

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Adaptive Hybrid Semantic Service Matchmaking Based on Feature Learning Classification

WEI JIANG^{1,2}, HUIQIANG WANG¹, GUANGSHENG FENG¹, SHICHEN ZOU³, and ZIHAN GAO¹

¹College of Computer Science and Technology, Harbin Engineering University, Harbin, 150080, China

²College of Computer Science and Information Engineering, Harbin Normal University, Harbin, 150080, China

³Huawei Nanjing Research Institute, Huawei Technologies Co., Ltd., Nanjing, 210046, China

Corresponding author: Guangsheng Feng (e-mail: ica@hrbeu.edu.cn).

This paragraph of the first footnote will contain support information, including sponsor and financial support acknowledgment. For example, "This work was supported in part by the U.S. Department of Commerce under Grant BS123456."

ABSTRACT With the help of service composition, dynamic building systems have become an effective way to meet the needs of users in large-scale dynamic computing environments, such as cloud computing. Semantic service matching (SSM) uses semantic information to mine the services that meet the needs of users from the vast amount of services and improves the efficiency of the service system and user satisfaction. High quality SSM is the premise and key to guarantee the dependability of the service system. At present, some common types of SSM methods based on logical or non-logical matching have some defects in the face of False Positive and False Negative problems, leading to the recall and precision is not ideal. Combining different types of SSM methods is an effective way to improve this situation, but how to adaptively combine different service matching methods is still a difficult issue. In this paper, we propose an adaptive hybrid semantic service matchmaking method based on feature learning classification. First, we transform the service matching into a two-classification problem based on multidimensional feature vectors. Then, the random forest method is used to adaptively combine different service matching methods to avoid the complexity of traditional threshold setting method. Experimental results show that the proposed method has some advantages in terms of precision and recall.

INDEX TERMS feature learning, random forest, service composition, service matchmaking, semantic service

I. INTRODUCTION

SEMANTIC services have the characteristics of machine understandable and reasoning, which can realize interoperability at the semantic level and enhance the automation and intelligence of a service system. A semantic service system involves service publishing, service matching, service composition, etc. Semantic service matchmaking (SSM) can satisfy the user's needs by finding the suitable services and enhance the efficiency of the service system and improve user satisfaction, which is the premise and key for service system dependability. The common methods of SSM can be divided into two categories: logic-based service matching and non-logic-based service matching. The logic-based service matching method is mainly aimed at logical reasoning and deduction of semantic service description information. For example, SPARQLent [1] uses the SPARQL to describe preconditions, effects and input/output concepts of semantic

services, and then uses the RDF entailment rule to do service matchmaking. Logic-based service matchmaking performs well on precision ratio due to strict logical reasoning, however, the recall ratio is not ideal because of its excessive strictness. Non-logical-based service matching is based primarily on text similarity, XML/RDF graph matching, data mining and other non-logical reasoning methods. Although the method is easy to implement, it has obvious defects in precision.

Since the purpose of a service is satisfying the needs of users, for SSM, it is a subjective feeling from a user whether the candidate service given by the matcher can satisfy the user's needs. Because of the difference between the user's perception and the matching results calculated by the service matcher, there are two issues in semantic service matchmaking, which are False Positive and False Negative.

1) **False Positive.** Based on the logic analysis on con-

cepts of input/output between the service and users' demands, the semantic service matchmaking is a confirmed success. In fact, the provided service is difficult to complete users' demands, which means matchmaking is a failure. One of the reasons for this problem is that the concept definition in ontology cannot accurately capture the semantics of the real world, which is reflected in the difference of ontology granularities. For instance, a user needs to get the price of a hybrid sedan, and the service matchmaker returns a query service for all car prices to the user due to semantic deduction. However, user believes that true semantic distance between "hybrid sedan" and "car" is too large. The service does not meet their needs from user's subjective judgement. This situation was noted in the semantic service retrieval test set in OWLS-TC. Another reason is that many SSM methods match the input/output of services without considering the functional information contained in preconditions and effects of the service, which results in failure of the service matching.

- 2) **False Negative.** False Negative means that the semantic service matcher affirms that a service that matches the user service request does not match in fact. The main reason for this problem is the over strict logical matching restrictions. There are some problems in ontology based on strict logical partitioning. In the real world, the concept of similar semantics may be completely exclusive in the ontology tree. For example, debit cards and credit cards can both make payment. When a user requests to use a debit card to pay, it is often implicit that the user can pay using a credit card as well. However, debit cards and credit cards are divided into two completely mutually exclusive concepts in the ontology tree by the property of overdraft account, where a service which matches the user's service request is considered as mismatched.

The main reason for False Positive and False Negative is the use of a single type of service matching method, which has flaws in either way. In recent years, hybrid methods that combine different types of SSM methods are emerging, such as OWLS-MX3, XSSD, LOG4SWS, etc. But how to combine different service matching methods adaptively is still a challenging problem. Many hybrid service matchmakings use different matching methods to calculate service matching degrees, and then sum all the matching degrees according to the predefined weight values. And the result which exceeds a certain threshold is considered as a successful match. However, the weights and the threshold settings are not described in details. In fact, the process of service matching can be attributed to classifying the results of service matching as matching success and failure with the service matching degree as the feature value. As a result, service matching is transformed into a two-classification problem based on multidimensional feature vectors. In this regard, we propose

an adaptive hybrid semantic service matchmaking method, which includes:

- 1) To address the False Negative problem caused by strict logic, we propose a matching method based on approximate logic. Using concept contraction and abduction to achieve relaxation of the concept ontology constraints, the probably semantic service which satisfies users' demands can be discovered;
- 2) To address the False Positive problem in the process of SSM, we propose a service precondition and effect matching method. Supplemented by a series of SSM methods based on non-logical method, the proposed method can avoid the service match result is difficult to meet the needs of the business user, reduce dependency reduction threat;
- 3) Based on the random forest method, we propose an adaptive hybrid SSM method based on feature learning classification. The SSM algorithm results are used to construct a service matching feature vector. By optimizing and cutting, we use the random forest classification method to determine the matching degree of the service and user request, achieving adaptive hybrid of different matching methods.

