Research Article

The Fusion Model of Multidomain Context Information for the Internet of Things

Bing Jia,¹ Shuai Liu,¹ Yushuai Guan,¹ Wuyungerile Li,¹ and Weiwu Ren²

¹*College of Computer Science, Inner Mongolia University, Hohhot, China*

²*School of Computer Science and Technology, Changchun University of Science and Technology, Changchun, China*

Correspondence should be addressed to Shuai Liu; cs.liushuai@imu.edu.cn

Received 20 August 2017; Accepted 11 October 2017; Published 13 November 2017

Academic Editor: Yin Zhang

Copyright © 2017 Bing Jia et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Internet of Things aims to provide the user with deep adaptive intelligence services according to the user's personalized characteristics. Most of the characteristics are presented in the form of high-level context. But it often lacks methods to obtain high-level context information directly in the Internet of Things. In this paper, so as to achieve the corresponding high-level context information using the specific low-level multidomain context directly obtained by different sensors in the Internet of Things, we present a machine learning method to construct a context fusion model based on the feature selection algorithm and the multiclassification algorithm. First, we propose a wrapper feature selection method based on the genetic algorithm to obtain a simpler and more important subset of the context features from the low-level multidomain context, by defining a suitable fitness function and a convergence condition. Then, we use the decision tree algorithm which is a multiclassification algorithm, based on the rules obtained by training the subset of context features, to determine which high-level context the record set of the low-level context information belongs to. Experiments confirm that the model can be used to achieve higher classification accuracy without more significant time consumption.

1. Introduction

The Internet of Things technology is a network expanded to Internet-enabled objects, whose main function is to connect these objects [1]. It not only greatly improves the convenience of networks but also meets needs that people could not imagine before [2, 3]. The Internet of Things is full of various kinds of information about communication, sensing, and computing information to provide the user with more intelligence services [4]. The context data which is produced by the process of providing intelligence services by sensors and intelligent devices in the Internet of Things is massive and valuable, because the context data can be used to affect the human's service experience in many different ways. The "context" is not a new concept, but up till now, no unified definition of "context" exists either theoretically or practically. This is because different researchers put forward different definitions based on different backgrounds, different understandings, and different perspectives [5]. Schilit and Theimer

proposed the concept of "context" [6] first, and they thought that the context information includes the user's position, the user's identity, the physical objects around the user, and the interaction state of the devices used by the user. Then, many researchers [5–9] proposed the definition of context information based on their own research field and perspectives. On the whole, these definitions were defined based on the traditional viewpoint of "user-center," which mainly consists of three basic essential elements: human, machine, and environment. However, for the Internet of Things, the context information should be coordinated between two humans, a human and an object, and two objects. We adopt the definition in reference documentation [10], which regards context information as the interaction information between human, object, machine, and environment in the Internet of Things. It contains both the preset static information and the dynamic information caused by the interaction. In order to adapt to the user's personalized needs, the provided service should have significant and personalized characteristics to

adapt to the user's context feature [11]. Usually, during the lifetime of the service in the Internet of Things, when the user changes the low-level multidomain context, such as location, temperature, or illumination, the high-level context for the user's personalized characteristics may always be changing [12]. So, it is vital to get the high-level context feature by fusing the low-level context information of the user's multidomain environment timely [13].

