Preprints, 10th IFAC International Symposium on
Advanced Control of Chemical Processes
Shenyang, Liaoning, China, July 25-27, 2018

# A Platform for Fault Diagnosis of High-Speed Train based on Big Data ⋆

**Quan Xu** * **Peng Zhang** * **Wenqin Liu** * **Qiang Liu** *
**Changxin Liu** * **Liangyong Wang** * **Anthony Toprac** *
**S.Joe Qin** **

* *State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China (e-mail: quanxu@mail.neu.edu.cn).*
** *Mork Family Department of Chemical Engineering and Materials Science, University of Southern California, Los Angeles, CA 90089, USA(e-mail: sqin@usc.edu)*

**Abstract:** High-speed trains are very fast (e.g. 350km/h) and operate at high traffic density, so once a fault has occurred, the consequences are disastrous. In order to better control the train operational status by timely and rapid detection of faults, we need new methods to handle and analyze the huge volumes of high-speed railway data. In this paper, we propose a novel framework and platform for high-speed train fault diagnosis based on big data technologies. The framework aims to allow researchers to focus on fault detection algorithm development and on-line application, with all the complexities of big data import, storage, management, and real-time use handled transparently by the framework. The framework uses a combination of cloud computing and edge computing and a two-level architecture that handles the massive data of train operations. The platform uses Hadoop as its basic framework and combines HDFS, HBase, Redis and MySQL database as the data storage framework. A lossless data compression method is presented to reduce the data storage space and improve data storage efficiency. In order to support various types of data analysis tasks for fault diagnosis and prognosis, the framework integrates online computation, off-line computation, stream computation, real-time computation and so on. Moreover, the platform provides fault diagnosis and prognosis as services to users and a simple case study is given to further illustrate how the basic functions of the platform are implemented.

*Keywords:* Fault Diagnosis, High-Speed Train, Big Data, Cloud Computing, Edge Computing

## 1. INTRODUCTION

Railway operations world-wide are increasingly concerned with reducing traffic accidents, which is a long-term socio-economic problem of all rail systems. Although the installed base of high-speed trains in China is currently leading the world, failure rates are high compared to ICE in Germany and TGV in France, for example, there is a total of 547 faults from January to June in the year of 2013 according to Jia and Li (2014). Given the high demand for stability and reliability in railway systems, better methods for system life-cycle management are required.

High-speed trains are very fast (e.g. 350km/h) and operate at high traffic density, so once a fault has occurred, the consequences are disastrous, not only in traffic disruption across the regional railway network, but very often in the safety of passengers and crew. In order to better control train operation by timely and rapid detection of faults, we need new methods to handle and analyze high-speed railway data. With the continuing expansion of China's high-speed railway network and increasing train speeds, the data generated by high-speed train operation is increasing day by day. Timely analysis and processing of this data, as well as mining for hidden information in large data sets, has become a top priority. Traditional signal processing methods, however, have been developed for use on small data volumes, and their performance is far less than required for processing massive high-speed data. For such data volumes, revolutionary measures are needed for data accessibility, analysis and management.

With the rapid development of cloud computing and big data technologies Bughin et al. (2010), the management and timely analysis of high-speed railway data has become feasible. However, at present there are few publications in the area of high-speed train monitoring and diagnosis Liu et al. (2016), and even few for the general problem of fault monitoring and diagnosis based on big data. Thaduri *et al.* Thaduri et al. (2015) gave an overview of big data technologies in the context of transportation, with specific reference to railways, and proposed the idea of combining and integrating existing transportation data modules to support maintenance decision making. However, these papers do not address the problem of fault diagnosis for

⋆ Project supported by the National Natural Science Foundation, China(61490704, 61440015) and the National High-Tech. R&D Program, China (No. 2015AA043802).

high-speed trains, and do not provide any in-depth study of frameworks for large data processing. Shao *et al.* Shao et al. (2013) proposed a framework for data management of high-speed railway equipment, where cloud computing combined with a MapReduce programming model based on the Hadoop platform provided a feasible technical solution. But the MapReduce programming model can only support batch processes; as a result, this proposed framework can only address a small number of the data analysis scenarios of high-speed train systems.

