J Intell Manuf (2017) 28:1125–1137
DOI 10.1007/s10845-015-1066-0

CrossMark

A new paradigm of cloud-based predictive maintenance for intelligent manufacturing

Jinjiang Wang[1] · Laibin Zhang[1] · Lixiang Duan[1] · Robert X. Gao[2]

Received: 30 November 2014 / Accepted: 13 March 2015 / Published online: 19 March 2015

Abstract Advances in cloud computing reshape the manufacturing industry into dynamically scalable, on-demand service oriented, and highly distributed cost-efficient business model. However it also poses challenges such as reliability, availability, adaptability, and safety on machines and processes across spatial boundaries. To address these challenges, this paper investigates a cloud-based paradigm of predictive maintenance based on mobile agent to enable timely information acquisition, sharing and utilization for improved accuracy and reliability in fault diagnosis, remaining service life prediction, and maintenance scheduling. In the new paradigm, a low-cost cloud sensing and computing node is firstly developed with embedded Linux operating system, mobile agent middleware, and open source numerical libraries. Information sharing and interaction is achieved by mobile agent to distribute the analysis algorithms to cloud sensing and computing node to locally process data and share analysis results. Comparing to the commonly used client–server paradigm, the mobile agent approach enhances the system flexibility and adaptability, reduces raw data transmission, and instantaneously responds to dynamic changes of operations and tasks. Finally, the presented cloud-based paradigm of predictive maintenance is validated on a motor tested system.

Keywords Cloud computing · Cloud manufacturing · Mobile agent · Predictive maintenance

✉ Jinjiang Wang
jiw09005@foxmail.com

[1] School of Mechanical and Transportation Engineering, China University of Petroleum, Beijing 102249, China

[2] Department of Mechanical Engineering, University of Connecticut, Mansfield, CT 06269, USA

Introduction

The emerging of cloud computing (Armbrust et al. 2010; Venters and Whitley 2012; Wang et al. 2014) has created new opportunities for the manufacturing industry, which is undergoing a major transformation named as cloud manufacturing. Cloud manufacturing is a recently proposed manufacturing model (Xu 2012), with characteristics of on-demand service-oriented, customer centric, dynamically scalable and reconfigurable, and distributive collaboration (Wu et al. 2013). Generally, such benefits are dependent on trouble-free operations of the various machine elements (Teti et al. 2010).

During the past several decades, a significant amount of research has been undertaken to develop maintenance strategies such as breakdown maintenance, preventive maintenance, and condition based maintenance including models and algorithms in manufacturing (Jardine et al. 2006; Chouikhi et al. 2014). With the rapid development of predictive science, prognosis as a valuable tool, is introduced to predict remaining service life of machinery and machining tools (Peng et al. 2010; Wang et al. 2013). It leads to a more efficient maintenance strategy as predictive maintenance. It involves condition monitoring, fault diagnosis, remaining service life prediction, and maintenance plans by providing scientific and technological information basis for the decision making.

In the predictive maintenance, all condition measurements are usually collected and transmitted to a centralized server, and then the measurements are processed in a centralized server based on models and algorithms for fault diagnosis, prognosis and maintenance scheduling (Arab et al. 2013; Zhang et al. 2013). Inspired by the success of cloud computing, a cloud based predictive maintenance (Yang et al. 2014; Wang 2013) is envisioned as shown in

Vlt in Vortrag?

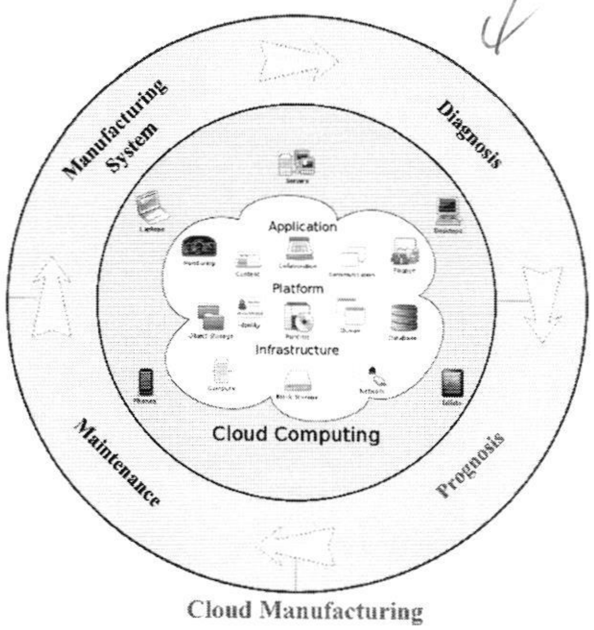

Fig. 1 Cloud predictive maintenance as an integral part of cloud manufacturing

Fig. 1. As a vital part of cloud manufacturing, it is a new-generation service-oriented technology to support multiple enterprises to deploy and manage predictive maintenance services including machinery condition monitoring, data analysis and diagnosis, prognosis, and maintenance planning over the internet. It surely benefits from the ability of sharing data and knowledge on machines and processes among different applications and locations, seamlessly and collaboratively, by adopting the cloud techniques.

It is meaningful and of great significance to develop cloud-based predictive maintenance at machine and execution control level to improve the reliability, availability, and safety of cloud manufacturing. Research on cloud-based predictive maintenance is still at its infancy. A number of research challenges remains open including (Zhang et al. 2010): (1) automated service provisioning: it has long been an issue to allocate resources from the cloud to satisfy its service while reducing the operational cost; (2) energy management: it has become a major issue to design energy-efficient data centers and improve system's energy efficiency; (3) traffic management and analysis: how to manage cloud data and reduce data traffic over the cloud is an important issue; (4) storage technologies and data management: software frameworks can introduce compatibility issues with legacy file system and applications, due to unstandardized interface and storage structure; (5) novel cloud architectures: large data centers come with the limitations of high energy expense and high initial investment for constructing data centers. Implementation of small data centers, especially mobile cloud centers remains to be a challenge.

To address the aforementioned challenges, this paper presents a new paradigm of cloud predictive maintenance based on mobile agent technology. To adaptively allocate resources, an embedded cloud sensing and computing node is firstly developed. It is instrumented with embedded Linux operating system, mobile agent middleware, and open source numerical libraries. To reduce data traffic, improve data management and system compatibility, mobile agent technology (Bradshaw 1997) is investigated by distributing the analysis algorithms to cloud nodes instead of transmitting raw measurements for information sharing and interaction. The presented cloud-based paradigm of predictive maintenance is validated on a motor tested system.

The rest of paper is constructed as follows. In "Mobile agent technology" section, theoretical background of mobile agent technology is briefly introduced. Next, a cloud sensing and computing node to support mobile cloud service is developed and instrumented with embedded Linux operating system, mobile agent middleware, and open source numerical libraries as discussed in "Development of cloud sensing and computing node" section. A mobile agent based cloud predictive maintenance paradigm is presented in "Mobile agent based cloud maintenance framework" section. In "Experimental study" section, experimental study on a motor tested system is used to demonstrate the effectiveness of the presented method. Next, the strength, practical issues and challenges of the presented mobile agent based paradigm are briefly discussed in "Discussions" section. The conclusions are finally drawn in last section.

