# Big Data Analytic Service Discovery using Social Service Network with Domain Ontology and Workflow Awareness

T. H. Akila S. Siriweera, Incheon Paik
School of Computer Science and Engineering
University of Aizu, Aizu-Wakamatsu, Fukushima, Japan
siriweera@gmail.com, paikic@u-aizu.ac.jp

Jia Zhang
Carnegie Mellon University
Silicon Valley, USA
jia.zhang@sv.cmu.edu

Banage T. G. S. Kumara
Dept. of Computing & Information System,
Sabaragamuwa University of Sri Lanka,
Belihuloya, Sri Lanka
btgsk2000@gmail.com

*Abstract*: **In the era of Big Data, data analysis gives strong competition power to enterprises. As services for Big Data Analysis (BDA) become prevalent, analysis services with intelligence and autonomy using automatic service composition show very bright prospects in the BDA market. Service composition consists of four stages: workflow generation, discovery, selection, and execution. In this paper, we propose a novel service discovery approach that considers two key concerns in the discovery domain towards better quality as well as effective service composition. BDA services are fine grained according to the domain and functional behaviors. The services need a domain context-aware and precision-guided discovery approach. Therefore, we propose domain ontology-based service discovery. It is mainly focused on the BDA domain for precise service discovery considering all behavioral signatures between queries and services. As for the second concern, components in composed services depend greatly on each other in situations such as workflow for data analysis. We show that linking services together considering sociability or user preference gives better discovery performance. We propose a Linked Social Service Network (LSSN) with multiple feature attribute-based service discovery for BDA. Our approach combines two advantages, the precision and sociability of Web services. The experimental results show that both of these methods perform well based on their perspectives, better than previous approaches.**

***Keywords:*** **Big data analytics; Web service discovery; service composition; Linked Social Service Networks; domain ontology**

## I. INTRODUCTION

According to the Big Data Value Association of the European Union, governments in Europe could save $149 billion by using Big Data Analytics (BDA) with deep learning to improve operational efficiency. BDA can provide additional value in every sector where it is applied, leading to more efficient and accurate processes [1]. Deep learning is an emerging trend in BDA in parallel with predictive analytics. BDA is thus increasing its importance from every aspect and will open more innovation and opportunities than we expect.

In contrast to the exponentially growing importance of BDA, it is still a very time-consuming and resource-consuming process. Users have invested $44 billion in the BDA domain in 2014 alone from industries that are looking for fruitful analytical results [2]. The BDA process raises extreme challenges in data preparation, modeling for analysis and adoption of the matured models. For example, we must *understand data*, *address data quality*, and *deal with outliers* for more meaningful results in *data preparation*. In *modeling*, the process requires *modeling*, *testing* and *re-modeling* until it satisfies the requirements for analysis. *Adopting a matured model* can be considered the basic meaning of *deployment*, but the trustworthiness of the BDA process must be assured and the model should be precisely articulated to goals and business objectives.

Therefore, we believe that the automation of the BDA process is the most desirable approach to the BDA domain. As the first step, we have previously proposed an intelligent BDA architecture based on Automatic Service Composition (ASC) [3]. ASC is a well-known technology for automating diverse stepped intelligent processes [4]. As the second step, we successfully achieved the planning stage of the ASC, which is the first stage of the ASC process [5]. In this paper, we have addressed the second stage of the ASC process, which is the discovery stage for the BDA domain. We have addressed the two major concerns of effective service discovery and efficient service composition for the BDA procedure.

According to our experience and studies of the BDA domain, we have identified that BDA services are fine grained according to the domain and context. For example, in the *data preparation stage*, the composition system should distinguish *ConvertFileXmltToCsv* vs *ConvertFileExcelToCsv* and in the *modeling stage*, *ClusteringWithKMean* and *ClusteringWithRepetitiveKMean*. The composition system must also incorporate domain context-awareness and precision-guided service discovery for effective service discovery. Therefore, we propose a domain ontology-based service discovery method to identify the precise services from the service registry. Our domain is BDA and the functional behavior of the ontology represents the stages of the Cross-Industry Standard Process for Data Mining (CRISP-DM), which is the foundation of the data science process of the intelligent BDA architecture that we proposed in the first phase of the overall research [3]. The classes and subconcepts of the ontology represent the respective stages and steps of the CRISP-DM process. The domain ontology-based service discovery method aims to exploit semantic meaning of the matrixes that are used in the services to acquire hidden domain knowledge. It also includes a behavioral signature-level approach to ensure the highest possible precision rate. This method is oriented to discover the precise services from the service registry that fulfil effective service discovery for the ASC of BDA process automation.