In contrast, general data-driven modeling and fault diagnosis methods have seen increasing development and application in recent years Qin (2012). Some researchers have indicated that these new monitoring and fault diagnosis methods may find useful application to data from high-speed train operation Liu et al. (2016).

On the one hand, a large amount of high-speed train operation data may help us to diagnose the faults, but on the other hand, it also brings great technical challenges to researchers for large data fault modeling and diagnosis. Researchers need to spend more time to store, preprocess and manage these data before they can begin modeling and diagnosing. In order to effectively use and efficiently analyze high-speed train operation data, this paper proposes a framework and platform for high-speed train fault diagnosis based on big data. The framework and platform provides data storage, preprocessing, management, computing, visualization and other functions to help researchers quickly build fault modeling and diagnostics, and researchers only focus on modeling and diagnosis itself without having to consider the basic tasks such as large data storage and management. The platform also provides some basic fault modeling methods which are used on small sample dataset to researchers, researchers can directly use them or expand them.

The paper is organized as follows. First, data description for high-speed train is discussed in Section 2. The system architecture and modules are introduced in Section 3. In Section 4 we apply a case study to real-world data. We present conclusion and future work in Section 5.

## 2. DATA DESCRIPTION FOR HIGH-SPEED TRAIN

High-speed train operation data have typical characteristics of big data, with details as follows:

- Volume
  There are a very large number of measurement points and feedback loops to be monitored and controlled in the operation of high-speed trains; typically, in a long group (16 vehicles), more than 2000 sensors, of which 300 are used for voltage and current measurement at 40 microsecond sampling periods. A round-trip will produce more than 500GB of data per train over an estimated 5 hours average travel time. At present, there are more than 2000 high-speed trains in operation in China, so the amount of high-speed train operational data will quickly escalate from terabytes in size to petabyte levels. In addition to such operational data, a large amount of maintenance data is also produced.
- Variety
  High-speed train data sources are from three levels: train level, vehicle level, and device level.For each of these levels of data, sampling frequencies and data characteristics differ greatly. Railway departments have developed document formats for a large amount of operation and maintenance data. In contrast, monitoring and testing processes produce information in video and image formats. Therefore, high-speed train data include large amounts of structured data and unstructured data, with multi-structure, multi-scale, and irregular sampling features.
- Velocity
  The requirement for data processing speed in the application of fault detection to high speed train operation is very fast. This requires rapid acquisition of sensor data in real-time, followed by quick execution of status diagnoses and determination of trends in system health indicators.
- Value
  Under normal operation, high-speed train equipment operates at steady state, with very small stochastic variance in operating parameters. However, the development and application of fault diagnosis methods requires monitoring operational data over relatively long periods, even the whole life cycle of the train, time over which data variation occurs and can indicate fault modes.

In short, the big data of high-speed train operation and maintenance contains great value and opportunities for high-speed train fault diagnosis. Applying fault detection methods using existing fault diagnosis systems and frameworks, however, is very difficult given the massive amount of data, multiple data formats, and requirements for real-time processing speed. There is, therefore, a real need to create a suitable framework that supports the development and application of fault diagnosis methods to high-speed train operation.