Mobile agent technology

Mobile agent, emerging from the distributed artificial intelligence field, is characterized by distributively executing the tasks in parallel based on autonomous entities, named as agents (Cucurull et al. 2009). Agent based technology has been applied in a variety of different areas, such as e-business, transportation, process control, telecommunications, health care, semiconductor industry, workflow management, routing and resource scheduling (Tripathi et al. 2002; Monostori et al. 2006; Chen and Wang 2014; Hsieh and Lin 2014; Archimede et al. 2014; Bandyopadhyay and Bhattacharya 2015). Mobile agent is a software program with mobility, which means it is able to migrate across the network. Agents organize themselves into a heterarchical structure characterized by the high-level autonomy and co-operation. The scalability of the system means it is easy to modify the functionalities of some agents or add some new agents to adapt the changes of the system. Thus, the important properties of an agent are autonomy, intelligence, adaptation,

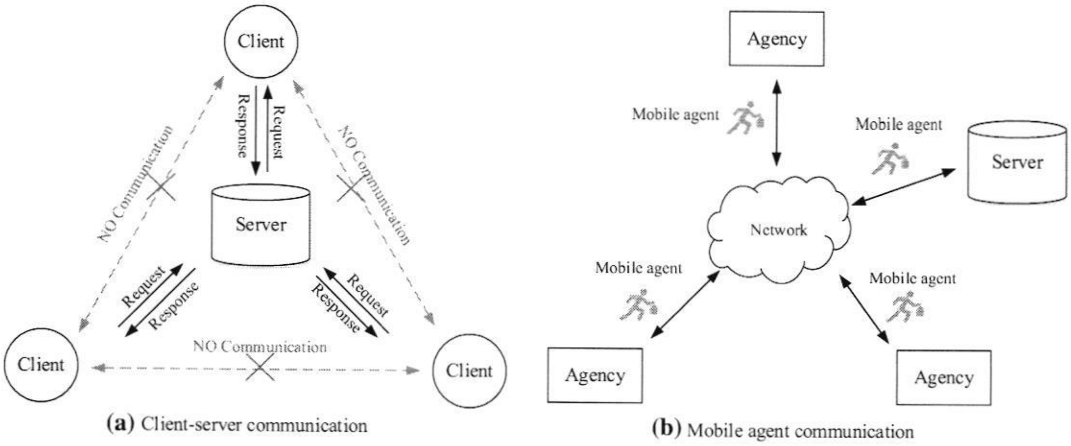

Fig. 2 Topology illustration of client–server paradigm and mobile agent paradigm

reconfigurability and co-operation; thus mobile agent is very suitable for cloud computing paradigm (Singh and Malhotra 2012).

Agent technology can significantly enhance the design and analysis of problems under the following scenarios: (1) the application domain is geographically distributed (e.g., the factories distributed in the world), (2) the subsystem exists in a dynamic reconfigurable environment (e.g., the machines need to be reconfigurable to perform different functionalities), (3) the subsystem needs to interact with each other more flexible (e.g., machine-to-machine communication in cloud manufacturing to enhance efficiency and information sharing). Mobile agent attains a better, more efficient, and flexible mode of communication in a distributed dynamic changing environment comparing with the client–server paradigm. Figure 2 demonstrates the differences between the mobile agent paradigm and the client–server paradigm in terms of communication.

As seen in the client–server paradigm, the client has to maintain a communication link with the server (e.g., service provider) for information transmission by requesting and responding scheme. Since no communication link is built among the clients, the communication among the clients has to go through the server. All data are transmitted and stored in the central server, thus it is easy to manage the data and control the data security.

In mobile agent paradigm, the fixed relations are no more applied. A mobile agent, carrying the designated tasks and functionalities, can be sent out from an agency into a network and roam among the agencies and information servers. It can be executed on those agencies and information servers to finish its task on behalf of its owner. The ability of migrating from one agency to another and processing functions to a remote agency offers the potential benefits of reduced network traffic and bandwidth requirements. Thus, mobile agents provide several advantages over other distributed computing models, such as the client–server paradigm, including: (a) Agents can provide better support for the clients, especially for the clients with intermittent connection to a network; (b) Agent based communication can be more robust. Asynchronous communication in mobile agent paradigm provides reliable transport between the agency and the server without requiring reliable communication. Moreover the mobile agent is capable of providing services when a server is unavailable; (c) Mobile agents can be dynamically created during runtime and dispatched to the network to perform new tasks with the updated code. In the client–server paradigm, the functions are all predefined, and are difficult to upgrade. Therefore, the mobility of mobile agents provides distributed applications with significant flexibility and adaptability which are both essential to satisfy the dynamically changing requirements in a distributed environment; (d) Agents exhibit intelligence from applying fixed rules to the capabilities of reasoning, planning and learning. In volatile and dynamic scenarios, mobile agents learn to adapt their behaviors to the dynamic environments, by acquiring new knowledge and skills to improve their performance. However, since the data are distributed in remote nodes, it also poses challenges on resources management and data security control. The advantages and disadvantages of mobile agent paradigm and client–server paradigm are summarized in Table 1.

Therefore, mobile agents can provide distributed applications with significant flexibility and adaptability for cloud manufacturing in a dynamically changing environment. Increasing interest in agent technology leads to numerous mobile agent systems being developed. Several notable mobile agent systems including Mole, Aglets, Concordia, D'Agents, Ara, TACOMA, JADE, and Mobile-C have been reported in the literature (Baumann et al. 2002; Lange and

Table 1 Comparison between mobile agent paradigm and client–server paradigm

	Client–server paradigm	Mobile agent paradigm
Resource management	Easy	Difficult
Relationship	Fixed	Dynamic
Reconfigurability	Low	High
Scalability	Difficult	Easy
Data traffic	Heavy	Low
Data security control	Easy	Difficult
Algorithm mobility	No	Yes

Oshima 1998; Wong et al. 1997; Bellifemine et al. 2008; Gray et al. 2002; Peine 2002; Johnansen et al. 2002; Chou et al. 2009). The first six mobile agent systems are not compliant to agent standards, whereas JADE and Mobile-C are two IEEE FIPA compliant agent systems. Mobile-C (Chou et al. 2009) has a small footprint and supports multiple platforms. In Mobile-C, the agents are coded using C/C++ language, and are easy to be integrated into resource-constrained systems with interfaces to the hardware, such as embedded systems. In addition, Mobile-C uses an embedded C/C++ interpreter—Ch (Cheng 2006b), to support the execution of Mobile-C agent code. Thus, Mobile-C is selected as the mobile agent system to develop a cloud-based predictive maintenance paradigm due to its prevalence in mobile agent computing and support for multiple platforms.

Development of cloud sensing and computing node

To enable timely acquisition of machinery condition and pervasive computing in cloud manufacturing, an embedded cloud sensing and computing node with high computation power is developed. It is instrumented with embedded Linux operating system, mobile agent middleware, and open source numerical libraries to facilitate the development of cloud predictive maintenance algorithms.

The cloud sensing and computing node is designed with three boards as shown in Fig. 3. The volume of the cloud node is about $4 \times 2.4 \times 0.65$ in.^{3}. The top one is a wireless communication board, while the middle one is a computing board, and the sensing board is located in the bottom. Three boards are connected together through predesigned connectors (Chen and Wang 2008). The computing board is implemented using a finger size embedded computer called Gumstix. The Gumstix board communicates with the sensing board through I^{2}C bus, and connects to the wireless communication board through a parallel port. The hardware architecture of cloud sensing and computing node is detailed shown in Fig. 4.