II. SSM BASED ON LOGICAL METHOD

SSM based on logical method usually uses Deductive Reasoning on service semantic to determine whether the semantic between services and user needs is equivalent or reasonable logic containment relations in terms of input/output. The principles of process are: (a) All inputs of semantic services should be matched by the input of user service requests. Only in this way can the service request provide enough input information to ensure the service work normally. (b) All outputs of the service request should be matched by the output of the semantic service. Only in this way can the output information of the service be satisfied with the requirements of the user. Usually the semantic service input/output function parameters are given in the representation of concept in an ontology tree, and the preconditions and effects can be described using first-order logic rule description language, such as SWRL and PDDL. These conditions allow semantic services to support logical reasoning, so it is feasible and effective to use logical methods to match semantic services. However, such strict methods can cause the deterioration of false negative problem, resulting in low recall of service matching. Therefore, while we propose SSM based on strict logic, a SSM method based on approximate logic is proposed by introducing the concept of logic contraction and abductive thought [2], which alleviates the problem of False Negative.

A. STRICT LOGIC MATCHING METHOD

Matching based on strict logic is the most common SSM method at present, and most of the methods follow the principle $\forall C \in I_S, \exists C' \in I_R : C \sqsubseteq C' \wedge \forall D' \in O_R \exists D \in O_S : D \sqsubseteq D'$, in which C, C', D, D' are logical concepts in ontology. However, in some cases such a principle can lead

to deterioration of the False Positive problem, such as partial lack of service or request input and output concepts, as shown in Figure 1.

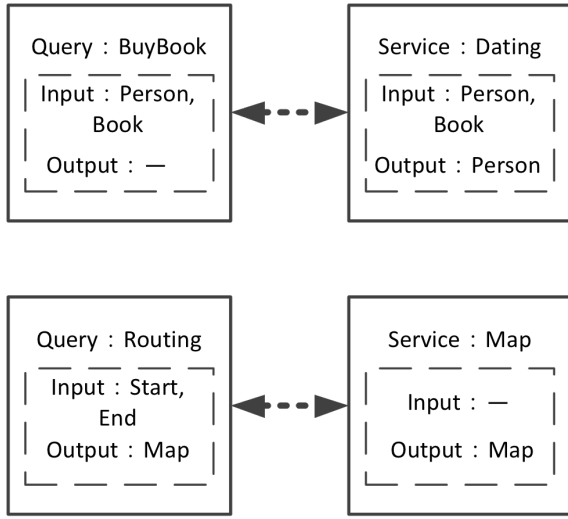


FIGURE 1. False Positive problem in service matching.

There are two groups of customer service requests and semantic services in Figure 1, a group in which a user needs a service to buy books. However, by matching criterion, a dating service with the different output same input and has nothing to do with the needs of business users is given to the user. Similarly, the matcher may also incorrectly match a service that has no input but has the same output as the service request and erroneously matches the user. The main reason for such a situation is that $\forall C \in I_S, \exists C' \in I_R$ put up a surjective relation between the input of semantic service concept and request, therefore, the lack of definitions of service semantics, even key concepts in service input, is wrongly tolerate which also occur on output. To improve this situation, it is necessary to ensure that the mapping between input and output concepts is injective, which guarantees that key concept mapping is not missing.

In fact, the input/output of the semantic service and the user request constitute a bipartite graph, in which the concepts are the nodes, and the weights of the concepts in the ontology tree are the degrees of difference. Thus, the injective mapping between the concepts of the input and output is the maximum matching of the bipartite graph, and the matching points in the maximal allocation must cover all the inputs (outputs) of the semantic service completely. There are many mature algorithms for solving bipartite graph matching, which will not be discussed in details here. Finally, the two nodes of the matching edge in the maximal matching are regarded as a concept pair, which will be added to the concept mapping set A or B according to whether it is input or output.

In addition, since it is not convenient for the matcher to report the matching degree between services and requirements with a single matching criterion, the matching results

are defined as *Exact*, *Plug-in*, *Subsumes*, *Subsumed-by* and *Logical Fail*, according to the semantic service input/output bipartite graph maximum matching mentioned above.

Definition 1. Let S and R be semantic service and user service request respectively, and I_S , I_R , O_S and O_R are collections of semantic concepts of inputs and outputs of semantic service S and user service request R , which are defined on the same ontology tree. Based on the strict logic of semantic service matchmaking, we divide the matching degree of S and R into the following situations:

1) *Exact*

$$(\forall C \in I_S, \exists C' \in I_R : (C, C') \in BCM(I_S, I_R) \wedge C \equiv C') \wedge (\forall D' \in O_R, \exists D \in O_S : (D, D') \in BCM(O_S, O_R) \wedge D \equiv D') \quad (1)$$

2) *Plug-in*

$$(\forall C \in I_S, \exists C' \in I_R : (C, C') \in BCM(I_S, I_R) \wedge C \sqsupseteq C') \wedge (\forall D' \in O_R, \exists D \in O_S : (D, D') \in BCM(O_S, O_R) \wedge D \sqsubseteq_1 D') \quad (2)$$

3) *Subsumes*

$$(\forall C \in I_S, \exists C' \in I_R : (C, C') \in BCM(I_S, I_R) \wedge C \sqsupseteq C') \wedge (\forall D' \in O_R, \exists D \in O_S : (D, D') \in BCM(O_S, O_R) \wedge D \sqsubseteq D') \quad (3)$$

4) *Subsumed-by*

$$(\forall C \in I_S, \exists C' \in I_R : (C, C') \in BCM(I_S, I_R) \wedge C \sqsupseteq C') \wedge (\forall D' \in O_R, \exists D \in O_S : (D, D') \in BCM(O_S, O_R) \wedge D \sqsupseteq_1 D') \quad (4)$$

5) *Logical Fail* When there is no matching relation between S and R , the strict logical match between S and R is considered as fail.

Both Plug-in and Subsumes matching are concepts of semantic service output, which comprise the concept of user service request output. The difference is that the concept of service output in Plug-in is a direct sub node of the user request output concept in an ontology tree, so that the output of the service is closer to the expected output of the user in the true semantic distance. In addition, Subsumed-by matching is achieved by relaxing restrictions on some outputs, because more generalized outputs may partially meet the needs of users, but the matching degree is obviously inferior to that of Plug-in and Subsumes. Note that this generalization must be limited on the direct parent node in the service request output concept, which prevents False Positive caused by excessive generalization; On the other hand, this generalization cannot be performed on input, since the input concept generalization of user requests for service may not satisfy the input constraints, leading to service failure. In summary, the matching degree of the five results is sorted according to the degree of semantic similarity: *Exact* > *Plug-in* > *Subsumes* > *Subsumed-by* > *Logical Fail*.

B. APPROXIMATE LOGIC MATCHING METHOD

SSM based on strict logic reasons and judges the subsumption relations between input and output ontology concepts with formal methods, which is too strict when describing user needs with strong fuzziness, and may lead to excessive exclusion of compatible services, resulting in False Negative. As shown in Figure 2, an example from OWLS-TC4 is demonstrated.