For our second major concern, we have studied efficient service composition for ASC in the BDA domain. BDA services are highly dependent on each other in situations such as workflow for data analysis. For example*, data preparation*

978-1-5090-2675-3/16 $31.00 © 2016 IEEE
DOI 10.1109/ICWS.2016.49



IEEE computer society

Authorized licensed use limited to: SOUTHERN METHODIST UNIV. Downloaded on December 23,2020 at 05:07:39 UTC from IEEE Xplore. Restrictions apply.

*stage*, *modeling for analysis* and *deployment* are unavoidable stages of the BDA process. Each stage depends on the prior stage and therefore the stages of the BDA process are heavily interdependent. In most cases, BDA service consumers are not limited to a single service request from a service repository; they want to locate multiple services that can work together. This allows peer users to address more complex functions by combining services in an efficient manner. Satisfying complex functions is one of the biggest challenges in the BDA process. This means that according to the workflow, these services are consumed regularly and therefore show strong social interaction with peer services within the service network (registry). Therefore, it is better to use an approach such as linked service-based discovery [6]. This aims to facilitate efficient workflow discovery. Discovering workflows is the most recognized approach to efficient service composition. However, in most cases, such approaches are oriented to achieve solutions that are near optimal rather than more accurate [7]. From the BDA perspective, however, services are fine grained according to the domain and context. We therefore have an additional constraint on BDA for effective service composition for workflow discovery: to maintain accuracy as well as optimality. We propose a Linked Social Service Network (LSSN) with multiple feature attributes-based service discovery to achieve both constraints while seeking efficient service composition for the BDA domain.

The remainder of this paper is structured as follows. In Section II we discuss the preliminaries. In Section III we present the proposed methods for the discovery stage of the ASC process. In Section IV, we describe our implementation and evaluation of these two methods according to their perspectives. In Section V we discuss related work. Section VI concludes the paper.

## II. PRELIMINARIES

In this section, we discuss the two key techniques in both discovery approaches. BDA Domain Ontology is the key background of both approaches; it is the tool for finding fine-grained services as well as managing fine-grained task requirements by manipulating their behavior signature-level descriptions. LSSN is the other important technique in the second process.

### A. Domain Ontology for the BDA

Our aim is to locate fine-grained services in the BDA domain and achieve the highest precision of effective discovery in the BDA domain. Ontologies are used to acquire hidden knowledge and semantic information. However, ontology-based methods raise several concerns when they are dealing with specific domain and cutting-edge information. We can overcome the drawbacks of general ontologies by using a domain ontology.

Our tool thus inherits all the benefits of the domain ontology such as being lightweight and domain oriented, being contextual, coherent, and reusable, and making it easy to design interoperable tools [8]. Figure 1 shows the part of the ontology that we created for the BDA domain [5]. From here onwards, we consider this ontology as the BDA domain ontology.

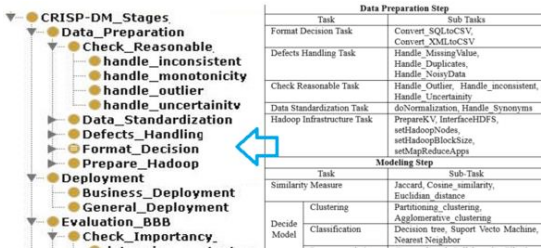


*Figure 1.* Part of Domain Ontology vs Task Table

A given task has its own input and output (I/O) variables. Their parameters can be defined as data properties of the respective named items. These items represent the respective tasks of the BDA process. This approach allows us to manage parametric levels (inputs and output) of tasks throughout the discovery process. However, accuracy is reduced when we use a domain ontology that considers additional matrixes (e.g., I/O, ports) to calculate the similarity [9]. Therefore, we have proposed a heterogeneous ontology approach to cover the three types of matrixes that we encounter during the discovery process.