The high computational power of the cloud node is achieved through the integration of hardware computing resources and the embedded numerical computing software packages. The Gumstix embedded computer is one of the world's smallest full function computers with a size of 20 mm × 80 mm × 8 mm. It uses an Intel Xscale type processor PXA-255 installed with an embedded Linux operating system. The CPU's operating speed can reach up to 600 MHz. Various memory resources are also instrumented in Gumstix including 128 MB RAM, 32 MB flash memory, and ample external memory spaces provided by a Type II compact Flash adapter. These memory spaces are directly accessible through Gumstix file system for computation and data storage.

A customized sensing board is designed and fabricated to meet the sensing requirements in machinery condition monitoring applications. A multimodal sensing approach by incorporating active sensing with passive sensing to achieve a better monitoring result. The sensing board consists of an Atmega128L CPU for real-time data acquisition and communication with the multiple devices such as Gumstix computing board, 16-bit analog to digital converter (ADC), signal conditioning circuits (e.g., signal processing of accelerometer and strain gage), an active sensing sig-

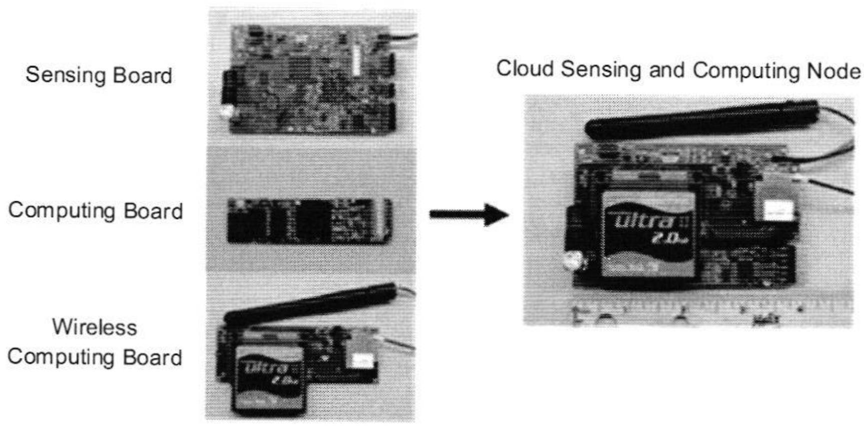

Fig. 3 The developed cloud sensing and computing node

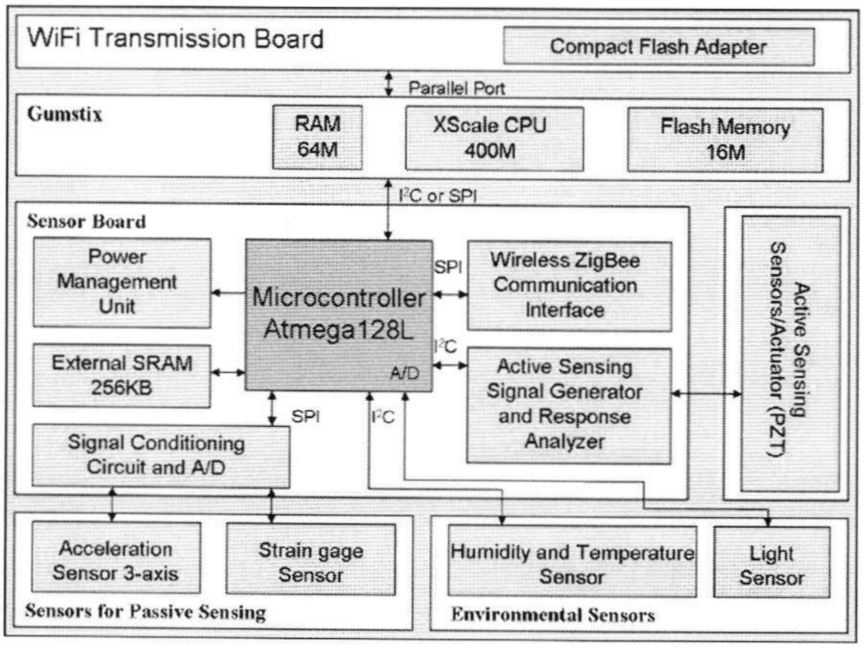

Fig. 4 Hardware architecture of cloud sensing and computing node

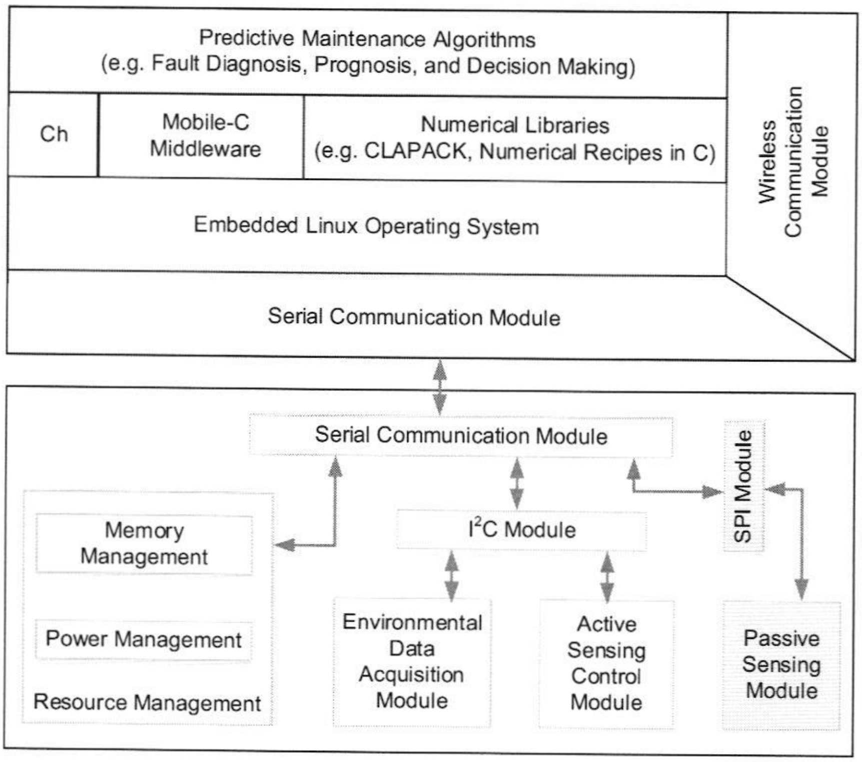

Fig. 5 Software architecture of cloud sensing and computing node

nal generator and response analyzer (e.g., active sensing with Piezoelectric transducer), and an external static random access memory (SRAM) (e.g., real-time data buffering).

To facilitate the implementation of predictive maintenance algorithms on cloud sensing and computing node, a number of numerical libraries are integrated. Thanks to the open source software packages Ch, CLAPACK, and Numerical Recipes in C (Chen and Wang 2008), which make it easy to perform predictive maintenance algorithms on cloud nodes and build an open software architecture. A two-layer software architecture of cloud sensing and computing node is shown in Fig. 5.

In the lower layer, sensor data acquisition modules are implemented in the microcontroller of sensing board. The lower layer embedded software manages data acquisition and sensing board communication. The passive data acquisition is handled in a timer interrupter processing module. Active sensing control module uses I^2C serial communication to transmit data and send commands, while passive sensing module communicates with the microcontroller via serial peripheral interface (SPI) bus. The acquisition module of environmental sensors, such as temperature and humidity, uses I^2C to communicate with the Microcontroller.

