2017 International Conference on Sensing, Diagnostics, Prognostics, and Control

Aircraft Health Management based on Big Data Flow and Fault Rules Network

Guigang Zhang, Sujie Li, Jian Wang
Institute of Automation Chinese Academy of Sciences, Beijing China, 100190
E-mail: {guigang.zhang, sujie.li, jian.wang}@ia.ac.cn

Abstract—**Big data plays an increasingly important role in aircraft integrated health management. This paper introduces the aircraft health management cloud platform, fault feature extraction methods based on deep learning and fault rules network construction, optimization and division.**

Keywords-Air craft Health management, Big data flow, Fault rules network, Deep learning, Rules

I. INTRODUCTION

With the Internet and Internet of Things applications more and more widely, more and more time-series big data streams were produced. The time-series big data stream of the aircraft fault diagnosis includes time-series monitoring data of tens of thousands of aircraft sensors. These time-series big data stream are the dynamic rule triggers that establish various decision support systems. For example, once the aircraft engine monitoring data changes, it is possible to trigger the rules of various fault modes. Then we can determine the aircraft fault mode in time, the engine maintenance personnel can decide what measures to take.

A core premise of the aircraft health management monitoring system under the big data stream is to construct a rule processing system with fast, accurate and real-time timing data stream. The airline should realizes the health of the engine in real time, airline maintenance personnel should keep abreast of the health of hundreds of aircraft, and make real-time decision-making. The great challenges of the regular processing include a huge amount of data processing, dynamic changes, and the processing of real-time requirements. And the current rule processing research and a variety of rules-based decision support system is basically still in dealing with a small amount of time-series data stream, or deal with big data but the low real-time requirements (regular rule processing wait).

The rest of the paper is organized as follows. In Section II, we summarize the related work.

II. RELATED WORK

In the following, we will summarize the research work on the parallel processing of massive rules under the time-series big data stream.

A. Rule Processing Theory and Algorithm under Time-Series Data Stream

In the decision support system based on the time-series big data stream, it is very important to deal with the rules effectively. Rule engine plays an indispensable role in the processing of rules. In the judgment of facts, the rule engine contains three phases: matching, selecting, and executing [1]. In the traditional rule pattern matching algorithms, the RETE algorithm [2] [3], the TREAT algorithm [4] and the LEAPS algorithm and the RETE2 algorithm are the most typical.

The RETE algorithm is an efficient pattern matching algorithm used to implement a production rule system. The algorithm has been applied to most of the rules engines, such as Drools [5] [6], ILOG, and HAL [7] [8], as well as corresponding application systems.

The research of rule processing theory and algorithm is developing in two directions: active rule engine and big data processing. In order to improve the efficiency of rule engine processing, it is used to optimize the rules before using the preprocessing method [9]. When we face some complex event processing requirements, we use the association rules and a variety of effective algorithms to deal with the rules for further rule mining and processing [9] [10]. For complex large-scale rule processing [12] [13], especially in the face of big data processing, the current research is mainly through the complex rules of the model mining model optimization [14], the use of a variety of machine learning algorithm [15] to optimize the rule processing process and automatically establish the processing mechanism, and using parallel processing mechanism [16].

B. Semantic rule description language

At present, the study of rule description language, basically still use simple IF, ..., THEN, ... In general, the rule language can be divided into structured and Markup, usually xml. The main rule describe languages as follows. SRL (Structured Rule Language): A structured rule description language defined by Fair Isaac [17]; DRL (Drools Rule Language): A structured rule description language defined by Jboss [18]; RuleML (Rule Markup Language): The xml rule description language defined by www.ruleml.org [19]; SRML (Simple Rule Markup Language): The xml rule description language [20]; BRML (Business Rules Markup Language): The xml rule description language; SWRL（A Semantic Web Rule Language): The xml rule description language defined by www.daml.org and SPL+ language [21] [22].

In addition to the above rule description language, the most typical and better semantic rule language is Semantic Objects Description Language. The language is based on the invention patent of Prof. Phillip Sheu and NEC Corporation of Japan, and use it to describe more complex semantic rules [23] [24].