### B. Linked Social Service Network (LSSN)

Workflow discovery is one of the existing techniques used in service composition. It is widely adopted in services that are highly dependent on workflow, such as bioinformatics, scientific research and industry use cases [10, 11]. The BDA process also depends heavily on ancillary tasks in the workflow; for instance, it cannot do *modeling* without *data preparation* and *analysis* cannot be done without completed *modeling* tasks. Therefore, we proposed an LSSN-based discovery method to address this critical issue. LSSN is one of the most efficient methods used in workflow discovery [12]. Chen et al. [6] proposed a Global Social Service Network (GSSN) creation method. In that method, dependency satisfaction rate (DSR), QoS preference, sociability preference and preferential service connectivity (PSC) are considered in creating the service network.

DSR refers to the functional relationship between services. However, the consideration of DSR in existing methods does not reflect the semantic functional relationship between services. It therefore reduces the tendency to affinity of functionally related services. Therefore, we revised the GSSN creation method by replacing the DSR factor, replacing it with hybrid term similarity (HTS) between services. Then we make Linked Social Service Network with BDA services according to the above method. Here onwards, we call it as the LSSN. According to our experimental results, this approach achieves more effective GSSN with respective to being close to similar services, with a higher response rate of functionality and scalability, as well as discovery ability using the network.



Authorized licensed use limited to: SOUTHERN METHODIST UNIV. Downloaded on December 23,2020 at 05:07:39 UTC from IEEE Xplore. Restrictions apply.

**Proposed method of discovering Workflow using LSSN—*thick-dots-lead-you*:** We propose the method of discovering workflow shown in Figure 2.

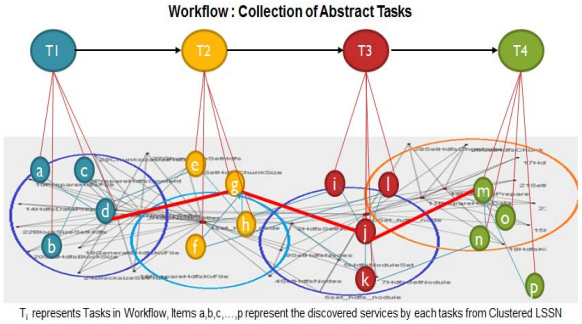


*Figure 2.* Proposed Workflow Discovery Based on LSSN

Assume we are given a workflow containing the tasks $T_1$, $T_2$, $T_3$, and $T_4$. In the first round, we must find the respective services from the clustered LSSN that have the highest semantic similarity with the given tasks in the workflow. Here, for $T_1$ we discovered services *a, b, c*, and *d*; for $T_2$ we discovered *e, f, g*, and *h*; and so on. In the second round, we find the sociability of each service (discovered by $T_i$ tasks) within each cluster and rank the discovered services according to their sociability. Here, sociability means the ratio of number of links within the cluster of a given service. We assume that the services with highest sociability have the highest tendency to link to the respective services that are discovered by ancillary tasks. In this example, service *d* has the highest sociability and the highest probability of linking with *e, f, g,* and *h*. This approach is dramatically easier than the *link-as-you-go* concept introduced in GSSN [6]. We have named our approach *thick-dots-lead-you.* Here, *dots* refer to semantically similar services for given tasks and *thickness* refers to the sociability of these services in the LSSN. Thus, *thick-dots* refers to services that are semantically similar to task $T_i$ and have more sociability. They have more tendency to affinity with more sociable (*thickness*) services (*dots*) discovered by ancillary task $T_{i+1}$ and so on. Thus, this method allows us to achieve a workflow with low energy, which means that it functions in an efficient manner.

## III. BDA Web Service Discovery

Here we discuss the two proposed approaches for the discovery stage of the ASC process for BDA automation.

### A. Domain Ontology-based Service Discovery

As the first concern, BDA services are fine grained according to the domain and their functional behaviors. Services must also be domain context aware and we must have a precision-oriented discovery approach. Figure 3 shows the architecture of the proposed approach. This method facilitates more precise and effective service discovery.