## 3. SYSTEM ARCHITECTURE AND MODULES

### *3.1 System Architecture*

High-speed train operation data has typical big data characteristics, and the processing and analysis of train data is the core of the fault diagnosis framework for high-speed trains. Hadoop is currently the most popular big data processing platform. Given current development trends, Hadoop as a data processing platform standard will not change in the near term future, so the proposed high-speed train fault diagnosis platform uses Hadoop as it's basic framework. Since it is not possible to deploy large-scale computing clusters in high-speed trains, the framework uses a combination of cloud computing and edge computing Pedro et al. (2015) and a two-level architecture (Figure 1) that handles the massive data of train operation. Fault systems include cloud-based fault diagnosis system (*CFDS*) deployed at a central cloud computing station (lower part of figure 1) and on-board fault diagnosis systems (*VFDS*) (upper part of figure 1) deployed on high-speed trains. The two systems are connected via GSM-R network. The *CFDS* is composed of data collection, data storage, computing framework, data analytics, algorithm

library and diagnosis/prognosis services modules, and the *VFDS* is made up of real-time diagnosis/prognosis and non real-time diagnosis/prognosis modules. The *CFDS* is responsible for processing big data of high-speed train operation and provides a variety of data services, including data analytics, diagnosis, prognosis, retrieval, and query.

In order to ensure reliable, real-time execution of fault diagnosis, some functions of fault diagnosis are implemented by the *VFDS*, and some functions of fault diagnosis that require massive data or extensive computation are implemented by the *CFDS*. Models for real-time fault diagnosis and prognosis modules that run in the VFDS are downloaded from a model services component of the *CFDS*. The *VFDS* can request fault diagnosis services from the *CFDS* through the GSM-R network. Real-time fault diagnosis and prognosis modules (local diagnosis and prognosis) in the *VFDS* are responsible for processing real-time data, and a portion of these results, as well as some of the real-time data, are transferred to the *CFDS* through non real-time diagnosis and prognosis modules. The *CFDS* uses the received data to perform an auxiliary diagnosis for faults not found by the *VFDS* using cloud-based diagnosis and prognosis models/algorithms, and returns the results to the non real-time diagnosis and prognosis module of the *VFDS*. In general, the *VFDS* models of real-time fault diagnosis and prognosis modules are distinct from the *CFDS* models executed by the cloud computing center, which are simplified to enable use by the *VFDS* on the high-speed train. The non real-time diagnosis and prognosis module of the *VFDS* only requests cloud services from the *CFDS* and does not download the models which store in the cloud computing center.

Because of GSM-R network bandwidth limitations, only a small part of high-speed train operation data can be transmitted to the cloud computing center in real-time; as a result, the bulk of operation data must be stored on the train. When the train is stationed, operation data can be downloaded from on-board data recording equipment and stored in the *CFDS* via the internet. These massive datasets from the operation of different trains may then be used to evaluate and validate our fault diagnosis and prognosis models, updating them as required. The real-time fault diagnosis and prognosis modules of the *VFDS* will then download and update the on-board *VFDS* models on the trains (see data flow in Fig. 1). If a model of real-time fault diagnosis and prognosis modules of the *VFDS* only need to update some parameters, the *CFDS* will only send those parameters to *VFDS* to update the model, if not, the *CFDS* will send the new model to *VFDS*.

### *3.2 Modules*

*Data Collection:* Many of high-speed train equipment and sub-system vendors are protective of their technology, providing only limited access to information data. In many cases, this is limited to a log file that records operational data in a specified format, with any additional access to data made difficult. The framework platform must therefore include tools for parsing files in a variety of formats to extract relevant data, a feature we provide with a data collection component that can be configured to