C. Theory of Graph Cutting

In this paper, we need to divide the rule network into k parts by using the method of graph cutting, and finally allocate them to the k processors for parallel processing. Thus, in essence, it is the problem of graph cutting [25].

The cutting of the graph uses hash algorithm. Although the algorithm can satisfy the approximate equilibrium of the computational scale of the k-subgraph, but in fact, the

978-1-5090-4020-9/17 $31.00 © 2017 IEEE
DOI 10.1109/SDPC.2017.96

algorithm has great limitations, it does not consider the data communication between subgraphs. So it will have a huge communication overhead, affecting the operation of large-scale map processing platform efficiency.

Numerous studies have transformed the problem of graph cutting into a mathematical problem, namely, The Max-Flow and Min-Cut Problem.

The cutting of the graph is divided into the cutting of the static map and the cutting of the dynamic graph. With the emergence of some special applications in recent years, especially for time-series big data stream, the cutting of dynamic graph has become increasingly important, and gradually become a hot spot for researchers [26] [27].

I. Pandis and A. Pavlo argue that the final goal of the graph cutting is to analyze the balance between the workload and the amount of data, and finally assign them to the k-subgraph and perform the maximum parallelization and calculation [28] [29].

The graph cutting algorithm includes Metis [30], Schism [31], Sword [32], JECB [33], PLP [28], Accordion [34], and E-Store [35].

According to the analysis of the graph cutting algorithms, we can see that some algorithms only support the static graph cutting, but does not support the dynamic graph cutting; some algorithms only consider the task equilibrium distribution, but did not consider the communication costs between subgraphs; some algorithms cannot better support the workload balance problem of the subgraph; some algorithms can support the workload balance, but cannot solve the problem of access to hot dependent subgraph blocks.

So, there is no better algorithm to support the rule network cutting. Therefore, we need to study a dynamic data cutting that supports the regular network under time-series big data stream. At the same time, it is necessary to consider the minimum communication cost between the subgraphs, the balance between the processor tasks, and reduce the dynamic imbalance caused by sub-block block access.

III. Cloud Platform of Aircraft Health Mangement

A. *The cloud platform of aircraft health management*

The Fig. 1 shows the cloud platform of aircraft health management.