Domain Ontology-based service discovery has three main stages. Stages and their substages are associated with each other. However, it is not necessary to repeat the complete discovery process every time. Thus, Stages 1 and 2 are required only once for a given service registry. Thus, we have streamlined these substages and need only execute Stage 3 for dynamic discovery requirements.

***Scenario 1:*** In Table 1, WF# represents workflow number and T# represents task number. In the table, $T_r$ and $T_s$ are two tasks retrieved from WF1 and $T_j$ and $T_k$ are two tasks retrieved from WF2. A service registry contains the services shown in Table 2, where S# represents service number. We can see that $T_r$ and $T_j$ are requesting different queries but the same I/O parameters. $T_s$ and $T_k$ are different requests and differ by "with" and "without" clauses and one input parameter.

TABLE 1: SAMPLE WORKFLOWS VS TASKS

| WF# | T # | Service name | Output | Input parameters |
|---|---|---|---|---|
| WF1 | $T_r$ | ConvertFileXmlToCsv | csvFile | Dataset |
| | $T_s$ | fuzzy_set_approach _WithOutput Directory | result | csvFile inputClustersFolder outputwWorkingFolder |
| WF2 | $T_j$ | ConvertFileExcelToCsv | csvFile | dataset |
| | $T_k$ | fuzzy_set_approach_Wit houtOutput Directory | result | csvFile, inputClustersFolder |

TABLE 2: SAMPLE SERVICES AND THEIR I/O PARAMETERS

| S# | ServiceName | Output | Input parameters |
|---|---|---|---|
| S0 | convertXmlToCsv | csvFile | dataset |
| S1 | convertXlToCsv | csvFile | dataset |
| S2 | ConvertExcelToCsv | csvFile | dataset |
| S3 | fuzzySetApproach WithOutputDirectory | File | inputVectorsFolder, inputClustersFolder, outputwWorkingFolder |
| S4 | fuzzySetApproach WithoutOutputDirectory | File | inputVectorsFolder, inputClustersFolder, |
| S5 | fuzzySetApproach WithThresholdValue | File | inputVectorsFolder,inputClustersFolder,thresholdValue |

**Stage 1: Initial Setup**

*a.* *Build the BDA domain ontology:* We proposed the domain ontology mentioned in Figure 1.

*b.* *Prepare the service registry:* We have prepared the service registry with BDA services.

**Stage 2: Clustering**

*c.* *Calculate the HTS between services:* We have used an improved version of our previous method for calculating HTS between services [13]. It allows us to reduce noise while dealing with I/O matrixes. It is briefly explained under *Step 3* of the next stage.

*d.* *Cluster the service registry:* Clustering services is an early step of service discovery. It is important in calculating more precise service similarity. It is more efficient to consider ontology relationship between terms in services [13]. In our approach, we improve the existing HTS method of calculating service similarity. It has a big impact on reducing the noise during the similarity calculation process.



Authorized licensed use limited to: SOUTHERN METHODIST UNIV. Downloaded on December 23,2020 at 05:07:39 UTC from IEEE Xplore. Restrictions apply.