In the upper layer, Ch is an embeddable C/C++ interpreter which supports matrix computation and provides a set of high-level numerical analysis functions for data analysis. Ch is also the execution engine of mobile agents Mobile-C selected in this work. The CLAPACK library is a C version of LAPACK library which provides routines for solving systems of linear equations, linear least-squares problems, eigenvalue problems, and singular value problems. All the functions support real and complex matrices, in both single and double precision (Anderson et al. 1999). Numerical Recipes in C is another good tool for people who program in C and work with mathematics (Press et al. 1992). It covers a wide range of algorithms from solving systems of linear equations to determining eigenvectors and singular value decompositions, solving differential equations, and calculating typical signal processing techniques (e.g., Fast Fourier Transforms) for implementation of predictive maintenance algorithms.

Mobile agent based cloud maintenance framework

Erklärung über Framework

Cloud manufacturing implies an integrated cyber-physical system that can provide on-demand manufacturing services, digitally and physically, to best utilize the manufacturing resources (Wang et al. 2014). It aims at offering a shared pool of resources such as manufacturing software tools, manufacturing facilities, and manufacturing capabilities. Operational reliability of industrial machines or assets has significant influences on increasing service availability, multi-task coordination, high product quality, flexibility and productivity in industries. But as the service time increases, operational reliability would decrease; thus assuring a satisfactory level of reliability during the useful life of machines becomes a primary task.

Motivated by the great potential of cloud computing and cloud manufacturing, a new paradigm of cloud-based predictive maintenance based on mobile agent is presented and investigated for better utilization of cloud services and manufacturing resources. It is a new-generation service-oriented technology to support multiple enterprises to deploy and manage predictive maintenance services over the internet. Figure 6 illustrates mobile agent based cloud predictive maintenance in the cycle of cloud manufacturing, including independent parts of physical manufacturing system, diagnosis, prognosis, and maintenance.

Cloud sensing and computing node is deployed in the distributed manufacturing processes, and monitors the health status of machineries based on condition measurements without interrupting the normal operations. It also provides the cloud computing service with high computational capability. Such services are coded in mobile agents including fault diagnosis, prognosis, and maintenance planning. Fault diagnosis provides information such as whether there are abnormal behaviors of a manufacturing process, if yes, where the root-locations of failures are and what is the failure type for decision making and maintenance actions. Today, with the rapid development of predictive science, prognosis as a valuable tool, moves toward a more efficient maintenance approach: predictive maintenance. Generally, a typical failure development can be classified into several stages from failure initiation to the final functional failure. Not all stages can cause the machinery breakdown and require overhaul maintenance. Based on the diagnosis result of failure modes and severity, predictive techniques can help determine how fast the degradation is expected to progress from its current state to functional failure and offer a trade-off maintenance strategy. Figure 7 shows the relationship among the cost, time to failure and the reliability of machines. As discussed in (Larry 1995), when time to failure equals zero, the system will go into breakdown status. The reliability of the system decreases as the time to failure of the system approaches to zero. The performance cost of system increases while the maintenance cost decreases. Thus, the total cost, as the sum of the performance cost and the maintenance cost, decreases firstly, and then increases. Predictive maintenance with the capability of precisely predicting the time to failure and reliability of the system can provide useful information for the decision of an economical maintenance schedule.

Mobile agents are created during the system operation and are able to move to different cloud sensing and computing nodes over the network. Different types of mobile agent could be created and dispatched as needed. For example, the users could dispatch mobile alert agents to the distributed manufacturing processes for monitoring daily operations. The engineering team with certain expertise could send the mobile agents with data analysis algorithms which will roam over the network to perform complex tasks, such as damage diagnosis, prognosis, and maintenance planning.

Thus, the mobile agent and developed cloud sensing and computing node are two essential parts in the presented cloud predictive maintenance framework. It integrates and connects the manufacturing equipment (things) into a service oriented network, and it is also developed using embedded system technology to enable the wide deployment in the distributed environment for timely status retrieval of manu-

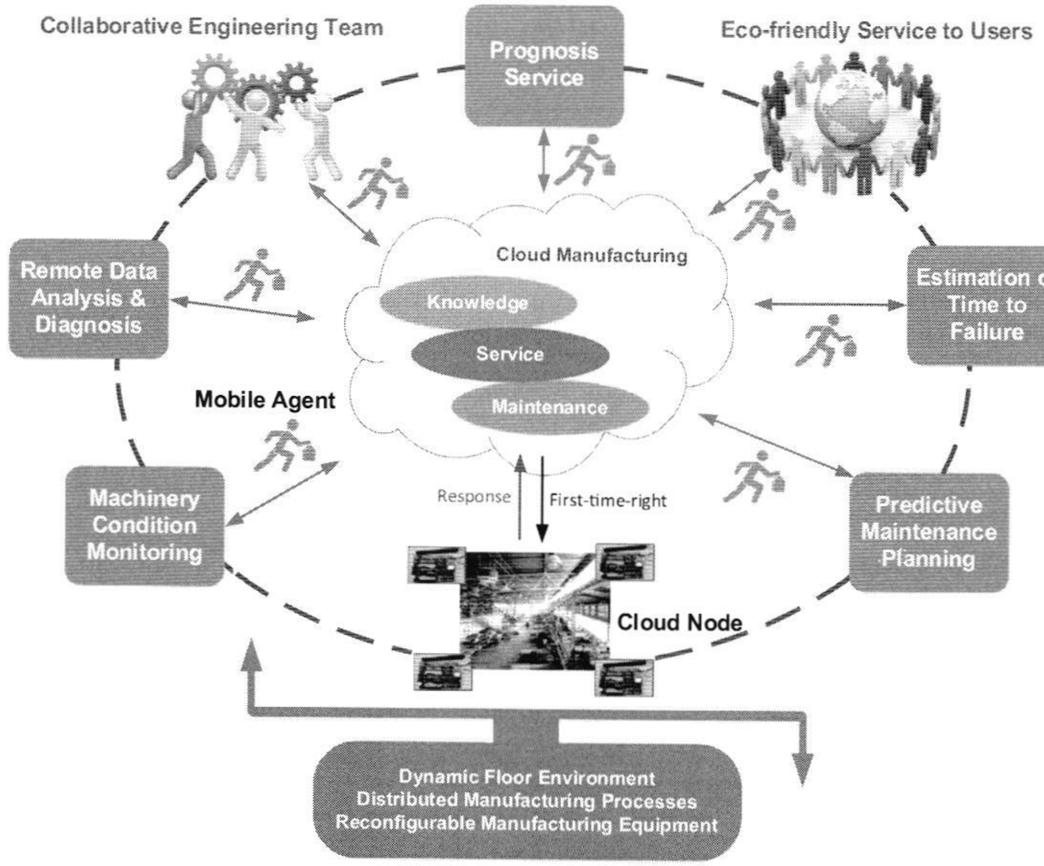

Fig. 6 The proposed paradigm of cloud predictive maintenance based on mobile agent