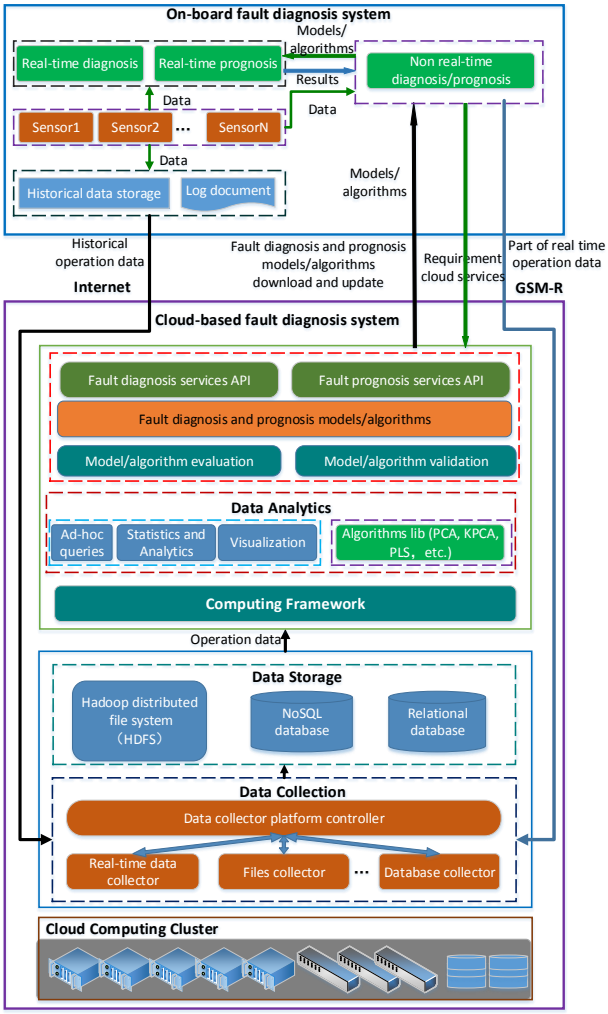


Fig. 1. System architecture and data flow of the proposed framework and platform

import data from a variety of structured and unstructured data formats.

Data quality is a big problem for the high-speed train data collected. To improve the usability of the data, the collected data will be preprocessed, including data validity and consistency verification, etc. In addition, the framework provides some basic algorithms (e.g. *K-NN*, *K-means*, *LOF*, and so on) to help users to clean the data. The system uses *K-NN* algorithm as a default setting, users can also choose the appropriate cleaning algorithms according to their own needs.

*Data Storage:* In traditional data storage, relational databases are preferred, but they cannot provide sufficient performance in big data environments Jiang et al. (2014). As a result, NoSQL databases have been developed to address the challenges posed by multiple data types in large-scale data sets and to support big data applications Chang et al. (2006). NoSQL databases, however, lack the atomicity, consistency, isolation, and durability (ACID) constraints of traditional relational databases. Therefore, our framework uses a variety of storage engines to efficiently store and manage massive high-speed train data, including: *1) HDFS*: storing unstructured files data (e.g. documents, images, videos), *2) HBase (NoSQL database)*:

storing volume data (e.g. operation state data of equipment, vehicle operation data), *3) MySQL*: storing high-speed train basic data (e.g. train ID, line, driver ID,...), dictionary data, analysis results, etc, and *4) Redis (Key-Value in-memory database)* : storing frequently updated and accessed data.

*HBase* database is a typical column-oriented database and is used in our framework as the core database to store and manage the large volume data of high-speed trains. Because the data of each column is characteristically of uniform type, we can effectively compress them to save storage space and improve data storage efficiency. The data compression process is shown in Figure 2. Data collectors persist collected data in the *Redis* database and write a copy of the data to a local file as backup in case of *Redis* database failure. If *Redis* crashes, the data persisted in local files can be reloaded to *Redis* to avoid data loss. The compression module periodically reads data in blocks from the *Redis* database and compresses the data block using a lossless data compression algorithm designed for the characteristics of high-speed train data. Compressed data is then stored in the *HBase*; when the data is stored successfully, the corresponding data in the local file is deleted to save storage space. This data storage algorithm 1 is defined as follows:

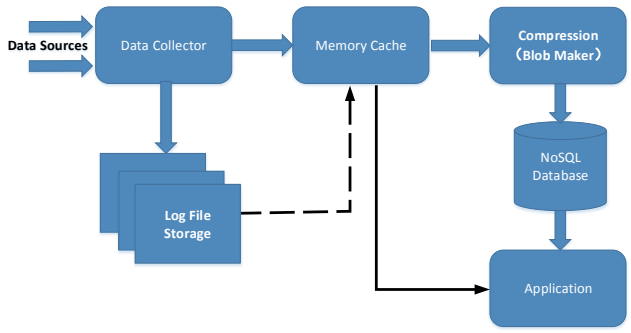


Fig. 2. High-speed train data compression process

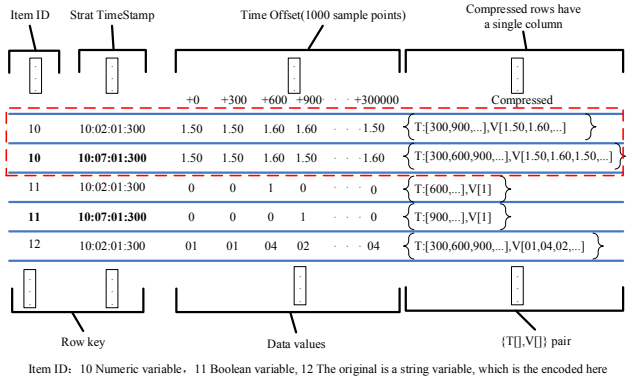


Fig. 3. An example of the lossless data compression method.

*Computing Framework:* The big data software processing architecture is built based on the distributed file system *HDFS* and uses *YARN* to achieve unified resource management and scheduling (Figure 4). The architecture provides computation modules such as online computation (*HBase*), off-line computation (*Map/Reduce*), stream computation (*Storm*), real-time computation (*Spark*) and so on. It integrates *Kafka*, *Flume* and *Sqoop* tools to achieve data uptake. Data mining class libraries such as *MLlib*, *SparkR* and *Mahout*, and analysis tools such as *Pig* and *Hive*, are all integrated to provide users the ability to mine and analyze high-speed train data. The core application of the framework is to provide the *FDP* services (*FDP class library*), a set of methods developed for fault detection on big data, for the fault diagnosis and prognosis of high speed trains. The framework thus aims to allow researchers to focus on fault detection algorithm development, with all the big data management requirements handled transparently by the framework.

**Algorithm 1:** High-speed train data compression

**Input:** Uncompressed data in the *Redis*
**Output:** {T[ ],V[ ]} pairs for each data item in a data block to be compressed
**Initialization:** Sampling a data block to be compressed from *Redis*
**Step 1:** Get all data of an uncompressed data item from the data block
**Step 2:** Sort data according to the sample time (sample time offset)
**Step 3:** Get the first N time series data, $Data[N]$, $N$ is the number of compressed data at a time (such as N=1000 in Figure 3)
**Step 3.1:** Determine the data type of the data item, if the data item is *numeric variable*, go to *Step 3.2*, if *Boolean variable*, go to *Step 3.3*, if *string variable*, go to *Step 3.4*, otherwise, go to *Step 3.5*
**Step 3.2:** Set search index $k = 1$, find the first different data value from $k$
**if** $Data[i].value$ != $Data[k].value$ **then**
  Record $Data[i-1].time$ and $Data[i-1].value$ (offset time and value of previous data), and get {T[$Data[i-1].time$], V[$Data[i-1].value$]}.
**end**
Then, set search index $k$ = i, find the first different data value from $k$
**if** $Data[h].time$ != $Data[k].value$) **then**
  get {T[$Data[i-1].time$, $Data[h-1].time$], V[$Data[i-1].value$, $Data[h-1].value$]}.
**end**
Loop execution until the last data is processed, if the different data value is the last $Data[N].value$, record its time offset and value, get {T[$Data[i-1].time$, $Data[h-1].time$, ..., $Data[N].time$], V[$Data[i-1].value$, $Data[h-1].value$, $Data[N].value$]}, and the process is end.
**Step 3.3:** Compare the number of true (1) and false (0), get the smaller one and its each time offset $TimeOffset[time]$, and then get {T[$Data[TimeOffset[1]].time$, ..., $Data[TimeOffset[TimeOffset.Count]].time$], V[$Data[Index[1]].value$]}.
**Step 3.4:** Encode the *string variables*, when the number of encode type = 2, go to *Step 3.3*, otherwise, go to *Step 3.2*
**Step 3.5:** Do not compress.
**Step 4:** Get the next N time series data from the data block and repeat *Step 3.1* to *Step 3.5*.
**Step 5:** Repeat *Step 4* until all data is compressed.