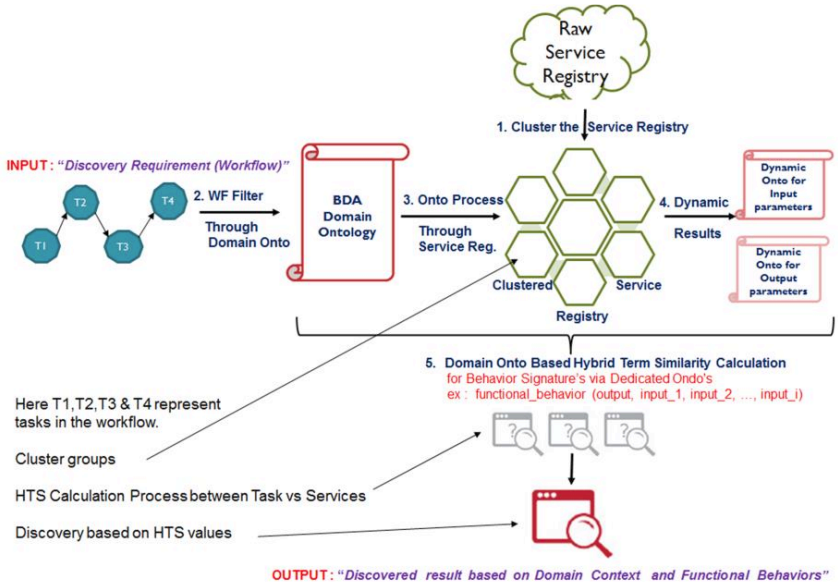


*Figure 3.* Architecture of Proposed Domain Ontology Method

**Stage 3: Discovery**

The Workflow Generator creates an abstract workflow to satisfy the functional properties of a user request in the planning stage of the ASC framework. We can define a plan for the planning problem as $\pi(\text{Task}_1, \text{Task}_2, \ldots \text{Task}_n)$, where $\text{Task}_i$ is a classical action. The sequence of actions will form an abstract workflow guiding service composition. The idea of service discovery is to find candidate Web services for each action $\text{Task}_i$ of the abstract workflow. Service discovery uses the weights of similarity values of a query (tasks) vs services in the service registry. It is the abstract description of the discovery process. Below, we discuss the three main steps of the discovery stage.

e. *Find the most suitable cluster group:* We discover the most suitable cluster groups for the respective requests $\text{Task}_i$. We calculate the semantic similarity between all cluster centers vs the respective requests in $\text{Task}_i$. Then we select the most suitable cluster group for $\text{Task}_i$, based on the value of similarity. We have briefly explained the proposed method to calculate the similarity below.

f. *Calculate similarity: task vs cluster group:* We proposed a way to calculate the similarity based on HTS. It contains three major steps. Step 1: Semantic feature extraction; Step 2: Ontology learning; and Step 3: Feature similarity calculation.

*Step 1: Semantic Feature Extraction:* In calculating the similarity between a Web service and task, we extract the service features (inputs, outputs and service names) of Web services from WSDL and task features (inputs, outputs and task name) from the BDA Domain Ontology.

*Step 2: Ontology learning—create dynamic ontologies:* We use an ontology learning process for input and output features. We first identify complex terms, calculate TF-IDF and rank them. We then process these using an "Ontology Generation" method.

*Step 3: Feature Similarity Calculation:* We have improved our previous HTS method, as mentioned earlier [13]. The new approach is far more accurate than the existing HTS method because it uses additional constraints such as I/O matrixes during the calculation of similarity.

*Distinguishing fine-grained services:* As an example, we take the *ConvertFileXmltToCsv* and *ConvertFileExcelToCsv* services. According to the proposed method, we tokenize the terms (capital letters and hyphens) of the given concepts and create the ontology shown in Figure 4. Then, it is easy to distinguish the difference between the *Excel* and *Xl* terms used in Web service names. With this approach, we have improved accuracy and reduced noise dramatically.

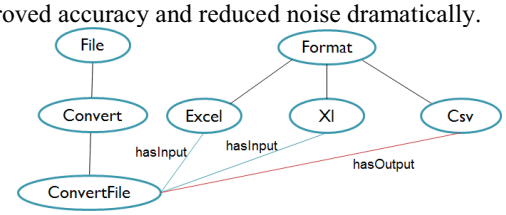


*Figure 4. Sample Ontology Generated for the Two Services*

g. *Discovery:* In this step, we sort the results of the previous step and pick the highest-ranked similar set of services as the discovered services for the given tasks.

According to Scenario 1, we create a service registry with the six services mentioned, continue the above steps, and finally compute the similarity values. Here we sorted the top three results of tasks vs services according to their rank.

The top three for $T_r$ are S0, S1, and S2. For $T_j$, they are S2, S1, and S0. For $T_s$, the top three are S3, S4, and S5 and for $T_k$, they are S4, S3, and S5.



Authorized licensed use limited to: SOUTHERN METHODIST UNIV. Downloaded on December 23,2020 at 05:07:39 UTC from IEEE Xplore. Restrictions apply.

This show how we have precisely discovered the required services among fine-grained services (compare $T_s$ with $T_k$) and how fine-grained task requirements discovered despite their similar functionalities only change behavior signatures (compare the top three results of $T_r$ vs $T_j$ and $T_s$ vs $T_k$.)