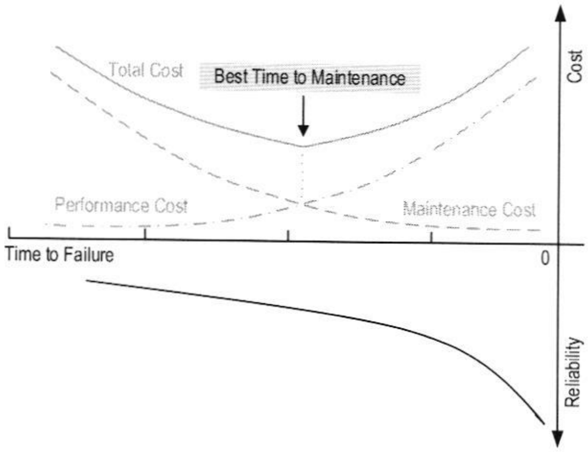

Fig. 7 The relationship among time to failure, reliability and cost

facturing resources. Such internet of things' (IoT) (Distefano et al. 2014) approach makes the manufacturing resources and health status available and accessible over the network. On the other hand, an open and non-proprietary communication standard based on the Extensible Markup Language (XML) is implemented in the mobile agent technology to support machine-to-machine communications and possible interoperability. Moreover, the knowledge-based intelligent computational algorithms can be coded in the mobile agent, and it enables users to search and share data easily by allowing the data from different sources to be processed.

Such mobile agent based cloud predictive maintenance paradigm possesses the following characteristics:

1. *Service-oriented* All functions of the predictive maintenance system are delivered as cloud-based services accessible via network using mobile agents. No expensive software packages are needed for local installation and maintenance.
2. *Accessibility and promotion of robustness* As an integrated solution for modular and configurable services, cloud-based platform can increase robustness and adaptability of existing manufacturing processes. Alternative pay-as-you-go maintenance services and other varying options can be picked from the cloud when necessary or applicable, and then be integrated into the mobile agent platform.

3. *Resource-aware* It allows monitoring and predicting machine utilization and conditions, locally or remotely, by significantly reducing the amount of data transmission; so that decision making for maintenance becomes resource-aware and well informed.
4. *Collaborative and distributive* As an open platform utilizing XML language, it enables information sharing among different applications/machines at different locations, seamlessly and collaboratively.

Experimental study

To evaluate the effectiveness of presented mobile agent based paradigm, six induction motors with different failure modes in a motor tested system are used to mimic the distributed manufacturing processes. The system is driven by 1-hp induction motors, with the speeds varied from 0 to 6000 rpm. Current probes (Fluke *i*200*s*) are clamped on one of the three-phase wires to the motors, measuring the current signals. The shaft rotation speeds are controlled by speed controllers. Static loads are applied through load discs, and variable loads are applied by magnetic brake systems through bevel gearboxes and belt drives. The configuration of the motor tested system via network is illustrated in Fig. 8.

In this study, six motors with identical models but different incipient defects have been investigated as described in Table 2. They are power supplied under the same current supply frequency of 50 Hz and same loading conditions. Take the motor with broken rotor bar defect as an example, the raw current measurements, the envelope of current and their spectrum analysis are illustrated in Fig. 9. Figure 9a shows the raw motor current signal and its envelope. From the spectrum analysis of motor current as shown in Fig. 9b, it is seen that the motor supply frequency contains a major portion of signal energies, and dominates in the motor current. The spectra of motor current signals from a healthy motor and that from a motor with a broken bar are difficult to differentiate. Therefore it is difficult to identify the sideband harmonic signal which is the indicator of broken rotor bar. In comparison, the spectral comparison in Fig. 9c uses the current envelopes between the healthy motor and motor with broken bar. It has clearly shown the difference in the energy concentration associated with the broken rotor bar-related frequencies. Thus current envelope manifests defect related features, and is selected as the signal of interest in this study.

To analyze the failure modes of different motors, a general analytical framework of predictive maintenance is illustrated

Table 2 Test conditions of six induction motors

Index	Fault condition	Fault description
M1	Normal motor	Healthy, no defect
M2	Broken bar	Three broken rotor bars
M3	Bowed rotor	Rotor bent in the center 0.01″
M4	Unbalanced rotor	Unbalance created by adding three washers on the rotor
M5	Stator winding defect	Three turns shorted in stator winding
M6	Defective bearing	Inner race defect bearing in the shaft end