Context fusion is a process to obtain the high-level context by dealing with the multidomain low-level context (which consists of monitoring, sensing data, and so on) based on some methods and prior knowledge. Nowadays, most researches have adopted the method of rule reasoning. For example, the middleware of Context Toolkit [14] supported context reuse and customization by the abstract representation. Through encoding program for the logical rules, Context Toolkit achieved conflict detection and reasoning by OOPS (Organized Option Pruning System). Gaia [15] adopted CORBA (Common Object Request Broker Architecture), which is a distributed component architecture based on the thought of operating system, to achieve efficient reasoning by using a first-order predicate logic model. CORBA also introduced the idea of probability and fuzzy logic to deal with the uncertainty. The project of PACE (Pervasive, Autonomic Context-aware Environments) introduced three-valued logic to handle some uncertainty information based on the graphical context modeling language CML (Context Modeling Language) [16]. The SOCAM (Service-Oriented Context-Aware Middleware) [17] based on OSGI (Open Service Gateway Initiative) and the context application middleware CoBrA (Context Broker Architecture) based on agent [18] have both used OWL ontology to achieve reasoning. The expression of the rule-based reasoning is direct, unified, and accurate. This method is more suitable for solving small-scale datasets, but it is difficult for it to deal with complex systems and large-scale datasets, because the relationships between the observed symptoms and the corresponding diagnoses in large-scale systems are more complex, so it is difficult to sum up the effective rules in view of the experience of experts. This paper has introduced a method of machine learning to construct a context fusion model in order to realize the fusion processing of large-scale and multidomain context information.

2. The Classification of the Context

There is no unified standard for the classification of the context. Different researchers proposed different partition methods based on their own research backgrounds, applications, and research requirements. Dey et al. proposed that the context information was location, identity, activity, and time. Context was defined as user and role, process and mission, position, time, and equipment by Kaltz et al. [19], in order to make an extensive range of the mobile and network context [20]. Schmidt et al. proposed that the context information was the human factor and the physical environment in [21]. Luo et al. [22] made a more detailed distinction about source, purpose, and varying frequency of the context information.

We analyzed the context information which may be produced in the new networking environment and the processes of service registration, service discovery, service selection, and service composition based on the needs of the user under the Internet of Things and defined the context information as the environment context information, the device context information, the user context information, and the calculation context information. Details are as follows.

2.1. The Environment Context Information. It is used to describe the context of service environment in the Internet of Things [23], including the specific environment, the scene environment, and the temporal environment. The specific environment refers to the measurable environmental status (such as temperature, humidity, and sound and light noisiness). The scene environment refers to a relatively stable environment (such as conference room or a café). The temporal environment includes some temporal objects (such as time point and time period) and the relations among temporal objects [24].

2.2. The Device Context Information. It mainly refers to the ability and outline of a device in this paper. The device is the carrier of the interaction between the service and the user, including the static device context and the dynamic device context. The static device context includes the device type context and the display performance. The dynamic device context includes the signal intensity, the moving speed, and the electricity of the device.

2.3. The User Context Information. It mainly refers to the ability and outline of the user in this paper. The user refers to the entity which can initiate the service demand in the Internet of Things (people, an ordinary object, etc.). It includes the static user context, the dynamic user context, and the historical user context. The static user context includes the user's identification, identity, and preferences. The dynamic user context includes the user's position, posture, and permission. The position includes the geometric position (specific latitude, longitude, and altitude obtained through GPS, etc.) and the relative position ("end of corridor," "the east side of the classroom," etc.).

2.4. The Calculation Context Information. It mainly refers to the feature associated with the calculation property, including the static calculation context and the dynamic calculation context. The static calculation context includes the computing capacity (storage capacity, processor capacity, etc.) and the network capacity (network type, network bandwidth, communication cost, etc.). The dynamic calculation context includes the CPU utilization rate, the memory utilization rate, and the communicable neighbor nodes.

3. Context Fusion Model

3.1. Structure of the Context Fusion Model. Some of the various contexts mentioned above can be directly and originally

FIGURE 1: The structure context fusion model.

obtained (e.g., light, remaining battery), which are called the low-level contexts.

The original acquisition is the process of obtaining various types of context information (or data) directly from a variety of context information sources (such as sensors, RFID readers, and cameras). The others need to integrate the low-level context information which was obtained from the information source to obtain the high-level context information (scenes, relative position, etc.). For example, a scene which the user is faced with cannot be obtained directly, and it must be a comprehensive analysis of the low-level context, such as the temperature, the humidity, the speed, the position, and the direction, to infer which scene the user faced (in the corridor, upstairs, etc.). The context information of the perception is often complex and has a multidomain for the Internet of Things, and it is difficult to sum up the effective rules based on the experience of the experts. So, it is not suitable to use the rule-based method for reasoning. This paper has introduced the machine learning method and given the context fusion model based on the genetic algorithm and the decision tree. The model structure is shown in Figure 1. First, the feature selection method is adopted to deal with the input dimension reduction. Then, it can determine the kind of high-level context by fusing the low-level context based on the classification algorithm. This method can be trained to obtain inference rules through a large number of samples, and it does not require human intervention. Although the inference result is not very accurate, it is easy to implement and apply in processing large-scale information. It will be more feasible if the requirements of identification accuracy are not very strict.