### B. LSSN with Multiple Feature Similarity-based Service Discovery

This method is designed to facilitate workflow discovery for efficient service composition by maintaining two critical factors, which are difficult to maintain in parallel. These are near-optimum solutions and highest precision. These two factors are important requirements of discovering BDA services to satisfy workflows based on the ASC technology. Therefore, we use two key techniques to address these two concerns.

*Scenario 2:* Assume that we have a workflow containing the following tasks. T1 = *ConvertFileXmlToCsv,* T2 = *fuzzy_set,* T3 = *evaluate_model.* These tasks represent *Data_Correction, Data_Modelling* and *Analysis*. To satisfy the given workflow, our registries have the following services: S0 = *ConvertXmlToCsv,* S1 = *ConvertXlToCsv,* S2 = *ConvertFileXlToCsv,* S3 = *fuzzySetApproachWithOutputDirec,* S4 = *fuzzySetWithoutOutputDirec,* S5 = *fuzzySetWithThreshold,* S6 = *EvaluateModel,* S7 = *AssessModel,* and S8 = *CheckModel*. Then we summarize those services under each task. This creates an LSSN based on their respective features, as described in Section II(B). Figure 5 shows the resulting LSSN for the given services. Then we follow the steps below for the discovery.

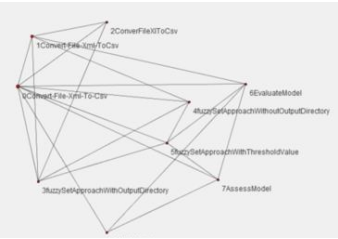


Figure 5: Discovered Path of Workflow based on SSV

Figure 6 displays the architecture of the proposed method. It contains the same three main stages as the previous method.

**Stage 1: Initial Setup**

a. *Build the BDA domain ontology:* We mentioned the domain ontology method in Section II(A).
b. *Create the LSSN:* This is described in Section II(B). When we create the LSSN, we first calculate the *Quality of Social Links (QSL)* between services. Network wiring between services is based on these QSL values.

*Calculate QSL between services:* We calculate the Linked Weights between services. Given service R and a set of target services $T_n$, the quality of the social link between R and $T_n$ services a $QSL(R,T_n)$ and can be denoted as follows:

$$QSL(R,T_n) = Q_{PSC}(R,T_n) \times [Q_{QOS}(R,T_n)^L \times W_1] + [Sim(R,T_n)^L \times W_2] + [Q_{QSP}(R,T_n)^L \times W_3)]$$

Here, $Q_{PSC}(R,T_n)$ denotes the quality of preferential social connectivity (PSC) of R and $T_n$, $Q_{QOS}(R,T_n)$ denotes the QOS of R and $T_n$, $Sim(R,T_n)$ denotes the HTS value between R and $T_n$, and $Q_{QSP}(R,T_n)$ denotes the quality of sociability preference (QSP) between R and $T_n$. L, $W_1$, $W_2$ and $W_3$ are constants. L=1, $W_1 = 0.2$, $W_2 = 0.4$ and $W_3 = 0.4$.

*$Q_{PSC}(R,T_n)$:* We ensure the quality of the social link by linking well-known popular services with higher connectivity. Those services then have higher probabilities of being linked by other services.

*$Q_{QSP}(R,T_n)$:* We ensure the quality of the social links by considering their past behavior (which services have worked together in workflows) and future behavior (which services will be used together). Thus, services that have higher frequency of interaction are more likely to link together.

*$Sim(R,T_n)$:* We ensure that semantically similar services link with each other.

Thus, we can ensure that the creation of the resulting network is based on the functionality, sociability, preferential connectivity and QoS between services. If the services have higher semantical similarity, or have worked together in workflows in the past, or are popular, then they have stronger links. According to Scenario 2, this service registry does not initially possess QSP or PSC values. However, it has functional similarities "*$Sim(R,T_n)$*" between services. Then the resulting LSSN is shown in Figure 5. Once it has created the network and peer users have started to use services for several workflows, then edges (links) between services will change according to the selections made. Thus, "most commonly associated services in workflows," "has more links," and "semantic similarity" imply higher linking probabilities to a given service in the network.