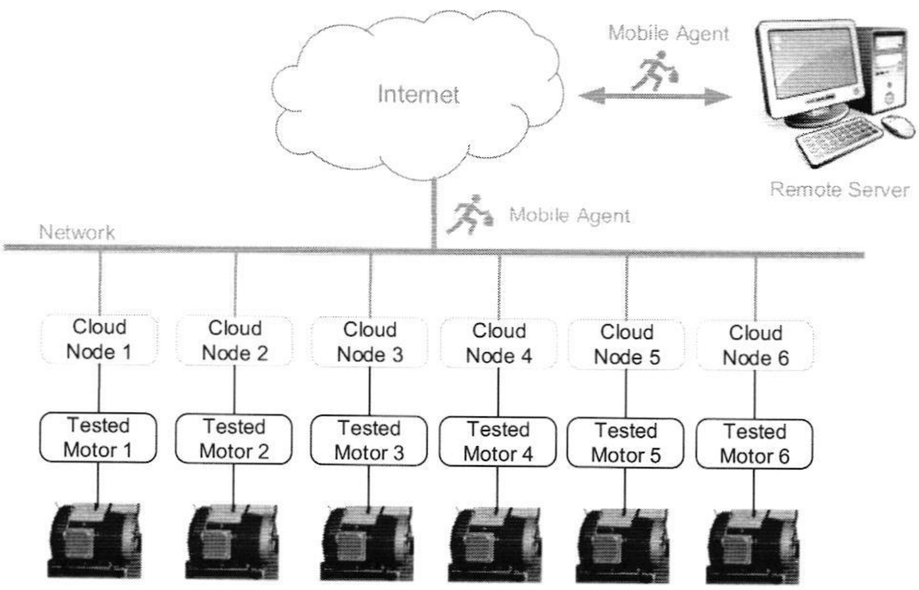

Fig. 8 The configuration of a motor tested system

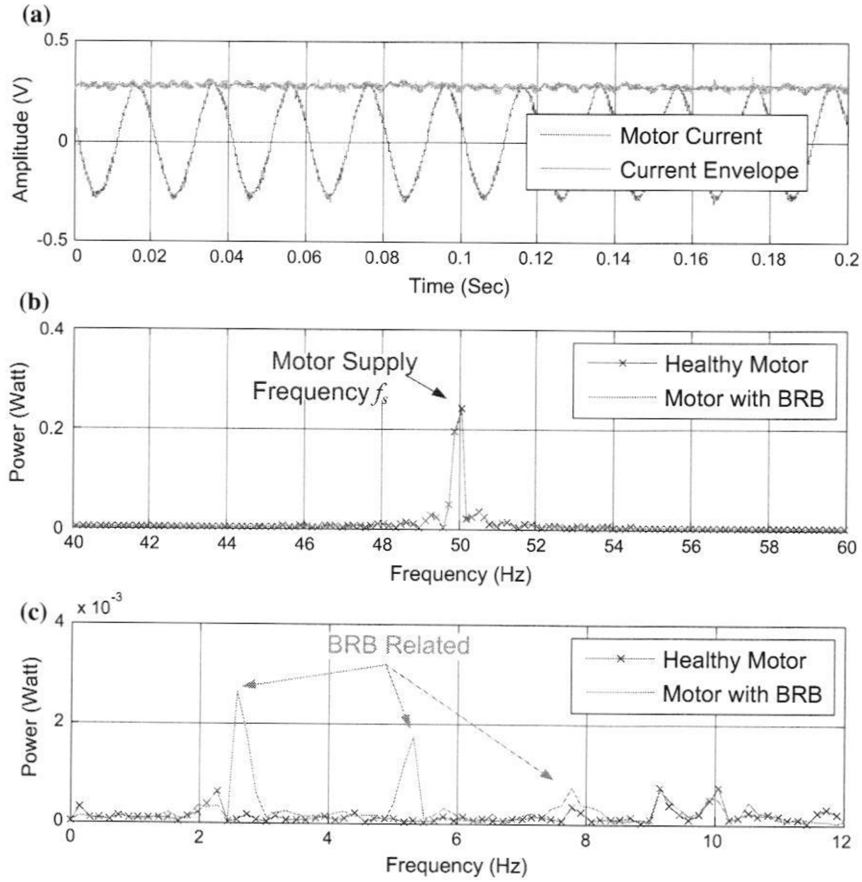

Fig. 9 Comparison of motor current signals and current envelopes between healthy motor and motor with broken rotor bars. **a** Time series signals of motor current and current envelope of a motor with broken rotor bars, **b** comparison of spectrum of motor current signals between a healthy motor and a motor with broken rotor bars, **c** current envelope spectrum comparison between healthy motor and motor with broken rotor bars

in Fig. 10. The aim of predictive maintenance is to provide decision support for maintenance scheduling by diagnosing the defects and predicting the remaining service life of motors. Based on the condition monitoring data measured from certain sensors, various data analysis and signal processing techniques can be involved to extract different features, such as statistical analysis, frequency analysis, model analysis, etc. In the conventional approach, all the raw sensing measurements are collected and transmitted to the central server, where the analysis is performed. Such approach brings huge traffic of data transmission especially in Big Data Era. In contrast, the presented mobile agent based approach codes the feature extraction algorithms in the mobile agents, and sends them to the cloud nodes for locally processing raw data, then brings the extracted features to the central server. Such new approach could highly reduce the amount of data transmission, thus improving the system efficiency.

Different feature extraction algorithms can be applied according to the signal property. In this study, five statistic features, three frequency features, and the coefficients

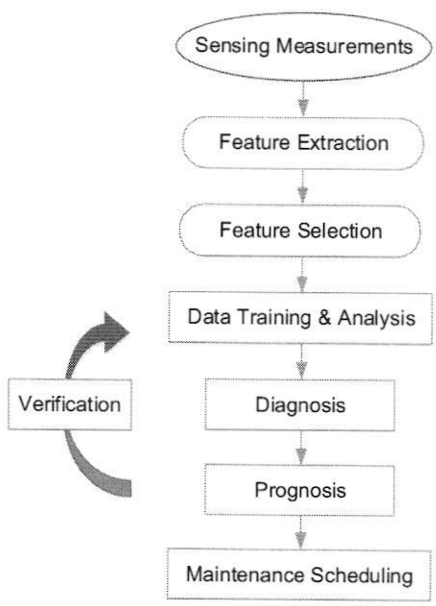

Fig. 10 Analytical framework for predictive maintenance

Table 3 List of extracted features

Domain	Features	Expression
Statistical	RMS	$x_{RMS} = \sqrt{\frac{1}{n}\left(x_1^2 + x_2^2 + \cdots + x_n^2\right)}$
	Skewness	$x_{SKEW} = \frac{1}{n}\sum_n \left(\frac{x_i-\mu}{\sigma}\right)^3$
	Kurtosis	$x_{KURT} = \frac{1}{n}\sum_n \left(\frac{x_i-\mu}{\sigma}\right)^4$
	Entropy	$x_{ENTR} = -\sum_{i=1}^{n} P(x_i)\log P(x_i)$
	Crest factor	$C_F = \max\|x_i\|/x_{RMS}$
Frequency	Air-gap eccentricity	Energy at $f_{ECE} = f_S\left[1 \pm k(1-s)/p\right]$
	Broken bar	Energy at $f_{BRB} = (1 \pm 2ks)f_S$
	Bearing	Energy at $f_{BNG} = f_S \pm kf_{defect}$
Model	AR coefficients	$x_t = \sum_{i=1}^{N} a_i x_{t-i} + \xi_t$

Agent head

```
<MOBILEC_MESSAGE>
 <MESSAGE message="MOBILE_AGENT">
  <MOBILE_AGENT>
   <AGENT_DATA>
    <NAME>Feature_Extraction_Agent</NAME>
     <OWNER>EMS</OWNER>
     <HOME>192.168.1.99:5050</HOME>
     <TASKS task="1" num="0">
<TASK num="0" return="features" complete="0" server="HOST_NAME_HERE:5051"
code_id="1" />
  <AGENT_CODE id="1">
  <![CDATA[
#include <sys/socket.h>
#include <sys/types.h>
#include <math.h>
#include <stdio.h>

double features[23];
int main() {
      printf("Mobile Agent #1 arrived at HOST_NAME_HERE\n");
      feature_sta[5] = statistic(); //pseudo code for feature extraction
      feature_sre[3] = frequency();
      feature_ar[15] = ar_model();
}
      ]]>
     </AGENT_CODE>
    </TASKS>
   </AGENT_DATA>
  </MOBILE_AGENT>
 </MESSAGE>
</MOBILEC_MESSAGE>
```

Indicates a Mobile-C message

The name of the agent

The home of the agent

Include C Library

Encapsulated algorithms in C language

Pseudo code of feature extraction algorithm

Agent Tail

Fig. 11 An example of mobile-C agent for feature extraction

of autoregressive model from motor current envelope are selected for motor fault diagnosis as listed in Table 3.

Five statistic features include the root mean square (RMS), Skewness, Kurtosis, Entropy, and crest factor. The RMS is a measure for the magnitude of a varying quantity. It is also related with the energy of the signal. Skewness is used to characterize the degree of signal asymmetry of the distribution around its mean, and Kurtosis indicates the spikiness of the signal. The crest factor is calculated from the peak value divided by the RMS value of the signal. According to the information theory, entropy provides a quantitative measure of the uncertainties associated with the signals. For the features in frequency domain, the energies at defect characteristic frequencies including broken bar (f_{BRB}), air-gap eccentricity (f_{ECE}), and defective bearing (f_{BNG}) are selected. Autoregressive model coefficients of motor current's envelope are also selected as the representative features (Wang et al. 2012).

These feature extraction algorithms are coded in the mobile agents, and the mobile agents are dispatched by the central server to the cloud nodes to perform tasks via the network. Figure 11 shows an example of the mobile-C agent with the coded feature extraction algorithms. Each agent is assigned its own identification (ID) number when it is created. The ID number lasts for the entire life of the agent. Agent migration is achieved through message passing. When

VLt Erwähnen, wieso das überhaupt relevant

31.

a mobile agent is dispatched, information related to the agent such as agent ID, tasks to be performed, and agent code for each task, is encapsulated into a mobile agent message. The message is an XML message with encapsulated feature extraction algorithms in C code (Cheng 2006a). The results from each task will be added into the mobile agent message, and then sent back to the central server for motor defect diagnosis.