In Figure 2, the user proposes a service request R , hoping to buy a book with a debit card and get confirmation information after the purchase is successful. An e-commerce service S can sell all kinds of documents (including books) to users with credit cards and provide users with receipts and pricing information. With the method described in Section II-A, the result of *Logical Fail* will be returned. But in OWLS-TC4, the user believes that the service and request are matched, where the False Negative occurs. The reason for this problem is that, while books and documents belong to different branches of object in the ontology tree, which are not completely logical mutually exclusive, credit cards and debit cards form strict logical mutual exclusion because of the property of whether they can overdraw. Klusch found that if the strict logic matching method was applied in OWLS-TC4, about 45% of the services in each service request set will be incorrectly categorized as mismatches [3]. In fact, if the user could give up the concepts which cause logical subsumption failure and only keep part of the concepts for calculating subsumption relations approximately, the approximate logical matching degree between semantic service and user request can be determined, where the erroneous judgement of strict logic matching result is compensated.

Inspired by [2], Logical Concept Constriction and Abduction are introduced when calculating approximate logical matching degree. The compatible parts CC (Concept Compatible) and the incompatible part CI (Concept Incompatible) of concept C and D can be found by logical concept constriction, which abandons some property constraints of the concepts. The Structural Abduction $SAC(CC, D)$ of CC relative to D can be obtained by performing abduction on the compatible part CC , and the approximate concept C' of C relative to D can be obtained from the mapping $\sigma[\bar{H}, H](CC^h)$. By computing the information quantity differences among C , C' and D , the approximate quantity of information can be calculated, based on which the approximate logical matching degree between the service and the request can be computed.

Definition 2. Logical concept constriction and abduction. The constriction of the concept C relative to the concept D is

$$LCC(C, D) = (CI, CC), \quad (5)$$

where CC represents the compatible part of C relative to D , and CI represents the incompatible part of C relative to D . Abduction concept CC^h is derived from C by following expansion:

$$CC^h = h_0 \sqcap rew(k), \quad (6)$$

where $rew(A) = A$, $rew(\neg A) = \neg A$, $rew(C \sqcap D) = rew(D) \sqcap rew(D)$ and $rew(\exists R.C) = \exists R.(h_i \sqcap rew(C))$. A is the atomic concept after expanding CC in the ontology tree. The structural abduction of CC relative to D is expressed as

$$SAC(CC, D) = H = (H_0, \dots, H_n), \quad (7)$$

where $\sigma[\bar{H}, H](CC^h) \sqsubseteq D$ and $\sigma[\bar{H}, H](CC^h) \sqsubseteq D$. The approximate concept of C relative to D is denoted as

$$C' = \sigma[\bar{H}, H](CC^h), \quad (8)$$

where $\sigma[\bar{H}, H] = \{h_0 \mapsto H_0, \dots, h_n \mapsto H_n\}$. The definition of approximate logical concept subsumption can be obtained from the definition and calculation method above.

Definition 3. Approximate logical concept subsumption. Concept C is approximate logic concept subsumed by concept D , if and only if the approximate concept of C relative to D is contained by the D logically:

$$\sigma[\bar{H}, H](CC^h) \sqsubseteq D, \quad (9)$$

where $C' = \sigma[\bar{H}, H](CC^h)$, $(CI, CC) = LCC(C, D)$ and $H = SAC(CC, D)$.

The following two concepts of debit cards and credit cards are taken as examples to illustrate the derivation process of approximate logical concepts subsumption. Suppose that the user agrees to relax restrictions on the debit card on the property of no overdraft, then

$$\begin{aligned} (CI, CC) &= LCC(DebitCard, CreditCard) \\ &= (\neg \exists allows.Credit^p, MediumOfExchange \\ &\quad \sqcap \exists issuedBy.Bank^p). \end{aligned} \quad (10)$$

From Definition 2, $CC^h = h_0 \sqcap Object^p \sqcap \exists hasValue.(h_1 \sqcap Value^p) \sqcap \exists issuedBy.(h_2 \sqcap Bank^p)$. And from the algorithm given in [2], we can obtain $\bar{H} = (h_0, h_1, h_2)$, $H = SAC(DebitCard, CreditCard) = (\exists allows.Credit^p, \perp, Company^p)$. Then it can be obtained that $\sigma[\bar{H}, H] = \{h_0 \mapsto \exists allows.Credit^p, h_1 \mapsto \perp, h_2 \mapsto Company^p\}$. Thus, the approximate concept of debit card is $DebitCard' = \exists allows.Credit \sqcap MediumOfExchange \sqcap \exists issuedBy.(Bank \sqcap Company)$. Obviously, $DebitCard' \sqsubseteq CreditCard$. Therefore, debit cards are approximated logical concept subsumed by the logic of credit cards, that is $DebitCard \sqsubseteq_{AC} CreditCard$.

Note that it is uncertain that which one of two mutually exclusive concepts the logical concept constriction is carried out on, so the approximate logical subsumption relation is also uncertain. As in the last example, abandoning the overdraft attribute in $CreditCard$ approves $CreditCard' = \neg \exists allows.Credit \sqcap MediumOfExchange \sqcap \exists issuedBy.(Bank \sqcap Company)$, thus, $CreditCard \sqsubseteq_{AC} DebitCard$ is valid.

In order to evaluate the matching degree of semantic service based on approximate logic, the approximate logic subsumption degree between concepts is calculated by introducing information quantity [4] to analyze the loss of information caused by concept contraction during approximate