The model contains two parts: feature selection and classification. The purpose of the feature selection is to decrease the dimension of the training samples and delete some features which are unrelated or weakly related to the

task, in order to obtain the simple but important feature subsets [25]. The sample can be trained to get classification rules on the basis of the feature subsets. Then, according to the classification rules, some low-level contexts can be classified as a kind of high-level context. In this model, dozens of dimensions (recorded as “ m ”) of the low-level context information constitute the original input space, and the high-level context corresponding to the low-level context information (recorded as “ t ”) constitutes the output space. We can obtain n ($n < m$)-dimensional compact feature subset by the feature selection algorithm (filter, wrapper, or mixed mode) for the original input space. Then, we can reduce the dimensions of the test data based on the feature subset. Finally, the classification and recognition task of each sample is completed by the multiclassification algorithm (Bayes, decision tree, or SVM).

Bayes classification follows Bayes theorem. Bayes theorem gives a method to calculate the posterior probability. Bayes classification provides a method which can combine practical learning algorithms and prior knowledge and observed data. It provides a beneficial perspective for understanding and evaluating many learning algorithms. The naive Bayes classification is the most commonly used method in Bayes classification. As the name suggests, this classifier uses the naive Bayes theorem to get the classification for a given variable value. The naive Bayes classifier is a very simple classification algorithm based on probability models with independence assumptions between predictors. The independence assumptions do not often have an impact on reality. Therefore, they are considered as naive. A naive Bayes model is very useful for large datasets, which is easy to build and with no complicated iterative parameter. Despite its simplicity, the naive Bayes classifier is widely used, because it usually behaves well and often outperforms more sophisticated classification methods.

Support vector machine (SVM) is a set of supervised learning methods used for classification, regression analysis, and outliers detection, which is derived from the statistical learning theory. It often yields great classification results from complex and noisy data. SVM is mainly used for two categories of classification problems. Although some of the papers mentioned that the K support vector machine combination can be used to solve the K class classification problem, this process requires some caution.

In the classification technique, the decision tree is a powerful classification method. It can be used to determine the characteristics of the training data segmentation, resulting in a good generalization. Decision tree algorithm can naturally deal with binary or multiclassification problems. And the leaf nodes can refer to any of the K classes concerned.

Through a lot of experiments, we find that the decision tree classification algorithm and the wrapper feature selection method based on the genetic algorithm have achieved good classification results in the context information for the Internet of Things. In this paper, the experiment will be given later in Section 4. Next, we will introduce the feature selection method based on the genetic algorithm and the multiclassification method of the decision tree.

3.2. Feature Selection Based on the Genetic Algorithm. Generally, there are three modes of feature selection, namely, the filter mode, the wrapper mode, and the mixed mode. The filter mode uses the properties of the data itself as an evaluation index for feature subset, and the wrapper uses the correct rate of the machine learning algorithm as the evaluation index of the feature subset [26]. In general, the filter feature selection is faster. The process of the selection is not related to the machine learning algorithm, so the feature subset may not be adaptable to the certain machine learning algorithm. This makes the result subset after feature selection not necessarily the optimum one. The wrapper mode is slower than the filter, because it needs to do cross-certification and more complex calculation. The feature subset can be adapted to the classification algorithm, so the selection result

is generally better. The mixed mode needs to do feature selection in two steps, so the computation time is very large while the accuracy rate is not improved significantly. So, the mixed mode is not used commonly. This paper has chosen the wrapper model to do the feature selection. The working principle of the wrapper is that it needs to package data into different feature subsets in accordance with dimensionality and make its selection through the correct classification rate. So, we need to search the whole feature subset space.