A total of 600 sets of features vectors corresponding to six different motors are extracted from the raw current envelope signals. Each feature vector consists of 23 features: 5 statistical features, 3 features from the frequency domain, and 15 AR coefficients. These extracted features are then fed into the support vector machine for motor defect diagnosis. Support vector machine (SVM) is a widely used pattern classification technique based on statistical learning theory; more details about its theory and implementation can refer to (Vapnik 1999; Widodo and Yang 2007). Two parameters including the cost parameter and Gaussian kernel parameter have been selected through a fivefold cross validation process to prevent over-fitting. Following a leave-one-out cross validation procedure, the motor defect diagnosis from SVM yields 99.8 % of the classification accuracy.

In the conventional approach, all raw data measurements are collected and sent to a central server for feature extraction and defect diagnosis. As the number of sensing nodes and the amount of sensing data increase, it brings huge traffic over the network. In the presented new paradigm of cloud maintenance, the mobile agent with feature extraction functionality is sent to the distributed cloud node, and then collects the analysis results to a remote server for further defect diagnosis. Take the data in the experimental study as an example, the motor current is acquired with the data sampling rate of 2 kHz, and the data with 10 s duration is selected as one data set. Considering one data point of 4 bytes, it needs to transmit 80,000 bytes/set data to the central server in the conventional approach. In the presented new paradigm, since the feature extraction is performed in the cloud node using the mobile agent, only the analysis results of about 92 bytes/set (e.g., 23 × 4 bytes) are sent back to the remote server. Thus the data traffic over the network is significantly reduced due to 99.77 % of data transmission reduction.

Sinn des Agents

Discussions

Problem

In cloud based predictive maintenance, huge amount of sensor measurements may be collected over the life cycle of a manufacturing system. The data size could be up to Terabytes, and even more in Big Data Era. In conventional centralized server paradigm, all sensor measurements are collected and transmitted to the centralized server which brings heavy traffic over the internet. As discussed in (Hadim and Mohamed 2006), sending one single bit of data consumes the same amount of energy as producing the same data by executing thousands of instructions. Thus, it is significant to reduce the amount of data transmission. In the presented mobile agent based paradigm, data processing algorithms are coded in mobile agents which are sent for localized data analysis, and then bring the analysis results back to the remote server. Such an approach not only significantly reduces data transmission, but also enhances the system flexibility since the data processing algorithms are not predesigned in the cloud nodes. In theory, any data processing algorithms and models can be deployed in a mobile agent. However, there are still some practical issues. Predictive maintenance typically involves intensive numerical computation for data analysis and decision making. The size and run-time of a mobile agent highly depend on the availability of numerical functions at cloud nodes. The size of mobile agent could be reduced if all necessary numerical functions are available at a cloud node; otherwise the size of mobile agent may affect its performance since it needs to carry the required numerical function code. For example, the size of mobile agent affects the run-time. The mobile agent code is typically executed interpretively, which makes it slower than executing binary codes (Chen and Liu 2010). If numerical functions are integrated into binary libraries at cloud nodes, the small mobile agent code and high diagnosis speed can be achieved. On the other hand, the large binary libraries also occupy the memory space at the cloud node. The utilization frequencies of numerical functions may need to be considered when implementing binary libraries in order to balance the size of mobile agent, run-time, and memory space. Moreover, the cloud node with an embedded system has limited computing resources comparing with the remote server. Computation intensive algorithms could be executed in the remote server by collecting required information from cloud nodes through mobile agents.

Problem

As discussed above, the benefits of the mobile agent based paradigm rest on the reduction of data transmission and enhancement of system flexibility. To achieve the goal of cloud-based predictive maintenance, a variety of challenges are involved and remain to be investigated.

1. Dynamic resource allocation and autonomous agent control. Computing resources, expert knowledge, computational intelligent algorithms, and machinery monitoring data, etc. are the vital parts for implementing cloud-based predictive maintenance. Dynamic resource allocation and autonomous agent control strategies need to be investigated to provide better services and implement efficient resources allocation and integration across the cloud manufacturing.
2. Heterogeneous data storage and analysis. Current predictive maintenance algorithms (Jardine et al. 2006; Peng et al. 2010; Wang et al. 2013) are usually developed

for structured sensing data. As the monitoring objects of interest increase in the dynamic manufacturing environment, a variety of multi-modality heterogeneous sensing measurements are collected. How to store and analyze these data with efficient predictive maintenance algorithms is a challenge.

3. Communication security. It has long been a challenging issue in cloud computing on guaranteeing user privacy and service security (Ahmad et al. 2014; Khan et al. 2013). To provide a secure cloud maintenance service, data security, network security, data locality, data integrity, authentication, and authorization need to be considered and addressed. In particular, intrusion detection in a mobile agent framework is an important issue. When the mobile agents travel over the network, the security of coded predictive maintenance algorithms and the analysis results must be properly managed.

Conclusions

This paper firstly presents a mobile agent based approach for predictive maintenance in cloud manufacturing. As an emerging technique, mobile agent based approach enables a new paradigm for predictive maintenance as remote services instead of conventional centralized approach, and provides distributed maintenance services across the manufacturing enterprises. In addition, mobile agent can flexibly deploy different services (e.g., signal processing algorithms) to adapt the changes of different operations and tasks in a dynamic manufacturing environment. As demonstrated in the experimental study, mobile agents distribute signal processing algorithms (e.g., feature extraction) to the cloud nodes instead of transmitting raw sensing measurements to the central server, and they can significantly reduce the traffic load over network, especially in Big Data Era.

Acknowledgements This research acknowledges the financial support provided by Science Foundation of China University of Petroleum, Beijing (No. 2462014YJRC039) and National Science foundation of China (No. 51204196). Support on design of cloud sensing and computing node in Michigan Technological University is appreciated. The valuable comments from anonymous reviewers are greatly acknowledged to help improve the paper's quality.

References

Ahmad, A., Maynard, S. B., & Park, S. (2014). Information security strategies: Towards an organizational multi-strategy perspective. *Journal of Intelligent Manufacturing*, *25*, 357–370.

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., et al. (1999). *LAPACK users' guide*. Philadelphia, PA: Society for Industrial and Applied Mathematics. doi:10.1137/1.9780898719604.

Arab, A., Ismail, N., & Lee, L. S. (2013). Maintenance scheduling incorporating dynamics of production system and real-time information from workstations. *Journal of Intelligent Manufacturing*, *24*, 695–705.

Archimede, B., Letouzey, A., Memon, M. A., & Xu, J. (2014). Towards a distributed multi-agent framework for shared resources scheduling. *Journal of Intelligent Manufacturing*, *25*, 1077–1087.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., et al. (2010). A view of cloud computing. *Communications of the ACM*, *53*(4), 50–58.

Bandyopadhyay, S., & Bhattacharya, R. (2015). Finding optimum neighbor for routing based on multi-criteria, multi-agent and fuzzy approach. *Journal of Intelligent Manufacturing*, *26*, 25–42.

Baumann, J., Hohl, F., Rothermel, K., Strasser, M., & Theilmann, W. (2002). MOLE: A mobile agent system. *Software-Practice and Experience*, *32*(6), 575–603.

Bellifemine, F., Caire, G., Poggi, A., & Rimassa, G. (2008). JADE: A software framework for developing multi-agent applications, lessons learned. *Information and Software Technology*, *50*(1–2), 10–21.

Bradshaw, J. M. (1997). *An introduction to software agents*. Cambridge: MIT Press.