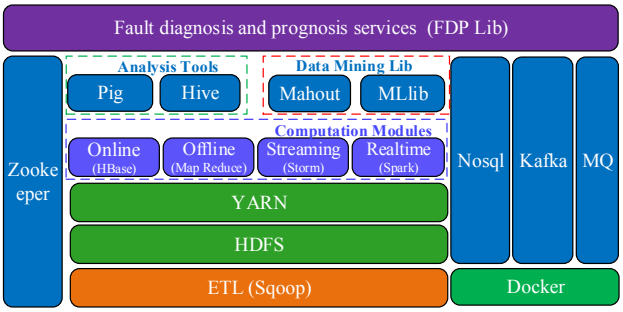


Fig. 4. Big data analysis framework for high-speed train Computing Framework

*Data analytics:* Based on the proposed computing framework, this module aims at implementing data query, statistics, and analytics to support fault modeling, diagnosis and prognosis. The operation data of high-speed trains are typical of time series data, so the Data Analytics module needs to provide analysis solutions targeted for this pattern. This includes feature extraction, mining, retrieval, clustering, and classification of time series data, as well as fault feature extraction, pattern matching, and configurable data query. When a fault has occurred and been marked during high-speed train operation, full characterization of the fault is supported by quick retrieval of relevant data in periods of time when the fault, or similar faults, have occurred in the cloud computing center's historical data. This provides strong support for follow-up modeling, diagnosis and prognosis for the particular fault. In addition, to improve efficiency of fault modeling and diagnosis, this module provides some basic big data modeling methods, such as *PCA*, *PLS*, *CPLS* (Qin and Zheng (2013), *K-NN* , *PLS* , and so on, in the algorithms library.

*Fault diagnosis and prognosis services:* This module provides fault diagnosis and prognosis services of high-speed train data. The framework supports multi-tenant management to ensure the isolation of services between different users. This module provides tools to help users to build and evaluate fault diagnostic models and deploy (register) them as services and provides the model download and update mechanisms from the *CFDS* to the *VFDS*, enabling synchronous updates and independent operation of the *VFDS*. Another important function of this module is to manage and monitor fault diagnosis and prognosis services. Different train units need to use different diagnostic models, and the same unit may, over time, use different diagnostic models, so multiple support services are needed to effectively manage the models. In addition, we need to monitor the quality of service (*QoS*) of each fault diagnosis and prognosis service so that we can identify problems and improve them. This module thus ensures that users are provided with highly available and reliable fault diagnosis and prognosis services. User can use this module in different ways for different purposes, for example, users can directly access model and algorithm services by invoking the API interface (*Restful API*); on the other hand, users can use this module to build fault diagnostic services.

*Performance Monitoring:* In order to facilitate users to evaluate different data modeling methods and understand cluster system resource cost, the system provides performance monitoring for the algorithm implementation process, so that users at the time of training model can better evaluate the efficiency of various modeling methods, in order to help users understand the execution efficiency of algorithm on the cluster.

*Visualization:* As a platform for big data analysis and diagnosis of high-speed train faults, the system should provide a self-driven data analysis experience; that is, users should be able to explore various conjectures and hypotheses regarding the data and quickly verify them. Therefore, not only should calculation and query speed be as quick as possible, but data visualization should be designed so that users can see query results and data overviews, allowing them to quickly derive the meaning and value of the displayed data.