**Stage 2: Clustering**

We also use clustering to reduce the search space. We use the following two steps to create clusters in the LSSN.

c. *Cluster the service registry:* We used the method described in Section d of A. QSL (calculation method from [6]) values are inputs to the cluster method.

**Stage 3: Discovery**

d. *Find the most suitable cluster group:* We used the method described under the domain ontology method, Sec *e* of A.
e. *Find Sociable Similarity Value (SSV) of tasks vs cluster group members:* Here we accumulate the sociability and semantic similarity into one value. We call it the Sociable Similarity Value (SSV). SSV calculation is as follows.

$$SSV(T_i,S_k) = F_{sim} \times Sim(T_i,S_k) + F_{soc} \times Soc(S_k,C_g)$$

Here, $F_{sim}$, a "similarity factor" is a constant. Our experiments showed that $F_{sim} = 0.8$ provides the best fit.

$Sim(T_i,S_k)$ = "semantic similarity" between Task $T_i$ and Service $S_k$, calculated based on the domain ontology



Authorized licensed use limited to: SOUTHERN METHODIST UNIV. Downloaded on December 23,2020 at 05:07:39 UTC from IEEE Xplore. Restrictions apply.

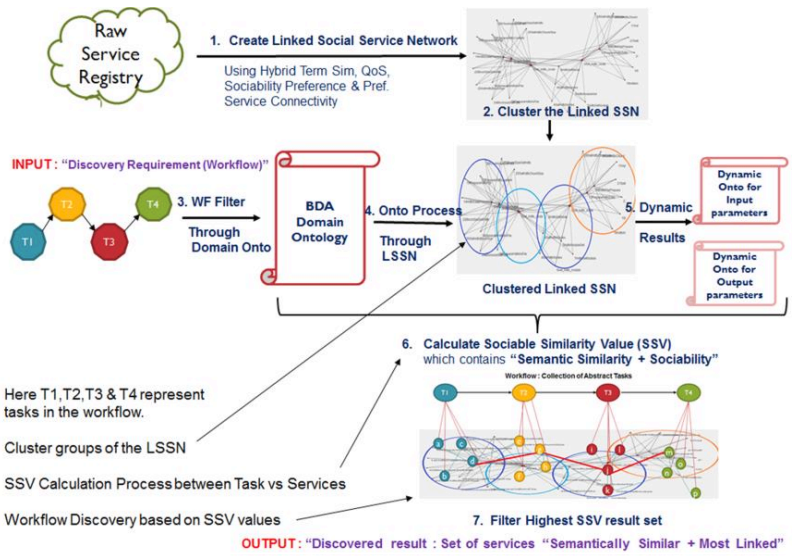


*Figure 6.* Architecture of Proposed LSSN based Discovery

discussed in Section f of A.

$F_{soc}$ = "Sociability Factor" = 1 – [$F_{sim}$ × Sim($T_i$,$S_k$)]. Here, Soc($S_k$,$C_g$) = Links ($S_k$,$C_g$) / Links ($C_g$);

Links($S_k$,$C_g$) = Number of Links populated by Service $S_k$ in cluster group $C_g$ in the LSSN and Links($C_g$) = the total number of links within cluster group $C_g$ in LSSN.

*f. Discovering services:* In this step, we sort the results of the above step and pick the set of services with the highest SSV as the discovered services for the given tasks.

According to Scenario 2, we create a service registry with the described services, continue the above steps, and calculate the SSV values. We next sort the top three results of tasks vs services according to their ranks. S0, S1, and S2 are the top three services for T1; S5, S3, and S4 are the top three services for T2; and S6, S7, and S8 are the top three services for T3. To make Figure 6 easier to understand, we have circled the respective services according to their SSV rank. The smallest circles show the third service and the largest circles show the first service. Thereafter, we followed the *thick-dots-lead-you* method based on the highest SSV values for each task. As Figure 7 shows, it is easy to identify the workflow for Scenario 2. This shows how *thick-dots* among discovered services for respective tasks allow us to achieve the workflow easily for Scenario 2. This proves that it is easy to discover the workflow while maintaining the two critical factors of discovering the requirements of the ASC process for the BDA domain. The process maintains higher precision as well as achieving the workflow. Our approach is designed to achieve a precise optimum rather than a near optimum.