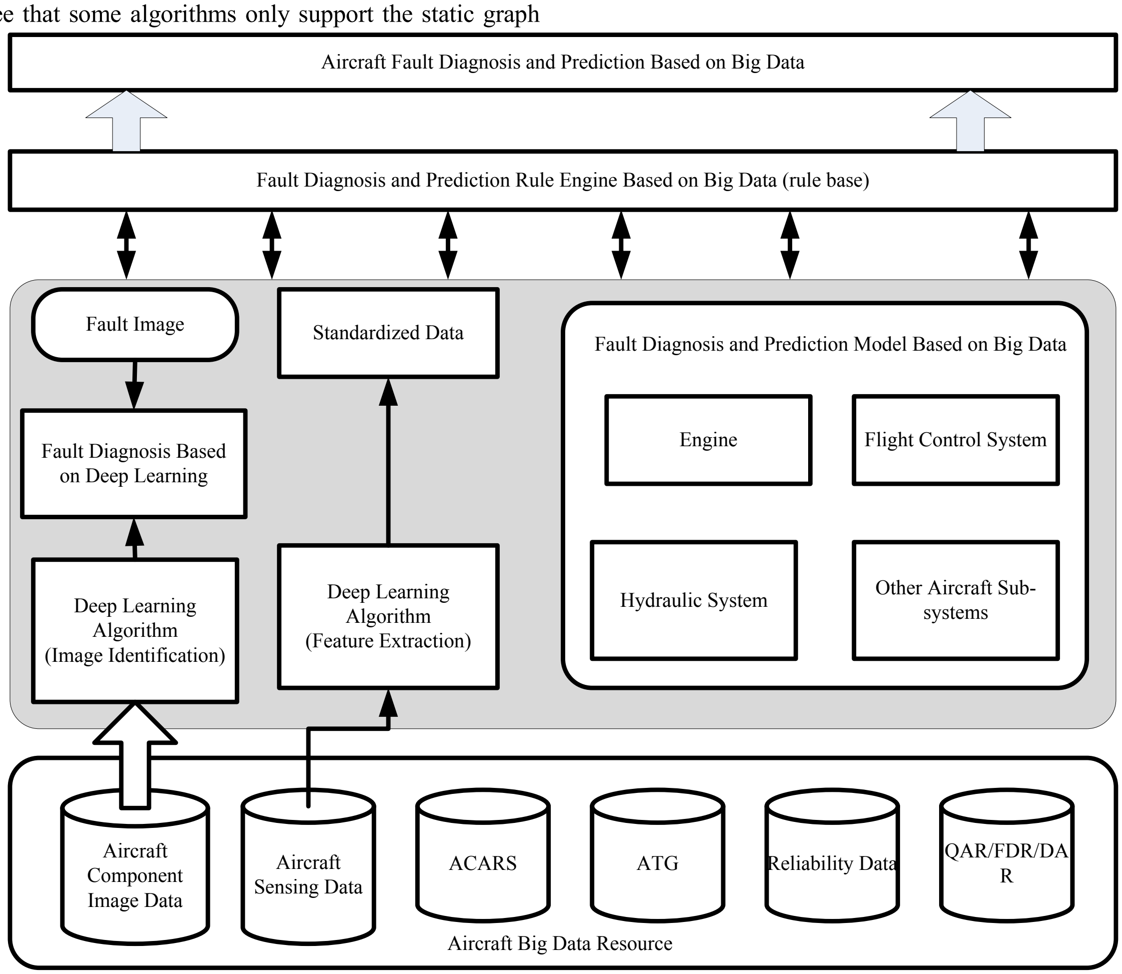

Figure 1. Cloud Platform of Aircraft Health Management

The cloud platform of aircraft health management mainly includes aircraft data layer, aircraft health management layer, rule engine layer, man-machine interface layer. The data layer mainly includes aircraft component image data (such as engine blade damage images), aircraft sensor data, ACARS / QAR / FDR data.

The main functions of the health management are Fault diagnosis and fault prediction. For the sensor data, an important problem is feature extraction. Through feature extraction to determine whether the violation of the failure mode rules and to perform fault diagnosis. For the image data, an important issue is to image recognition. Through the image recognition to determine whether the violation of the failure mode rules and to perform fault diagnosis. The other parts, through establish models for the various aircraft subsystems, combined with big data and algorithms to determine whether the violation of the rules and to perform fault diagnosis.

The rule engine layer refers to the generation of rule base, the optimization of rule network, the cutting of rule network and the processing of rule network. The rule base is based on fault diagnosis mode or fault prediction mode.

The man-machine interface layer presents the results to the aircraft health management supervisor.

B. Deep learning feature extraction and recognition

1) *Feature extraction algorithm based on autocoder*

The autocoder is a neural network that reconstructs the input signal as much as possible. In order to reconstruct the original signal, the autocoder must learn the most important intrinsic factors that can represent the input data. The core idea is to give a neural network, assuming its output and input are the same. By training the network, you can get the weight in each layer. Correspondingly, several different representations of the input samples can be obtained. That is, each layer represents a representation. These representations are features.

The training process is as follows.

- Enter the sample into an encoder;
- Get the output CODE;
- Use the output CODE of the first layer as the input signal of the second layer;
- Minimize the reconstruction error;
- Obtain the weight of the second layer parameters;
- Get the second encoder;
- Get the output CODE of the second layer.

The samples are progressively passed through multiple encoders, the output of the last layer of the encoder is the feature of the input signal.

2) *Image recognition based on deep learning*

For image recognition, it is not necessary to extract the artificially designed features, such as Gabor texture features, multi-scale wavelet features, and SIFT features. Using the original image, it is effective to recognize image by the convolutional neural network and the deep belief network.