Search strategies are generally divided into three types, namely, exhaustive search, heuristic search, and uncertain search [27]. The exhaustive search strategy can search for all possible feature subsets, and it will be able to find the optimal subset of features. But it is difficult to achieve the optimal solution for a large number of features, because the cost of the space and time is large. The heuristic search strategy will search for subset features according to a certain heuristic rule [28]. Its cost is less, but it is liable to fall into a local optimum, and the global optimum cannot be obtained. The uncertainty search strategy is a balance of the above two kinds of search, for example, the genetic algorithm. We use the genetic algorithm in this paper. Figure 2 shows the wrapper mode based on the genetic algorithm in the feature selection process. Because the context classification is a kind of multiclassification, the typical methods of Bayes, decision tree, and SVM are chosen as the evaluation function of feature subset. The decision tree is the main method used in this paper; Bayes and SVM are the methods used for comparative experiments.

In Figure 2, the parameter needs to be initialized first, which is a key of the genetic algorithm. The feature subset is coded as the only parameter in this paper. Then, the model completes the generation of feature subset, the evaluation of feature subset, evaluation stopping, and the verification of the result. An initial subset is randomly generated according to the initial parameters and the original population. The characteristic subsets of each generation are generated according to the relevant parameters calculated by the genetic operator. The fitness function is defined as follows:

$$\text{fit}(\alpha_i) \begin{cases} \text{CR}(\alpha_i) - \frac{\text{CR}_{\min}}{|S|} & \text{if } \text{CR}_{\max} + \text{CR}_{\min} \leq 2 * \text{CR}_{\text{avg}} \\ \text{CR}(\alpha_i) + \text{CR}_{\max} - 2 * \frac{\text{CR}_{\text{avg}}}{|S|} & \text{if } \text{CR}_{\max} + \text{CR}_{\min} > 2 * \text{CR}_{\text{avg}} \end{cases} \quad (1)$$

In (1), α_i is the i th generation individual. $\text{CR}(\alpha_i)$ is the correct classification rate of the subset. S is all subsets in this generation. $|S|$ is the number of subsets. $\text{CR}_{\max}$ is the maximum classification accuracy of the subset in the i th generation. $\text{CR}_{\min}$ is the minimum classification accuracy of the subset in the i th generation. CR_{avg} is the average classification accuracy of the subset in the i th generation.

Genetic operators are the key to implement the optimal search. There are three kinds of operators, namely, selection, crossover, and mutation [29]. The selection operator selects

the individual which can be inherited to the next generation by a certain strategy, based on the evaluation of individual fitness [30]. The crossover operator randomly selects two chromosomes according to a certain crossover probability and exchanges some of its genes in some way to form two new individuals. The mutation operator is used to generate a new individual by randomly changing some bits on the chromosome in a small probability P_m . The crossover operation is the main method of generating new individuals in the process of genetic operation. It determines the global

FIGURE 2: Feature selection process based on the genetic algorithm.

search capability [31] of the genetic algorithm. The mutation operation is an auxiliary method to generate new individuals, and it determines the local search ability of the genetic algorithm. The mutual cooperation between the crossover operator and the mutation operator will complete the global search and local search for the search space. It can make the genetic algorithm complete the search process with a good performance [32]. In this paper, two conditions of the algorithm convergence are designed: one is that the subset has achieved stability and the other is that generation quantity has been over the threshold.

3.3. Decision Tree Classification Algorithm. One of the commonly used methods in data mining is the decision tree learning method. The goal of the decision tree learning method is to create a model that predicts the value of the target variable based on several input variables. Each internal node corresponds to an input variable and there are edges to children for each possible input variable value. Each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf.

The top node of the tree is root. The decision tree can be established and redefined by the training of test samples. The building process of the decision tree is a machine learning process [33].