Chen, B., & Wang, J. (2008). Design of a multi-modal and high computation power wireless sensor node for structural health monitoring. In *Proceedings of IEEE/ASME international conference on mechatronic and embedded systems and application, Beijing, China* (pp. 420–425).

Chen, B., & Liu, W. (2010). Mobile agent computing paradigm for building a flexible structural health monitoring sensor network. *Computer-Aided Civil and Infrastructure Engineering*, *25*, 504–516.

Chen, T., & Wang, Y. C. (2014). An agent-based fuzzy collaborative intelligence approach for precise and accurate semiconductor yield forecasting. *IEEE Transactions on Fuzzy Systems*, *22*(1), 201–211.

Cheng H. H. (2006a). Mobile-C: A multi-agent platform for mobile C/C++ agents. http://www.mobilec.org

Cheng, H. H. (2006b). Ch: A C/C++ interpreter for script computing. *C/C++ User's Journal*, *24*(1), 6–12.

Chou, Y. C., Ko, D., & Cheng, H. H. (2009). Mobile agent-based computational steering for distributed applications. *Concurrency and Computation: Practice and Experience*, *21*(18), 2377–2399.

Chouikhi, H., Khatab, A., & Rezg, N. (2014). A condition-based maintenance policy for a production system under excessive environmental degradation. *Journal of Intelligent Manufacturing*, *25*, 727–737.

Cucurull, J., Marti, R., Navarro-Arribas, G., Robles, S., Overeinder, B., & Borrell, J. (2009). Agent mobility architecture based on IEEE-FIPA standards. *Computer Communications*, *32*(4), 712–729.

Distefano, S., Merlino, G., & Puliafito, A. (2014). A utility paradigm for IoT: The sensing cloud. *Pervasive and Mobile Computing*. doi:10.1016/j.pmcj.2014.09.006.

Gray, R. S., Cybenko, G., Kotz, D., Peterson, R. A., & Rus, D. (2002). D'Agents: Applications and performance of a mobile-agent system. *Software-Practice and Experience*, *32*(6), 543–573.

Hadim, S., & Mohamed, N. (2006). Middleware for wireless sensor networks: A survey. In *Proceedings of first international conference on communication system software and middleware, New Delhi, India* (pp. 1–7). doi:10.1109/COMSWA.2006.1665174.

Hsieh, F. S., & Lin, J. B. (2014). Context-aware workflow management for virtual enterprises based on coordination of agents. *Journal of Intelligent Manufacturing*, *25*, 393–412.

Jardine, A. K., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, *20*(7), 1483–1510.

Johnansen, D., Lauvset, K. J., van Renesse, R., Schneider, F. B., Sudmann, N. P., & Jacobsen, K. (2002). A TACOMA retrospective. *Software-Practice and Experience*, *32*(6), 605–619.

Khan, A. N., Kiah, M. L. M., Khan, S. U., & Madani, S. A. (2013). Towards secure mobile cloud computing: A survey. *Future Generation Computer Systems*, *29*, 1278–1299.

Lange, D. B., & Oshima, M. (1998). *Programming and deploying java mobile agents with aglets*. Reading, MA: Addison-Wesley.

Larry, T. (1995). *Machinery oil analysis: Methods, automation and benefits* (pp. 1–383). Park Ridge: Society of Tribologists and Lubrication Engineers.

Monostori, L., Vancza, J., & Kumara, S. R. T. (2006). Agent-based systems for manufacturing. *CIRP Annals-Manufacturing Systems*, *55*(2), 697–720.

Peine, H. (2002). Application and programming experience with the Ara mobile agent system. *Software-Practice and Experience*, *32*(6), 515–541.

Peng, Y., Dong, M., & Zuo, M. J. (2010). Current status of machine prognostics in condition-based maintenance: A review. *International Journal of Advanced Manufacturing Technology*, *50*, 297–313.

Press, W. H., Flannery, B. P., Teukolsky, S. A., & Vetterling, W. T. (1992). *Numerical recipes in C: The art of scientific computing*. Cambridge: Cambridge University Press.

Singh, A., & Malhotra, M. (2012). Analysis for exploring scope of mobile agents in cloud computing. *International Journal of Advancements in Technology*, *3*(3), 172–183.

Teti, R., Jemielniak, K., O'Donnell, G., & Dornfeld, D. (2010). Advanced monitoring of machining operations. *CIRP Annals-Manufacturing Technology*, *59*(2), 717–739.

Tripathi, A., Ahmed, T., Pathak, S., & Carney, M. (2002). Paradigms for mobile agent based active monitoring of network systems. In *Proceedings of 2002 IEEE/IFIP network operations and management symposium* (pp. 65–78).

Vapnik, V. N. (1999). *The nature of statistical learning theory*. New York: Springer.

Venters, W., & Whitley, E. A. (2012). A critical review of cloud computing: Researching desires and realities. *Journal of Information Technology*, *27*, 179–197.

Wang, L., Gao R. X., & Ragai I. (2014). An integrated cyber-physical system for cloud manufacturing. In *Proceedings of the ASME international manufacturing science and engineering conference, MSEC2014-4171* (pp. 1–8).

Wang, J., Wang, P., & Gao R. X. (2013). Tool life prediction for sustainable manufacturing. In *Proceedings of 11th global conference on sustainable manufacturing, Berlin, Germany* (pp. 230–234).

Wang, J., Liu, S., Gao, R. X., & Yan, R. (2012). Current envelope analysis for defect identification and diagnosis in induction motors. *Journal of Manufacturing Systems*, *31*(4), 380–387.

Wang, L. (2013). Machine availability monitoring and machining process planning towards cloud manufacturing. *CIRP Journal of Manufacturing Science and Technology*, *6*, 263–273.

Wang, S., Liu, Z., Sun, Q., Zou, H., & Yang, F. (2014). Towards an accurate evaluation of quality of cloud service in service-oriented cloud computing. *Journal of Intelligent Manufacturing*, *25*, 283–291.

Widodo, A., & Yang, B. S. (2007). Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, *21*, 2560–2574.

Wong, D., Paciorek, N., Walsh, T., DiCelie, J., Young, M., & Peet, B. (1997). Concordia: An infrastructure for collaborating mobile agents. In *Proceedings of the first international workshop on mobile agents (MA'97), Lecture Notes in Computer Science* (vol. 1219, pp. 86–97). Berlin: Springer.

Wu, D., Greer, M. J., Rosen, D. W., & Schaefer, D. (2013). Cloud manufacturing: Strategic vision and state-of-the-art. *Journal of Manufacturing Systems*, *32*, 564–579.

Xu, X. (2012). From cloud computing to cloud manufacturing. *Robotics and Computer-Integrated Manufacturing*, *28*, 75–86.

Yang, Y., Gao, R. X., Fan, Z., Wang J., & Wang L., (2014). Cloud-based prognosis: Perspective and challenge. In *Proceedings of the ASME international manufacturing science and engineering conference, Detroit, Michigan, USA, MSEC2014-4155* (pp. 1–6).

Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: State-of-the-art and research challenges. *Journal of Internet Services and Applications*, *1*(1), 7–18.

Zhang, Z., Wang, Y., & Wang, K. (2013). Fault diagnosis and prognosis using wavelet packet decomposition, Fourier transform and artificial neural network. *Journal of Intelligent Manufacturing*, *24*, 1213–1227.