## 4. CASE STUDY

In this section, a simple case study is given to illustrate how basic functions of the platform are implemented. At present, because of restrictions of the on-board system of high-speed trains, the platform is mainly used in off-line fault modeling and diagnostics of high-speed train data. Users can use this platform to build fault diagnostic models and deploy (register) as services and can also access existed model and algorithm services by invoking the API interface.

In addition to the existed models and algorithms that users can use directly, users can also build new models and algorithms based on existed models and algorithms. In the platform, users can choose different models to test their fault data set and compare performance indicators of those models, to facilitate the user to evaluate the diagnostic effect of the existed diagnostic methods on the data set.

The example dataset represents more than 33GB of Automatic Train Protection (*ATP*) system operation data from multiple high-speed trains collected at one railway station in China from April 1 to April 10 of 2015. The data was collected at a sampling rate 300ms and there are 129 data items, including sample time, train ID, driver ID, DRU information, FSC state, real speed, acceleration, EBP speed, and NBP speed.

PCR file data is transformed to store in the CFDS database by the following execution sequence: 1) operation data (a series of PCR files) is collected by on-board data recording equipment; 2) a PCR file collector automatically parses the PCR data files; 3) data of each parsed PCR data file is stored in the Redis database; 4) data in *Redis* database is cleaned; 5) data from *Redis* database is compressed and saved in the *HBase* and MySQL databases.

The total amount of data in the database is about 3.27 GB, *HBase* retains up to three versions of a column value by default to ensure the data reliable storage, so the data compression ratio is close to 30: 1, the compression algorithm is proved to be very effective.

With data thus housed in the platform, users can easily query and statistical fault-related data through the platform, e.g., users can easily query to get operation status information for a specific train in a specified period of time

before and after a fault occurs, and the result was shown in Figure 5. Of course, users can also get all operation status information for a specific fault that may occur in different trains or sensors, and so no. Moreover, we can use the platform to cluster and classify some of the key operation indices in order to observe possible faults, the result shown in Figure 6 is the clustering results for position correction information of the two on-board vital computers vc1 and vc2 of high-speed train. User can invoke some built-in algorithms to help them quickly start the study, such as, *PCA*, *PLS*.

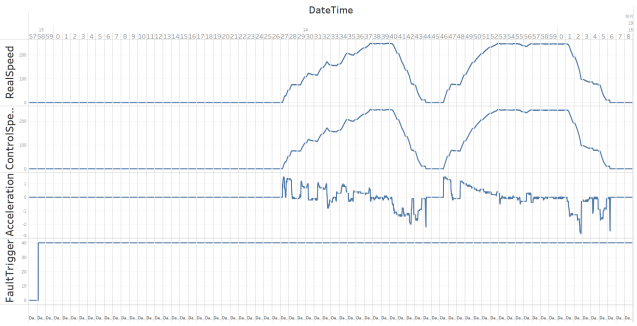


Fig. 5. Operation status information for a specific train in a specified period of time before and after a fault occurs.

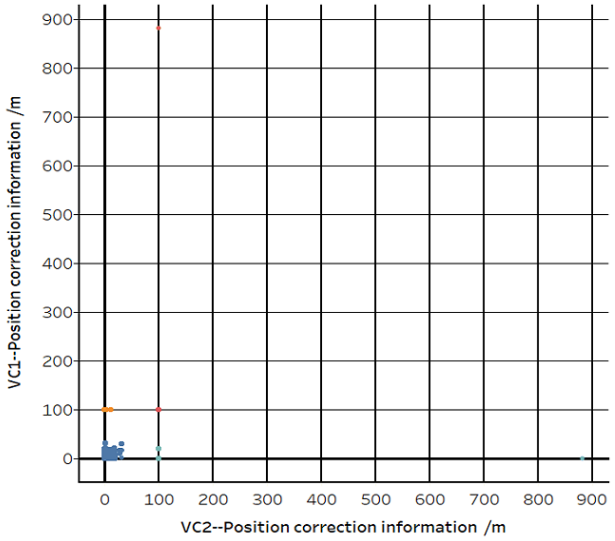


Fig. 6. The clustering results for position correction information of the two on-board vital computers vc1 and vc2 of high-speed train (177560 points).