Currently, there are lots of classic decision tree algorithms, such as the ID3 algorithm, the C4.5 algorithm, and the CART algorithm. The ID3 algorithm can only deal with

discrete data. The C4.5 algorithm has made some improvements to the ID3 algorithm according to the information gain ratio which is used to select the test attributes. Some of these are as follows: dealing with both continuous and discrete attributes, dealing with training data with missing attribute values, dealing with attributes with differing costs, and pruning trees after creation [33]. The CART algorithm cannot efficiently handle large-scale training sample data. Based on the above analysis, in this paper, we have chosen the C4.5 algorithm.

3.4. C4.5 Algorithm

3.4.1. The Calculation of the Information Gain. If the value of the property (recorded as “ a ”) of the sample will divide the sample set (recorded as “ T ”) into m subsets, namely, $T_1, T_2, \dots, T_m$, the formula for calculating the information gain is as follows:

$$\text{gain}(a) = \text{info}(T) - \sum_{i=1}^m \frac{|T_i|}{|T|} \times \text{info}(T_i). \quad (2)$$

In (2), “ $|T|$ ” is the number of samples in dataset “ T ”; “ $|T_i|$ ” is the number of samples in subset “ T_i ”; $\text{info}(T)$ is calculated as follows:

$$\text{info}(T) = - \sum_{j=1}^s \text{freq}(C_j, T) \times \log_2(\text{freq}(C_j, T)), \quad (3)$$

where $\text{freq}(C_j, T)$ is the frequency of the category of the sample data “ C_j ” and “ s ” is the number of categories of the sample “ T .”

3.4.2. The Calculation of the Information Gain Ratio. One has

$$\text{gainratio}(a) = \frac{\text{gain}(a)}{\text{splitinf}(a)}. \quad (4)$$

In (4), $\text{splitinf}(a)$ represents the split information, which is the potential information generated when “ T ” is divided into “ h ” parts; the formula is as follows:

$$\text{splitinf}(a) = - \sum_{i=1}^h \frac{|T_i|}{|T|} \times \log_2 \left(\frac{|T_i|}{|T|} \right). \quad (5)$$

3.5. Building the Decision Tree. The method of building a decision tree is proposed as follows:

- S1: Create the node “ N ” and start building the decision tree from the node.
- S2: If the samples are in the same class, the node becomes a leaf node. Label the node with this class.
- S3: Otherwise, for each property, the data should be dispersed if its data is continuous.
- S4: The information gain ratio is calculated for each attribute, and then the property which has the highest information gain ratio will be selected and labeled.

S5: The consistent value is calculated for the properties of each branch. And then it produces the branch with the same value.

S6: Let “S” be the branch set of the training test set. If “S” is empty, it needs to add a leaf node and be marked by the class.

S7: If “S” is not empty, go to “S4.”

To prevent overfitting between the established trees and the training samples to enhance the speed and accuracy of the subsequent classification, we usually need a pruning strategy. Pruning is a technique in machine learning. It reduces the scale of decision trees by eliminating sections of the tree that offer small power to classify instances. Pruning not only reduces the complexity of the final classifier but also improves the predictive accuracy by the reduction of overfitting. Common methods of calculating the classification error rate and the encoding length of the decision tree are used to prune the decision tree [34]. For each nonleaf node, the pruning method will calculate the expected classification error rate if the node is pruned based on the classification error rate. At the same time, according to the classification error rate of each branch and the weight of each branch, the expected classification error rate will be calculated if the node is not pruned. If the expected error rate gets larger because of the pruning, the pruning will be abandoned, and each branch of the corresponding node will be retained. Otherwise, each branch of the corresponding node will be pruned. In the pruning process, an independent test dataset is needed to be used to evaluate the accuracy of classification for the pruned tree, to retain a pruning decision tree which is the minimum expected error rate after being pruned.