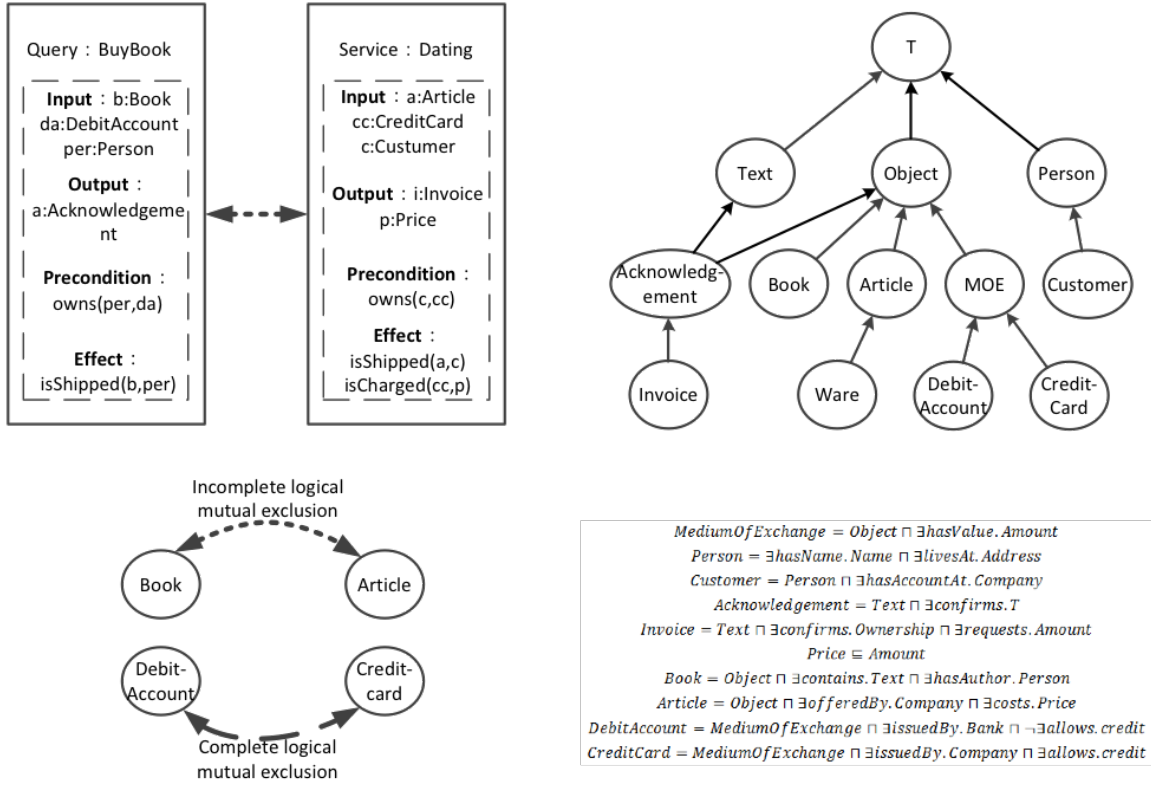


FIGURE 2. False Negative problem in service matching.

concept construction. The concept approximate calculation formula based on information quantity is as follows:

$$Sim_{Info}(C, D) = \frac{2 \times IC(LCA(C, D))}{IC(C) + IC(D)}. \quad (11)$$

The information quantity $IC(C)$ [5] is obtained by the probability of the concept C :

$$IC(C) = -\log P(C), \\ P(C) = n(C)/N, \quad (12)$$

where $n(C)$ is the number of occurrence of all sub-concepts contained in the ontology tree of the concept C . N is the total number of concepts in ontology. Obviously, $Sim_{Info}(C, D) \in [0, 1]$. Based on the above, approximate logic subsumption degree can be calculated as:

$$ALSD(C, D) = Sim_{Info}(C', D) \\ - (1 - Sim_{Info}(C', C)). \quad (13)$$

Because $Sim_{Info}(C, D) \in [0, 1]$, we have $ALSD(C, D) \in [-1, 1]$. Since concept C' , which is an approximate concept derived from logical concept construction and abduction of C , has a direct subsumption relation with concept D , the subsumption relation formed in the ontology tree must be a direct parent-child relationship. Thus, we propose two assumptions about approximate logical matching, which are approximate plug-in matching and approximate subsumed-by matching respectively.

Definition 4. SSM based on approximate logic. Let S and R be semantic service and user service request respectively, and I_S , I_R , O_S and O_R are the sets of input and output semantic concepts of semantic service S and user service request R defined on the same ontology tree respectively. Semantic service matchmaking based on approximate logic divides the matching degree of S and R into the following cases:

1) Approximate plug-in

$$(\forall C \in I_S \exists C' \in I_R : (C, C') \in BCM(I_S, I_R) \\ \wedge C \sqsupseteq_{AC} C') \wedge (\forall D' \in O_R \exists D \in O_S : (D, D') \\ \in BCM(O_S, O_R) \wedge D \sqsubseteq_{AC} D') \quad (14)$$

2) Approximate subsumed-by

$$(\forall C \in I_S \exists C' \in I_R : (C, C') \in BCM(I_S, I_R) \\ \wedge C \sqsupseteq_{AC} C') \wedge (\forall D' \in O_R \exists D \in O_S : (D, D') \\ \in BCM(O_S, O_R) \wedge D \sqsupseteq_{AC} D') \quad (15)$$

By calculating the approximate logic subsumption degree, approximate matching degrees under all kinds of the condition of service approximate logical matching can be obtained,

and the formula is as follows:

$$MatchAL(S, R) = \frac{1}{2} \left(\frac{\sum ALSD(C, C')}{|I_S|} + \frac{\sum ALSD(D, D')}{|O_R|} \right). \quad (16)$$

Since approximate logical subsumption relation is uncertain due to different concept constrictions, semantic services and user requests are likely to conform to both approximate plug-in matching and approximate subsumed-by matching. Therefore, it is necessary to calculate both degrees of service matching under different approximate matching. In which case the matching degree of service is higher and the value is positive, then the matching in that case will be considered as the match between semantic service and user request. If $ALSD(C, D)$ are less than or equal to 0 in the two matching cases, then the semantic service and the user request are considered as not matched on the approximate logic.

C. PRECONDITION AND EFFECT LOGICAL PLUG-IN MATCHING

The input and output of service cannot reflect the functional semantics of service, and this part is usually expressed in the form of precondition and effect of service logic. In addition to semantic explanation on services using ontology, SWRL (Semantic Web Rule Language), PDDL (Planning Domain Definition Language) and other languages which present rules in a semantic way are also used to describe service precondition and effect.

Since only using logical match on the concepts of service input and output will cause False Positive problem, we propose the precondition and effect logical plug-in matching of semantic service, by introducing the method used in LARKS [6].

The following principle is inherited from the software retrieval domain: Logical Specification Plug-in Matching.

Definition 5. Precondition and effect logical plug-in matching. Semantic service and user service request is precondition and effect logical plug-in matching, if and only if the precondition of the user request logical contains the precondition of the semantic service, and the effect of the semantic service logical implies the effect of the user request, which is

$$MatchPE(S, R) \text{ iff } |(P_R \Rightarrow P_S) \wedge (E_S \Rightarrow E_R)|. \quad (17)$$

The logical implication relation between precondition and effect can be determined by the concept of θ -inclusion proposed by [7], which is

$$\begin{aligned} (\forall p_S \in P_S \exists p_R \in P_R : p_R \leq_{\theta} p_S) &\Rightarrow (P_R \Rightarrow P_S), \\ (\forall e_R \in E_R \exists e_S \in E_S : e_S \leq_{\theta} e_R) &\Rightarrow (E_S \Rightarrow E_R). \end{aligned} \quad (18)$$

III. SSM BASED ON NON-LOGICAL METHOD

Service Signature not only contains hasInput, hasOutput, precondition, effect (IOPEs) and other functional parameters, but also contains many non-functional parameters, such as

serviceName, serviceCategory, qualityRating, textDescription and the metadata of the name and location of a service provider. It is more intuitive, easy to compare the degree of similarity between semantic service and the information requested by the user when performing text similarity matching on the service descriptions, which can retrieve the possible matches which are misjudged as mismatched by the logic-based method, which may help to alleviate the problem of False Negative.