Convolution neural network is a multi-layer network structure, each of which is composed of multiple feature maps, each feature map represents a feature; each feature map includes many independent neurons. Corresponding, the network layer of the convolutional neural network is divided into convolution layer and down-sampling layer. The network layer is not linear mapping. The process from the convolution to the down-sampling layer is a down-sampled process. The process from the down sampling layer to the convolution layer is a convolution filter process.

The deep belief network is one of the most widely used deep learning models. It consists of several Restricted Boltzmann Machines (RBM). In the network, each RBM is trained separately. The DBNs training process using layer-by-layer training method. The network trains one layer every time, the parameters are individually adjusted. When the layer is trained, the training results is the input of the next Layer. The process is completed until each layer is trained, and this process is also known as pre-training. Then according to the sample label value, the deep belief network uses the BP algorithm to fine-tune backwards.

Taking the engine rotor imbalance fault as an example, the rotor vibration and sound signal are collected, and the characteristics of the normal state and the fault state are extracted by the deep learning algorithm.

The engine rotor imbalance exception rules are as follows:

- IF the vibration characteristics is normal and the sound is abnormal, THEN failure.
- IF the vibration characteristics is abnormal and the sound is normal, THEN weak imbalance fault; before failure time = min (the remaining time).
- IF the vibration characteristics is normal and the sound is abnormal, THEN other failures.
- IF the vibration characteristics is normal and the sound is normal, THEN normal.

IV. ARCHITECTURE OF FAULT RULES ENGINE

The fault rule engine structure is shown in Fig. 2. Through the rule-based aircraft fault engine interface, the aircraft big data generates the fault rule network, then optimizes, divides and parallel processes the fault rules network. The results are sent to the aircraft safety monitoring personnel.

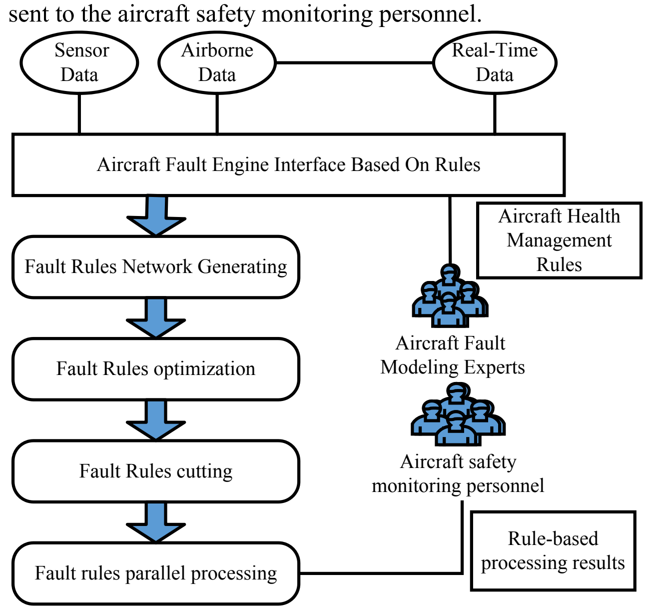

Figure 2. The fault rule engine structure

Fig. 2 shows the architecture of fault rule engine structure.

At first, there are lots of aircraft data in the big data center, which are sensor data, airborne data, QAR data and lots of real time data.

Aircraft fault modeling experts will set lots of rules based on the fault modes in the aircraft fault engine interface based on rules.

Millions of rules will generate a very big rules network. In order to save the computing time, this very big rules network will be optimized dynamically.

In order to process so big rules network, we should apply lots of processors to do this work. And so, we should cut this very big rules network in a good balance. And let each processor have about the same workload.

Finally, all processors will process this big rules network and obtain the results to the aircraft safety monitoring persons. And if there are some faults, we can make the decisions for the aircraft and arrange the maintenance so on.