4. Experiments and Results Analysis

In this paper, we use a context information dataset, “Sensor Signal Dataset for Exploring Context Recognition of Mobile Devices” proposed in [35]. There are thirty-two columns about the sensor information [25]. Among them, the first column and the second column show the sequence number of the test scheme and the times of the test repetition; the third column shows the context information of the time; the fourth column to the ninth column show the context information of the device direction; the tenth and the eleventh columns show the context information of the device stability; the twelfth column shows whether the device is in the hand; the thirteenth column to the nineteenth column show the context information of the illumination; the twentieth column to the twenty-third column show the context information of the temperature; the twenty-fourth column to the twenty-sixth column show the context information of the humidity; the twenty-seventh column to twenty-ninth column show the context information of the noise; the thirtieth column to the thirty-second column show the action context information of human behavior. The 10,470 records from the third column to the thirty-second column are chosen in this experiment, to constitute the original context dataset (10470*30). According to the image materials given by the authors [28], we organize

TABLE 1: Eight kinds of context scenes.

Scene context name	Class code
The equipment is on the table	1
The equipment is in the hands of man	2
Walking in the office	3
Walking in the corridor	4
Going downstairs	5
Walking in the coffee shop	6
Walking in the streets	7
Going upstairs	8

TABLE 2: Result of feature selection (E1).

Algorithm	Feature subset
Bayes	3, 5, 7, 9, 10, 11, 12, 13, 24, 27, 28, 30
Decision tree	3, 4, 9, 10, 14, 15, 18, 19, 20, 21, 23, 25, 29
SVM	3, 5, 6, 7, 10, 11, 15, 16, 17, 20, 28, 30, 32

TABLE 3: Real-time comparison (E1).

Algorithm	Training time	Test time
Bayes	9.4 s	0.3 s
Decision tree	10.6 s	0.4 s
SVM	27.3 s	2.1 s

the dataset into eight different kinds of context scenes which are shown in Table 1.

In the first experiment (E1), 2,960 records have been selected randomly as a feature selection dataset. The remaining data has been used in the classification experiment. The Bayes algorithm and the SVM algorithm have been used as comparison algorithms. The former uses a more representative method called naive Bayes. The latter uses the SVM multiclassification algorithm based on the voting mechanism and uses the C-SVM algorithm for each binary classification, whose kernel function is an RBF kernel function. The test results are as follows.

The results of feature subset selection of the three algorithms are shown in Table 2. From the table, the SVM algorithm and the decision tree algorithm select thirteen features, but the Bayes algorithm selects 12 features. We can see that the features in the third and tenth column are more important because these three algorithms all have selected these features.

Table 3 shows the time consumed by the three algorithms when training and testing. It can be seen that Bayes algorithm takes the least time, and the decision tree algorithm takes more time than Bayes. The SVM consumes more training and testing time than Bayes and the decision tree.

A comparison of the classification accuracy of the three algorithms is shown in Figure 3. As can be seen, the highest is the decision tree whose classification accuracy rate is more than 95%. The second is the SVM with nearly 89%. The lowest is the Bayes method with only approximately 57%.

So as to verify the adaptability of the proposed fusion model in the paper for different context data, we have carried

TABLE 4: Result of feature selection (E2).

Algorithm	Feature subset
Bayes	3, 4, 7, 8, 10, 11, 12, 13, 14, 15, 18, 20, 25, 27, 29
Decision tree	2, 3, 5, 7, 10, 11, 12, 15, 17, 18, 19, 22, 25, 26, 30, 31
SVM	2, 3, 24, 28, 30

FIGURE 3: Comparison of the classification accuracy (E1).

out another experiment (E2) using another 980 random records in the datasets. We also used the Bayes algorithm and SVM algorithm as comparison algorithms. The results of feature subset selection of the three algorithms are shown in Table 4. The results of classification accuracy of the three algorithms are shown in Figure 4.

Simulation results show that the multiclassification algorithm based on decision tree can achieve higher classification accuracy, compared to the classical Bayes and the SVM classification algorithm. It is suitable for solving the fusion of large-scale multidomain low-level context information. So, it can obtain the high-level context class represented by the low-level context information more quickly and accurately. In addition, the random forest is an extended version of the decision tree, so if the decision tree was replaced by random forest in the context fusion model, it can also achieve very good results.