A. SERVICE DESCRIPTION TEXT SIMILARITY MATCHING

When using service description text similarity matching, the service description textDescription is usually modeled by the text vector space model. In order to avoid the high-dimensional sparse problem caused by the increase of text amount, we first need to reduce the dimension using Natural Language Processing methods, such as word segmentation, stop word filtering, etc. Then, based on TF-IDF (Term Frequency - Inverse Document Frequency), the service description text similarity is calculated. Next, calculate the weights of text keywords in service and user request respectively, and then the distance between feature vectors can be obtained based on cosine theorem. The TF-IDF formula used in this paper is as follows:

$$\begin{aligned} TF_{i,j} &= \frac{n_{i,j}}{\sum_k n_{k,j}}, \\ IDF_{i,j} &= \log \frac{|D|}{|\{j : w_i \in d_j\}|}, \\ TFIDF_{i,j} &= TF_{i,j} \times IDF_{i,j}. \end{aligned} \quad (19)$$

After getting the TF-IDF weights, we can calculate $Sim_{Description}(S, R)$, which is the similarity of service description texts by cosine theorem.

Due to lack of uniform standards, service providers and service users may have different representations of the service description language, so computing text similarity of the service description may merely exist in the problems of low precision. Therefore, based on the text matching of service description, we introduce the expanded concept of service input and output, and propose an extended text similarity matching method for service description.

Definition 6. Extended concept expression. An extended concept expression of concept is the sequential splicing text of all the intermediate concept nodes which is passed by from the root node of the ontology tree. The ontology tree in Figure 2 is taken as an example, and according to the Definition 6, the extended concept expression of *CreditCard* is *andCreditCard^P(andMediumOfExchange^PObject^P)*. When the extended concept expression is used to calculate the similarity of text, Loss of Information [8] based on similarity is introduced. The losses of information of input and output ($LOI(I_S, I_R)$ and $LOI(O_S, O_R)$) between service and user request in expanded concept expression can be

obtained respectively to measure the difference of expanded concept expression between service and user request:

$$LOI(A, B) = 1 - \frac{|A \cup B| - |A \cap B|}{|A| + |B|}. \quad (20)$$

It can be seen that $LOI(A, B) \in [0, 1]$, and the greater the value is, the higher the similarity is. After calculating the text similarity of the extended concept expression, the extended service description text similarity can be obtained by the following formula:

$$\begin{aligned} Sim_{ExtD}(S, R) &= Sim_{Description}(S, R) \\ &+ LOI(I_S, I_R) \\ &+ LOI(O_S, O_R). \end{aligned} \quad (21)$$

B. ONTOLOGY STRUCTURE SIMILARITY MATCHING

The core idea of ontology matching method is mapping the information of semantic service and user's request to the ontology, and then the similarity of the whole semantic is obtained to realize the SSM by calculating the similarity of the corresponding nodes between ontologies. In general, the similarity of ontology structure can be measured from semantic distance, depth and information quantity.

Semantic distance refers to the length of the shortest path of the two concepts to be compared in the ontology tree through its nearest common ancestor node. The greater the semantic distance is, the greater the semantic difference of concept is. The similarity computation methods of semantic distance are as follows:

$$Sim_{Dist}(c, d) = 1/e^{Dist(c, d)}, \quad (22)$$

where $Dist(c, d)$ represents the length of the shortest path of the concept C and D with passing their nearest common ancestor node.

Depth is also an important indicator to measure similarity. With the increase of the depth of the nearest common ancestor nodes of the two concepts, the number of common attributes of concepts is more, and the semantic similarity is higher. In this paper, the depth similarity is defined as the ratio of the depth of the nearest common ancestor node to the maximum depth of the tree:

$$Sim_{Depth}(c, d) = \frac{Depth(LCA(c, d))}{Depth(Tree)}. \quad (23)$$

$LCA(c, d)$ represents the nearest common ancestor node of C and D. $Depth(Tree)$ represents the maximum depth of the ontology tree.

In addition, the quantity of information can also measure the semantic information size of a concept from the perspective of probability, which helps to analyze the similarity between concepts. The calculation formula of similarity based on information quantity has been given in section II-B.

By integrating semantic distance, depth and information quantity, we can calculate the similarity of ontology structure between two concepts and between two concept sets:

$$\begin{aligned} Sim_{Conc}(c, d) &= Sim_{Dist}(c, d) \\ &+ Sim_{Depth}(c, d) \\ &+ Sim_{Info}(c, d), \end{aligned} \quad (24)$$

$$Sim_{ConcSet}(C, D) = \frac{1}{|C|} \sum_{c \in C} \max_{d \in D} Sim_{Conc}(c, d). \quad (25)$$

And the ontology structure similarity between semantic service and user request can be obtained as follows:

$$\begin{aligned} MatchNL(S, R) &= \frac{1}{2} (Sim_{ConcSet}(I_S, I_R) \\ &+ Sim_{ConcSet}(O_S, O_R)). \end{aligned} \quad (26)$$

IV. FEATURE LEARNING CLASSIFICATION OF SSM BASED ON RANDOM FOREST

In order to realize the combination of different SSM methods and avoid the disadvantages of traditional fixed weight addition method, the matching results obtained by different types of semantic service methods are used as feature vectors of service matching, and the self-learning intelligent service matching classification is realized by using random forest method. Therefore, the SSM problem is transformed into a two-classification problem.

A. RANDOM FOREST METHOD

The idea of random forest method comes from decision tree. Because of its ease of explanation and interpretation, decision tree methods can easily handle the interaction between features, there is no need to worry about outliers or whether the data is linearly separable. However, the decision tree method has the problem of over-fitting. To address this problem, with the voting theory when multiple classifiers are combined into a single classifier, the core idea of random forest method is to generate multiple decision trees without high classification accuracy, which allows all trees to make decision by voting.