V. CONCLUSION AND FUTURE WORK

Big data plays an increasingly important role in aircraft integrated health management. A core premise of the aircraft health management monitoring system under the big data stream is to construct a rule processing system with fast, accurate and real-time timing data stream. This paper introduces the fault rule generation, optimization and division in the data stream rule processing. We proposed the aircraft health management cloud platform architecture, and we studied fault feature extraction and fault image recognition based on deep learning.

In the future, we will focus on the rule network cutting method and algorithms research. The rules network cutting is a very difficult and complex problem. In the future, we try to do a good algorithm for the rules network cutting.

ACKNOWLEDGMENT

We would like to thank all colleagues and graduate students who helped for our work. This research was supported by (1) Special Project for Civil Aircraft, MIIT; (2) Fund of Shanghai Engineering Research Center of Civil Aircraft Health Monitoring.

REFERENCES

[1] R. Bhargavi; Ravi Pathak; V. Vaidehi.Dynamic complex event processing — Adaptive rule engine.2013 International Conference onRecent Trends in Information Technology (ICRTIT).Pages: 189 - 194.

[2] Forgy, C.L. Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem. Artificial Intelligence.19(1982).PP: 17-37.

[3] Tomoya Kawakami; Tomoki Yoshihisa; Yutaka Yanagisawa; Masahiko Tsukamoto.A Rule Processing Scheme Using the Rete Algorithm in Grid Topology Networks.2015 IEEE 29th International Conference on Advanced Information Networking and Applications.Pages: 674 - 679.

[4] D. P. Miranker. TREAT: A better match algorithm for AI production systems. In proceedings of AAAI 87 conference on Artificial Intelligence, August 1987. PP: 42-47.

[5] N. Alex Rupp. The logic of the bottom line: An Introduction to the Drools Project.May 2004.

[6] Andrew Fobel; Nary Subramanian.Comparison of the performance of Drools and Jena rule-based systems for event processing on the semantic web.2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA).Pages: 24 - 30.

[7] P.-Y. Lee, A. M. K. Cheng. ``HAL: A New Match Algorithm with Low Match-Time Variance," submitted to IEEE Transactions on Knowledge and Data Engineering, Sept. 1998.

[8] P.-Y. Lee, A. M. K. Cheng. ``Reducing Match Time Variance in Production Systems with HAL," Proc. 6th Intl. ACM Conf. on Information and Knowledge Management, Las Vegas, Nevada, Nov. 1997.

[9] Anand Prem Kumar; N. Ramasubramanian.Rule set optimization for packet pre-processing using hash based algorithm.2016 International Conference on Microelectronics, Computing and Communications (MicroCom).Pages: 1 - 5

[10] M. S Saravanan; R. J Rama Sree.A simple process model generation using a new association rule mining algorithm and clustering approach.2011 Third International Conference on Advanced Computing.Pages: 265 - 269.

[11] Renan de Padua; Solange Oliveira Rezende; Veronica Oliveira de Carvalho.Post-Processing Association Rules Using Networks and Transductive Learning.2014 13th International Conference on Machine Learning and Applications (ICMLA).Pages: 318 - 323.

[12] Ravi Pathak; V. Vaidehi.An efficient rule balancing for scalable complex event processing.2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE).Pages: 190 - 195.

[13] Nijat Mehdiyev; Julian Krumeich; Dirk Werth; Peter Loos.Determination of Event Patterns for Complex Event Processing Using Fuzzy Unordered Rule Induction Algorithm with Multi-objective Evolutionary Feature Subset Selection.2016 49th Hawaii International Conference on System Sciences (HICSS).Pages: 1719 - 1728.

[14] Deepak Kumar Sharma; Naveen Prakash; Himani Sharma; Dheerendra Singh.Automatic construction of process template from business rule.2014 Seventh International Conference on Contemporary Computing (IC3).Pages: 419 - 424.

[15] Zhe Zhou; Chenglin Wen; Chunjie Yang.Fault Detection Using Random Projections and k-Nearest Neighbor Rule for Semiconductor Manufacturing Processes.IEEE Transactions on Semiconductor Manufacturing.Year: 2015, Volume: 28, Issue: 1.Pages: 70 - 79.