## 5. CONCLUSION AND FUTURE WORK

The data generated by high-speed train operations are increasing day by day and the data have typical characteristics of big data. Big data and cloud computing technologies have been rapidly developed in the last few years and are shown great value. To fully utilize advantages offered by big data and cloud computing, we propose a novel software framework that supports the development, management, and real-time use of fault detection algorithms on the big data of high-speed train operation. The framework combines cloud computing and edge computing and uses two-level architecture for the characteristics of high-speed train system. The framework provides services and methods for handling the massive, high volumetric rate of data generated in real-time by high speed trains, including data import, storage, and management methods, data analytics and visualization functions, and an algorithm library.

With the data handling and analysis methods in the framework, users can focus on the problem of identifying fault models and applying these to real-time high-speed train operation, a pressing need to avoid disastrous consequences that can occur when operating faults occur. Implementing platforms such as the proposed framework are a step toward safer high-speed train operation by enabling the detection of operational faults timely to ensure the passenger and crew safety, prevent the disruption of train network traffic, and avert loss of railway equipment and facilities.

Although the amount of test data is relatively modest, the some of the functions of the platform has been well verified. In the future work, we will collect TB or even PB level data from our project partners to fully validate the platform, including the effectiveness of the two-level architecture of the platform, each function of the cloud-based fault diagnosis system and the on-board fault diagnosis system.

## REFERENCES

Bughin, J., Chui, M., and Manyika, J. (2010). Clouds, big data, and smart assets: ten tech- enabled business trends to watch. *McKinsey Quarterly, McKinsey Global Institute*.

Chang, F., Dean, J., Ghemawat, S., Hsieh, W., Wallach, D., Burrows, M., Chandra, T., Fikes, A., and Gruber, R. (2006). Bigtable: A distributed storage system for structured data. *OSDI'06*, 10, 205–218.

Jia, H. and Li, L. (2014). Thinking on improving the manufacturing level and operation quality of high speed train in china. *Chinese Railways*, 1, 30–33.

Jiang, L., Cai, H., Jiang, Z., Bu, F., and Xu, B. (2014). An iot-oriented data storage framework in cloud computing platform. *IEEE Transactions on Industrial Informatics*, 10, 1443–1451.

Liu, Q., Zhu, Q., Qin, S., and Xu, Q. (2016). A comparison study of data-driven projection to latent structures modeling and monitoring methods on high-speed train operation. *CCC'16, 2016, China*, 6734–6739.

Pedro, G., Alberto, M., E., D., Anwitaman, D., Teruo, H., Adriana, I., Marinho, B., Pascal, F., and Etienne, R. (2015). Edge-centric computing: Vision and challenges. *SIGCOMM Comput. Commun. Rev.*, 45(5), 37–42.

Qin, S. (2012). Survey on data-driven industrial process monitoring and diagnosis. *Annual Reviews in Control*, 36(2), 220–234.

Qin, S. and Zheng, Y. (2013). Quality-relevant and process-relevant fault monitoring with concurrent projection to latent structures. *AIChE Journal*, 59(2), 496–504.

Shao, Y., Liu, R., Wang, F., and Chen, M. (2013). Research on big data management for high-speed railway equipment. *Applied Mechanics and Materials*, 462-463, 405–409.

Thaduri, A., Galar, D., and Kumar, U. (2015). Railway assets: A potential domain for big data analytics. *INNS'15, USA*, 457–467.