When a random forest is constructed, a decision tree is set up for each training subset, and the "forest" formed by several decision trees is generated. Each decision tree does not need pruning to prevent over-fitting. When using random forests for classification, K samples are extracted from the original training set according to the bootstrap sampling method. Then, K decision trees combination classification model is constructed based on these samples. Next, a test sample is classified according to the multiple decision subtrees which are randomly constructed. The results of each subtree are aggregated, and the majority voting method is used to get the final output. That is, when the independent variable X is given, every decision tree in the random forest has 1 vote to decide the optimal classification result [9]. The whole process is shown in Figure 3.

Random forest is usually the winner of many classification problems (usually slightly better than support vector

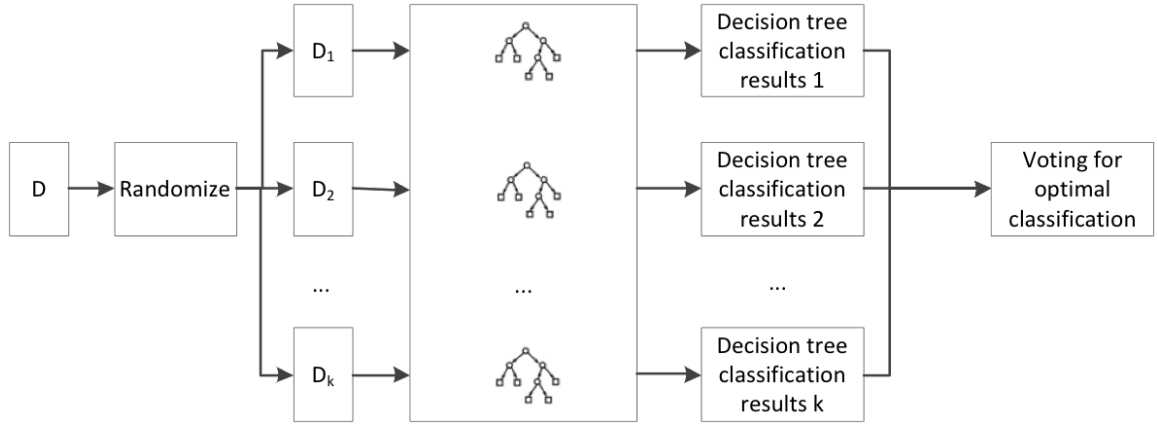


FIGURE 3. Decision process of random forest.

machines), which has fast and scalable characteristics, and avoids the drawbacks of large amount of parameter tuning for, for example, support vector machines. Since the stipulations of an agreement of outliers, 0-1 normalization, etc. have no effect on random forest, the data doesn't need to be preprocessed when random forest is used. Therefore, the use of random forest method is advantageous for different forms of attribute values in service matching feature state vectors [10].

For the random forest training process, we first randomly select 5% of the samples from the OWLS-TC4 test set as training set. The OWLS-TC4 test set contains a large number of evaluated matching pairs of semantic services and user requests. We first run the previous SSM method on these matching pairs, and combine the matching results of different methods into a ten-dimensional feature vectors $X = (x_1, x_2, \dots, x_{10})$, where $x_1, x_2, \dots, x_5 \in \{0, 1\}$ represents the result of SSM based on strict logic. Each item corresponds to a result defined in Definition 1, whose values are 0 or 1, and only one item in all five items can be 1. $x_6, x_7 \in [-1, 1]$ represents the results obtained by SSM based on approximate logic, each of which corresponds to one of the results defined in Definition 4. $x_8 \in \{0, 1\}$ represents the results obtained by precondition and effect logical plug-in matching. $x_9 \in \{0, 1\}$ represents the result of similarity matching of service description text. $x_{10} \in \{0, 1\}$ represents the result of ontology structure similarity matching. $y \in \{-1, 1\}$ represents the results of user evaluation of semantic services matching with user requests. Thus, the service matching feature state vector space $X \times y$ is formed.

B. SERVICE MATCHING FEATURE STATE SPACE

In order to coordinate the matching results of the strict logic and approximate logic matching and improve the training effect of feature data for classifier, it is necessary to do some pruning on the vector space of service matching feature state. The strict logic subsumption relations also satisfies the approximate logic subsumption, so approximate logic match-

ing is usually consistent with strict logic matching judgment on the successful samples of service matching. Therefore, the state space trimming is mainly aimed at the parts that are identified as failure based on strict logic matching, in order to avoid weakening the performance of classifier by the contradiction between strict logic matching and approximate logic matching results. According to the results of matching based on approximate logic and the real results of training samples, four subsets are separated from training samples ($x_5 = 1$), which is strict logical matching failure:

$$\begin{aligned}
 TS_1 &= \{(x_1, \dots, x_{10}, y) \in TS | y = 1 \\
 &\quad \wedge (x_6 \leq 0 \wedge x_7 \leq 0)\}, \\
 TS_2 &= \{(x_1, \dots, x_{10}, y) \in TS | y = 1 \\
 &\quad \wedge (x_6 > 0 \wedge x_7 > 0)\}, \\
 TS_3 &= \{(x_1, \dots, x_{10}, y) \in TS | y = -1 \\
 &\quad \wedge (x_6 \leq 0 \wedge x_7 \leq 0)\}, \\
 TS_4 &= \{(x_1, \dots, x_{10}, y) \in TS | y = -1 \\
 &\quad \wedge (x_6 > 0 \wedge x_7 > 0)\}. \quad (27)
 \end{aligned}$$

In subset TS_1 , $y = 1$ means that the samples are matched by the user, but neither strict logical matching nor approximate logical matching correctly match the result, resulting in False Negative. This will affect the training results of classifiers, so it is necessary to prune TS_1 directly from the state space. In subset TS_2 and TS_3 , approximate logic matching is used to obtain the correct matching result, and their eigenvalues should be rewarded. It is particularly important to note that in TS_2 , when the approximate logical matching results are motivated, the error results (mismatches) of the strict logic matching need to be modified ($x_5 = 0$). In subset TS_4 where $y = -1$, mismatched services and requests are designated as matching success, but they do not affect the correctness of strict logical matching. Therefore, the inverse weight adjustment of the value in the state space is needed to alleviate the influence of False Positive on the training results. The adjustment methods are as follows:

$$\begin{aligned}
TS_2 : & IF & x_6 \geq x_7, \\
& THEN & x_6 = \omega_1 x_6, x_7 = 0, \\
& ELSE & x_6 = 0, x_7 = \omega_2 x_7, \\
TS_3 : & IF & x_6 \geq x_7, \\
& THEN & x_6 = \omega_3 x_6, x_7 = 0, \\
& ELSE & x_6 = 0, x_7 = \omega_4 x_7, \\
TS_4 : & IF & x_6 \geq x_7, \\
& THEN & x_6 = (1 - \omega_1)x_6, x_7 = 0 \\
& ELSE & x_6 = 0, x_7 = (1 - \omega_2)x_7. \quad (28)
\end{aligned}$$