[16] Jo Ryan Basco; Bobby D. Gerardo; Cyreneo Dofitas Jr.; Yung-Cheol Byun.A Rule-based Parallel Processing to Speed-Up an Application.2012 IEEE 14th International Conference on Commerce and Enterprise Computing.Pages: 144 - 146.

[17] James Taylor, "WHY IS RULE MANAGEMENT TECHNOLOGY "REVOLUTIONARY, Part 2"," Business Rules Journal, Vol. 3, No. 3, (March 2002)

[18] DRL.https://docs.jboss.org/drools/release/5.2.0.Final/drools-expert-docs/html/ch05.html（2017）

[19] RuleML.http://2016.ruleml.org/(2017)

[20] SRML.http://xml.coverpages.org/srml.html(2017)

[21] SWRL.https://www.w3.org/Submission/SWRL/(2017)

[22] Guigang Zhang; Shu Wang; Xu ChengZhi; Zhiyuan Gong; Phillip C-Y SheuA Semantic Programming Language SPL+ - A Preliminary Report.20th IEEE International Conference on Tools with Artificial Intelligence(ICTAI2008).Pages: 274-281.

[23] Phillip C.-Y. Sheu, Atsushi Kitazawa:From Semanticobjects to Semantic Software Engineering. Int. J. Semantic Computing 1(1): 11-28 (2007).

[24] Phillip C.-Y. Sheu.Semantic Computing. Intelligent Information Processing 2008: 1-2.

[25] Marco Serafini, Rebecca Taft, Aaron J. Elmore,Andrew Pavlo, Ashraf Aboulnaga, Michael Stonebraker.Clay: FineGrained Adaptive Partitioning for General Database Schemas.Proceedings of the VLDB Endowment, 10(4):445–456, 2016.

[26] A.Jindal and J. Dittrich. Relax and let the database do the partitioning online.In International Workshop on Enabling Real-Time Business Intelligence, pages 65–80. 2012.

[27] S. Yang, X. Yan, B. Zong, and A. Khan. Towards effective partition management for large graphs. In ACM SIGMOD International Conference on Management of Data, pages 517–528, 2012.

[28] I. Pandis, P. Tözün, R. Johnson, and A. Ailamaki. PLP: Page latch-free shared-everything OLTP. Proceedings of the VLDB Endowment, 4(10):610–621, 2011.

[29] A. Pavlo, C. Curino, and S. Zdonik. Skew-aware automatic database partitioning in shared-nothing, parallel OLTP systems. In ACM SIGMOD International Conference on Management of Data, pages 61–72, 2012.

[30] G. Karypis and V. Kumar. Metis-unstructured graph partitioning and sparse matrix ordering system, version 5.0. http://glaros.dtc.umn.edu/gkhome/metis/metis/overview.

[31] C. Curino, E. Jones, Y. Zhang, and S. Madden. Schism: A workload-driven approach to database replication and partitioning. Proceedings of the VLDB Endowment, 3(1-2):48–57, 2010.

[32] A. Quamar, K. A. Kumar, and A. Deshpande. Sword: scalable workload-aware data placement for transactional workloads. In International Conference on Extending Database Technology, pages 430–441, 2013.

[33] K. Q. Tran, J. F. Naughton, B. Sundarmurthy, and D. Tsirogiannis. JECB: A join-extension, code-based approach to OLTP data partitioning. In ACM SIGMOD International Conference on Management of Data, pages 39–50, 2014.

[34] M. Serafini, E. Mansour, A. Aboulnaga, K. Salem, T. Rafiq, and U. F. Minhas. Accordion: Elastic scalability for database systems supporting distributed transactions. Proceedings of the VLDB Endowment, 7(12):1035–1046, 2014.

[35] R. Taft, E. Mansour, M. Serafini, J. Duggan, A. J. Elmore, A. Aboulnaga,A. Pavlo, and M. Stonebraker. E-store: Fine-grained elastic partitioning for distributed transaction processing systems. Proceedings of the VLDB Endowment, 8(3):245–256, 2014.