5. Conclusion

This paper has proposed a context fusion model based on the machine learning method to achieve the fusion of the multidomain low-level context information under the Internet of Things. First, the dimensions of the original data are reduced by using the wrapper feature selection method based on the genetic algorithm. Then, based on the decision

FIGURE 4: Comparison of the classification accuracy (E2).

tree classification algorithm, it completes the classification and recognition of the low-level context identification to determine which kind of high-level context it belongs to. The experimental results confirm the validity of the proposed model. If the recognition accuracy requirements are not particularly strict, the model will be feasible for large-scale context information. Further research is needed on the optimal selection of some related parameters in the algorithm.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (nos. 41761086, 61640013, and 61502254), the Program of Higher-Level Talents of Inner Mongolia University (no. 135138), and the Inner Mongolia Autonomous Region Science and Technology Innovation Guide Reward Fund Project under Grant no. 20121317.

References

- [1] Q. Liu, Y. Ma, M. Alhussein, Y. Zhang, and L. Peng, "Green data center with IoT sensing and cloud-assisted smart temperature control system," *Computer Networks*, vol. 101, pp. 104–112, 2016.
- [2] C. Zhang, Y. Yang, Z. Du, and C. Ma, "Particle swarm optimization algorithm based on ontology model to support cloud computing applications," *Journal of Ambient Intelligence and Humanized Computing*, vol. 7, no. 5, pp. 633–638, 2016.
- [3] B. Jia, *Research on Semantic-based Service Architecture and Key Algorithms for the Internet of Things*, Jilin University, 2013.