In the selection of incentive weights, inspired by [11] and based on the Bayesian posterior probability, we adjust the positive weights of state space value with the help of inference to the best explanation method. First, hypothesis H_i represents approximate logical matching and correctly identifies the services and requests that can be successfully matched ($i = 1, 2$ represents two results of approximate matching), and explanation E represents services and requests are actually successful matches in the training sets. The posterior probability is used to analyze the consistency between the hypothesis and the explanation, which is used as the incentive weight. Accordingly, the training set state space adjustment weight is set as

$$\omega_1 = P(H_1|E) = P(E|H_1)P(H_1)/P(E), \quad (29)$$

where $P(E|H_1)$ represents the frequency of the matched sample that is identified as the approximate Plug-in matching by approximate logical matching. Similarly, $\omega_2 = P(H_2|E)$, $\omega_3 = P(\neg H_1|\neg E)$ and $\omega_4 = P(\neg H_2|\neg E)$.

V. ADAPTIVE HYBRID SEMANTIC SERVICE MATCHMAKING EXPERIMENT

This section uses public service discovery test set to build a simulation environment for service discovery and matching. By collecting test results and evaluating a series of service discovery algorithms, we analyze the performance and effectiveness of this method.

A. SIMULATION SCENARIOS SETTINGS

Since there is no standard test set for OWL-S service discovery and matching, this paper uses service discovery test data set OWLS-TC4 to construct simulation experiment scenarios. OWLS-TC4 [3] is the fourth version of the OWL-S service test set developed by Klusch et al., which is to support the performance evaluation of semantic service match-making algorithm based on OWL-S description. OWLS-TC4 provides 1083 semantic services described by OWL-S 1.1, which come from 9 different fields: education, health care, food, tourism, communications, economy, weapons, geography and simulation. At the same time, OWLS-TC4 provides a set of 42 test requests for performance evaluation experiments. Part of OWL-TC4 comes from the public IBM UDDI registration center, and there are many institutions

TABLE 1. Selected OWLS-TC4 domain.

Fields	Number of services	Number of requests
tourism	197	6
economy	395	12
education	286	6

and researchers working to improve and expand it to realize semi-automatic conversion from WSDL to OWL-S. In this simulation experiment, services and requests from 3 fields are selected as experimental data sets, as shown in Table 1.

The hardware environment used in the simulation experiment of SSM is a PC with Intel Xeon® E3-1230v2 3.30GHz CPU and 16GB memory. The operating system is Windows 7 SP1 64 bit, which is configured with XAMPP to run the OWLS-TC4 test set. The ontology used in the experiment is based on the semantic dictionary WordNet designed by Princeton University. Based on the S2M2 framework, the semantic services and requests are parsed and the service matching simulation experiments are carried out. The test metrics used in the simulation experiment are precision ratio and recall ratio. Precision ratio refers to the ratio of true positive matches in the results obtained by the service matching method, and recall ratio refers to the ratio of true positive matches in the results obtained by the service matching method to the matching services specified by the test set. The advantages of the proposed algorithm are demonstrated by comparing with the logic matching, approximate logic matching, ontology structure matching and service text matching algorithms.

B. RESULTS ANALYSIS

In order to eliminate the influence of experimental error, 10 simulation experiments were carried out separately, and the average precision and recall of the 10 experiments under different algorithms were calculated in the end.

Note that the strict logic matching method obtains a two-classification result, namely, matching and mismatching, while other methods, including approximate logic matching, ontology structure matching and text service matching, return the order list of the service request and the matching degree. While a common method is setting a threshold to determine whether matching, our experiments are conducted to ensure the effectiveness of the algorithm under relatively fair conditions, in order to avoid the subjective bias caused by the simple threshold setting method. According to the matching service set size obtained by the strict logic matching method, the scale of the matching service set of other methods is demarcated, so all the algorithms are consistent with the number of services in the matching service set. Thus, the difference between precision and recall in fair comparison analysis algorithm is analyzed under relatively fair conditions.

Figure 4 shows the precision performance of 4 typical algorithms and the proposed algorithm in different domains of test set. It can be seen from the figure that the method

based on the service description text similarity matching method results is better than the other three non-hybrid algorithms in precision, while the ontology-based method on structural similarity is the worst. This is because ontology-based structural similarity and logical matching methods are very dependent on the degree of optimization constructed on the ontology tree. These methods are easy to integrate services with large concept ontology distance into the matching service set, leading to the emergence of False Positive, which directly affects the accuracy of service matching. On the other hand, although service description matching is able to improve the precision rely on text similarity, it is easy to cause False Positive phenomenon because of language ambiguity. The adaptive hybrid matching method proposed in this paper can improve the performance of service matching precision to a certain extent through the collaborative analysis of the results obtained by various algorithms.

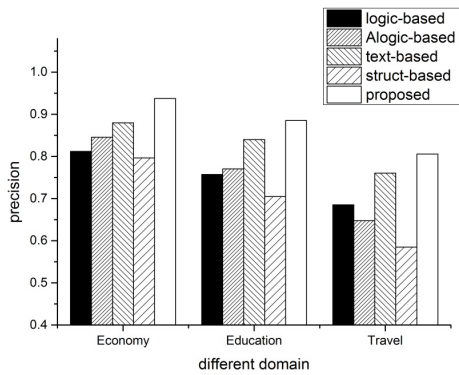


FIGURE 4. The precision of different algorithms in different domains.

Figure 5 shows the recall performance of 4 typical algorithms and the proposed algorithm in different domains of test set. Because of excessively restrictions of strict logic matching, many services that can be identified as matching are judged to be mismatched by strict logical reasoning, which leads to poor performance on recall in the strict logic matching method. Based on approximate logic matching, the concept of ontology constriction and abduction are implemented to relax the restrictions on concept of ontology, so that more potential matching services can be found and the recall performance can be improved. At the same time, ontology structure matching and service text matching can also improve the False Negative phenomenon caused by strict logic matching to some extent. The adaptive hybrid matching method proposed in this paper combines the advantages of various algorithms, and improves the performance of service matching recall by using the advantages of random forest method in the classification learning model.

