma, D., Kakkar, N., & Mehan, N. (2014). Comparison of brute-force and KD tree algorithm. *International Journal of Advanced Research in Computer and Communication Engineering*, *3*(1), 5291–5294.

WarrantyWeek (2016). US Automaker's warranty expenses. https://www.warrantyweek.com/archive/ww20160107.html. Online, accessed 26-Dec-2019.

Wieczerniak, S., Cyplik, P., Milczarek, J., et al. (2017). Root cause analysis methods as a tool of effective change. *Business Logistics in Modern Management*, *17*, 611–627.

Xu, Z., & Dang, Y. (2020). Automated digital cause-and-effect diagrams to assist causal analysis in problem-solving: A data-driven approach. *International Journal of Productions Research*, *58*(17), 5359–5379. http://dx.doi.org/10.1080/00207543.2020.1727043.

Xu, Z., Dang, Y., & Munro, P. (2018). Knowledge-driven intelligent quality problem-solving system in the automotive industry. *Advanced Engineering Informatics*, *38*, 441–457. http://dx.doi.org/10.1016/j.aei.2018.08.013.

York, D., Jin, K., Song, Q., & Li, H. (2014). Practical root cause analysis using cause mapping. In *Proceedings of the international multiconference of engineers and computer scientists*. Hong Kong: International Association of Engineers.

Yuniarto, H. (2012). The shortcomings of existing root cause analysis tools. In *Proceedings of the world congress on engineering*.

Zhang, Y., Ren, S., Liu, Y., Sakao, T., & Huisingh, D. (2017). A framework for Big Data driven product lifecycle management. *Journal of Cleaner Production*, *159*, 229–240. http://dx.doi.org/10.1016/j.jclepro.2017.04.172.

Zhou, K., Fu, C., & Yang, S. (2016). Big data driven smart energy management: From big data to big insights. *Renewable and Sustainable Energy Reviews*, *56*, 215–225. http://dx.doi.org/10.1016/j.rser.2015.11.050.