- [4] K. Kotis and A. Katasonov, "Semantic interoperability on the Web of things: The semantic smart gateway framework," in *Proceedings of the 2012 6th International Conference on Complex, Intelligent, and Software Intensive Systems, CISIS 2012*, pp. 630–635, July 2012.
- [5] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "CA4IOT: Context awareness for Internet of Things," in *Proceedings of the 2012 IEEE International Conference on Green Computing and Communications, GreenCom 2012, 2012 IEEE International Conference on Internet of Things, iThings 2012 and 5th IEEE International Conference on Cyber, Physical and Social Computing, CPSCom 2012*, pp. 775–782, fra, November 2012.
- [6] B. N. Schilit and M. M. Theimer, "Disseminating active map information to mobile hosts," *IEEE Network*, vol. 8, no. 5, pp. 22–32, 1994.
- [7] J. L. Encarnação and J. M. Rabaey, *Journal of Mobile Communication*, Springer US, Boston, MA, 1996.
- [8] D. Franklin and J. Flischart, "All gadget and no representation makes jack a dull environment," in *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*, pp. 155–160, 1998.
- [9] G. Chen and D. Kotz, "Survey of context-aware mobile computing research," Dartmouth Computer Science Technical Report, 2002.
- [10] P. Haitao, *Research on Feature Selection Algorithms in Machine Learning*, Shandong University, 2011.
- [11] Y. Zhang, M. Chen, N. Guizani, D. Wu, and V. C. Leung, "SOV-CAN: Safety-Oriented Vehicular Controller Area Network," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 94–99, 2017.
- [12] W. Wei, H. Song, W. Li, P. Shen, and A. Vasilakos, "Gradient-driven parking navigation using a continuous information potential field based on wireless sensor network," *Information Sciences*, vol. 408, pp. 100–114, 2017.
- [13] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [14] G. Pujolle, "An Autonomic-oriented Architecture for the Internet of Things," in *Proceedings of the IEEE John Vincent Atanasoff 2006 International Symposium on Modern Computing (JVA'06)*, pp. 163–168, Sofia, Bulgaria, October 2006.
- [15] G. Wu, S. Talwar, K. Johnsson, N. Himayat, and K. D. Johnson, "M2M: from mobile to embedded internet," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 36–43, 2011.
- [16] N. Koshizuka and K. Sakamura, "Ubiquitous ID: standards for ubiquitous computing and the internet of things," *IEEE Pervasive Computing*, vol. 9, no. 4, pp. 98–101, 2010.
- [17] Y. W. Kim, "Ubiquitous Sensor Network," ETPI. 2007.
- [18] S. Fukunaga, T. Tagawa, K. Fukui, K. Tanimoto, and H. Kanno, "Development of ubiquitous sensor network," *Oki Technical Review*, vol. 71, no. 4, pp. 24–29, 2004.
- [19] J. W. Kaltz, J. Ziegler, and S. Lohmann, "Context-aware web engineering: Modeling and applications," *Revue d'Intelligence Artificielle*, vol. 19, no. 3, pp. 439–458, 2005.
- [20] "Context awareness," http://en.wikipedia.org/wiki/Context_awareness.
- [21] A. Schmidt, M. Beigl, and H.-W. Gellersen, "There is more to context than location," *Computers & Graphics*, vol. 23, no. 6, pp. 893–901, 1999.
- [22] J. Luo, X. Qing, and S. Chen, "Context-based triggered task model in pervasive computing," *Journal of Chinese Mini-Micro Computer Systems*, vol. 25, no. 8, pp. 1542–1545, 2004.
- [23] Z. Wu, X. Tao, and J. Lv, "An ontology based dynamic context model," *Journal of Frontiers of Computer Science and Technology*, vol. 2, no. 4, pp. 356–367, 2008.
- [24] Y. Zhang, M. Qiu, C.-W. Tsai, M. M. Hassan, and A. Alamri, "Health-CPS: Healthcare cyber-physical system assisted by cloud and big data," *IEEE Systems Journal*, vol. PP, no. 99, 2015.
- [25] P. Qian, S. Wang, and H. Yan, "Further feature based fuzzy classification for Context change awareness in pervasive computing," *Application Research of Computers*, vol. 27, no. 5, pp. 1648–1652, 2010.
- [26] F. Alighardashi and M. A. Zare Chahooki, "The Effectiveness of the Fused Weighted Filter Feature Selection Method to Improve Software Fault Prediction," *Journal of Communications Technology, Electronics and Computer Science*, vol. 8, pp. 5–11, 2016.
- [27] L. Hui and Y. Cao, "Study of heuristic search and exhaustive search in search algorithms of the structural learning," in *Proceedings of the 2010 2nd International Conference on MultiMedia and Information Technology, MMIT 2010*, pp. 169–171, April 2010.
- [28] S. Liu, Z. Pan, W. Fu, and X. Cheng, "Fractal generation method based on asymptote family of generalized Mandelbrot set and its application," *Journal of Nonlinear Sciences and Applications*, JNSA, vol. 10, no. 3, pp. 1148–1161, 2017.
- [29] A. H. Beg and M. Z. Islam, "Novel crossover and mutation operation in genetic algorithm for clustering," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation, CEC 2016*, pp. 2114–2121, July 2016.
- [30] S. F. Chenoweth, J. Hunt, and H. D. Rundle, "Analyzing and comparing the geometry of individual fitness surfaces," in *International Conference on Consumer Electronics, 2005. ICCE. 2005 Digest of Technical Papers*, pp. 89–90, 2016.
- [31] H. Liu and S. Wei, "Analysis on genetic operators," *Computer Technology and Development*, vol. 16, no. 10, pp. 80–82, 2006.
- [32] B. Vázquez-Barreiros, M. Mucientes, and M. Lama, "ProDiGen: mining complete, precise and minimal structure process models with a genetic algorithm," *Information Sciences*, vol. 294, pp. 315–333, 2015.
- [33] M. Somvanshi and P. Chavan, "A review of machine learning techniques using decision tree and support vector machine," in *Proceedings of the 2nd International Conference on Computing, Communication, Control and Automation, ICCUBEA 2016*, August 2017.
- [34] L. Lin H, K. Chen, and H. Chiu R, "Predicting customer retention likelihood in the container shipping industry through the decision tree approach," *Journal of Marine Science Technology*, p. 25, 2017.
- [35] J. Mäntyjärvi, J. Himberg, P. Kangas, U. Tuomela, and P. Huuskonen, "Sensor Signal Data Set for Exploring Context Recognition of Mobile Devices," in *Workshop "Benchmarks and a database for context recognition" in conjunction with the 2nd International Conference on Pervasive Computing (PERVASIVE 2004)*, Vienna, Austria, 2004.