In Figure 6, the effect of state space pruning based on the consistency of testimony on the average precision ratio is compared. Average precision is the average value of all precision results in all test sets. From the results it can be seen that the state space pruning based on the consistency

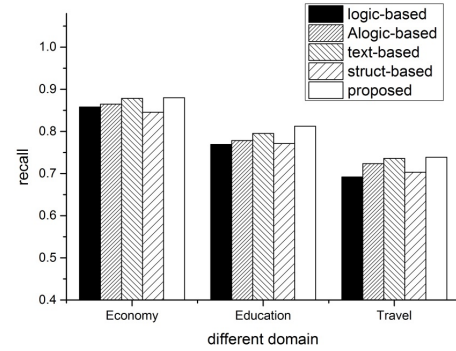


FIGURE 5. The recall of different algorithms in different domains.

of testimony has a significant improvement on the effect of random forest self-learning classification, which shows the effectiveness of the state space pruning.

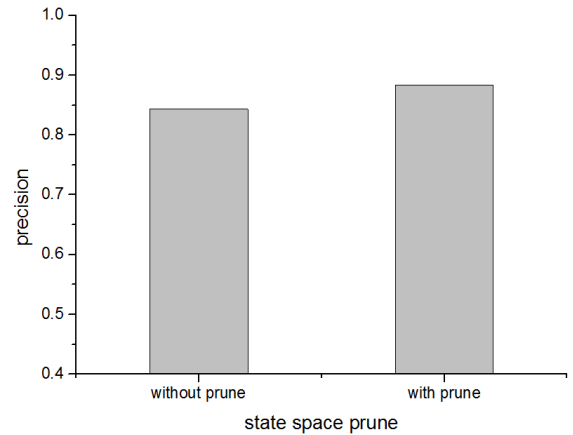


FIGURE 6. The effect on recall with state space prune.

Through a comprehensive analysis of the precision and recall performance of different algorithms in all 3 fields, we can see a special phenomenon that the trend of precision ratio based on strict logic matching and approximate logic matching method is different in the domain test set with higher recall rate and lower recall level. In the tourism service area with low recall level, the performance of precision matching based on strict logic matching is better than that based on approximate logic matching. But in the economic service field with higher recall level, it shows opposite results. The results also show that any single type of service matching method has more or less defects, and it is difficult to ensure good service matching results under all the conditions. The hybrid method combining different types of SSM method is the best scheme to realize the service matching with service dependability guarantee.

VI. SUMMARY

In recent years, with the rapid development of cloud computing technology and service-oriented computing architecture, more and more services have been developed and run online to meet the individual needs of users. Service discovery and matching for service dependability guarantee is the basis and key to build a dependable service composition to meet the needs of users. Semantic service has the characteristics of explicit semantics and machine understandable, so that services can be automated and intelligent interoperability at the semantic level, which helps to improve the dependability level of the service system. According to the False Positive and False Negative are currently facing problems of SSM, we propose an adaptive hybrid semantic service matchmaking based on feature learning classification. SSM oriented service trustworthiness guarantee is carried out based on random forest classification. Simulation results show that the proposed method has some advantages in terms of precision and recall.

REFERENCES

- [1] M. L. Sbdio, D. Martin, and C. Moulin, "Discovering semantic web services using sparql and intelligent agents," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 8, no. 4, pp. 310–328, 2010.
- [2] T. Di Noia, E. Di Sciascio, and F. M. Donini, "A tableaux-based calculus for abduction in expressive description logics: Preliminary results," *Description Logics*, vol. 477, 2009.
- [3] M. Klusch and P. Kapahnke, "The isem matchmaker: A flexible approach for adaptive hybrid semantic service selection," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 15, pp. 1–14, 2012.
- [4] D. Lin et al., "An information-theoretic definition of similarity," in *Icml*, vol. 98, no. 1998. Citeseer, 1998, pp. 296–304.
- [5] H. Amiri, P. Resnik, J. Boyd-Graber, and H. Daumé III, "Learning text pair similarity with context-sensitive autoencoders," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2016, pp. 1882–1892.
- [6] K. Sycara, M. Klusch, S. Widoff, J. Lu, and K. Larks, "Dynamic matchmaking among heterogeneous software agents in cyberspace," *Autonomous Agents and Multi-Agent Systems*, vol. 5, no. 2.
- [7] P. Idestam-Almqvist, "Generalization of clauses under implication," *Journal of Artificial Intelligence Research*, vol. 3, pp. 467–489, 1995.
- [8] M. Klusch, B. Fries, and K. Sycara, "Owls-mx: A hybrid semantic web service matchmaker for owl-s services," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, no. 2, pp. 121–133, 2009.
- [9] C. Lindner, P. A. Bromiley, M. C. Ionita, and T. F. Cootes, "Robust and accurate shape model matching using random forest regression-voting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1862–1874, 2015.
- [10] R. Caruana, N. Karampatziakis, and A. Yessenalina, "An empirical evaluation of supervised learning in high dimensions," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 96–103.
- [11] D. H. Glass, "Inference to the best explanation: A comparison of approaches," in *Proceedings of the AISB 2009 Convention*, Edinburgh, UK. www.aisb.org.uk/convention/aisb09/Proceedings. Citeseer, 2009.



WEI JIANG is engaged in teaching and researching as an associate professor and a master's Supervisor at Harbin Normal University. He is a member of China Computer Federation. His current research interests involve network security and autonomic computing.



HUIQIANG WANG received his received M.E. and Ph.D. degrees from Harbin Engineering University (HEU) in 1985 and 2005, respectively. From 2001 to 2002, He was at Queen's University, Ontario, Canada, as a senior visiting scholar. Now, he is engaged in teaching and researching as a professor and a doctoral advisor at HEU. Up to now, he holds ten Chinese patents. His research interests involve network security, cognitive networks, and autonomic computing.



GUANGSHENG FENG (M'18) received his B.E., M.E. and Ph.D. degrees from Harbin Engineering University (HEU) in 2003, from Harbin Institute Technology (HIT) in 2005, and from HEU in 2009, respectively. Now, he is an associate professor in HEU and researching at mobile data offloading in cellular networks and optimization. He is a full staff at HEU. His research interests involve edge computing, LTE (LTE-A) networks and wireless channel access control.



SHICHEN ZOU received his Dr. degree from Harbin Engineering University in China in 2018. Now he is a full staff at the Huawei Nanjing Research Institute, and his research interests involve network security, cognitive networks.



ZIHAN GAO received his M.E. and Master of Digital Media in 2015 from Communication University of China and University of British Columbia, respectively. He is currently pursuing Ph.D. degree in Harbin Engineering University. His research interests involve virtual and augmented reality, human computer interaction and interactive computer graphics.