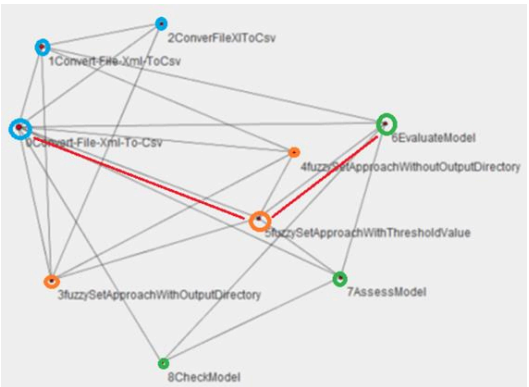


Figure 7: Discovered Path of Workflow Based on SSV

## IV. Experiment and Evaluation

To evaluate the quality, accuracy, and effectiveness of the precision we achieve, we analyzed the performance of both approaches together with two semantic methods. The two existing approaches are heterogeneous ontology-based service discovery (represented as "Onto" in the results graphs) and GSSN-based Workflow service discovery (represented as "Gssn" in the results graphs). In the analysis process, we evaluate the quality of discoveries in terms of *precision rate*, accuracy of discoveries in terms of *recall*, and validate the result using the *balanced F-measure*.

To evaluate the efficiency and effectiveness of workflow discoveries, we compare our LSSN approach with the existing workflow-discovering GSSN method. In that analysis, we evaluate the efficiency by using the *success rate* and effectiveness by evaluating *scalability* between LSSN and GSSN networks.

We used 300 BDA services with 30 queries (abstract tasks) that are designed to be used in BDA workflows. Ten different types of services are available for each query considering different aspects of their behaviors and requirements.

We evaluate the quality of discovery by calculating the *precision rate* of the discovery. We increased the number of



Authorized licensed use limited to: SOUTHERN METHODIST UNIV. Downloaded on December 23,2020 at 05:07:39 UTC from IEEE Xplore. Restrictions apply.

services and measured the precision. Figure 8 shows the results of that experiment.

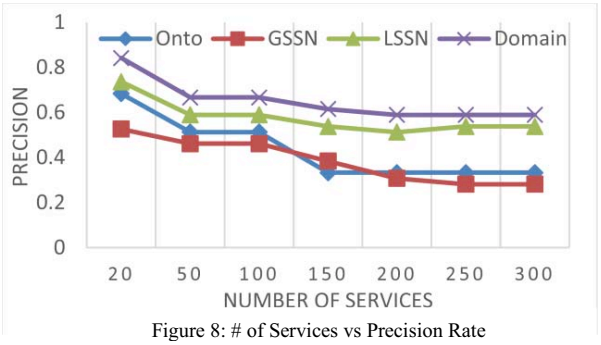


Figure 8: # of Services vs Precision Rate

We evaluate the efficiency of all four approaches by increasing the number of tasks from two to seven and calculating the micro-average of the *recall rate*. The results are shown in Figure 9.

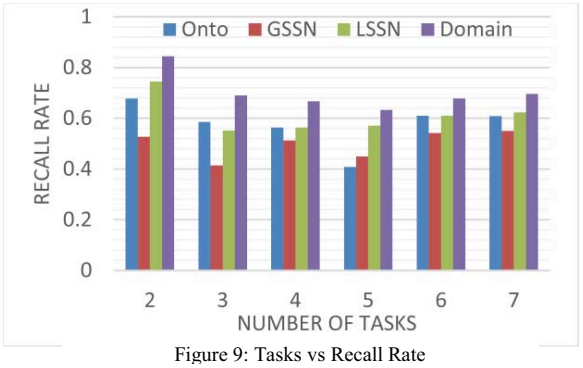


Figure 9: Tasks vs Recall Rate

When we compare Figures 8 and 9, it is clear that our proposed methods perform better than the two existing methods. However, heterogeneous ontology methods are used to achieve higher precision when dealing with multiple matrixes such as I/O and service name [8]. In the BDA domain, the method shows a low *precision rate* because it has an additional constraint, the fine-grained information about services. Nevertheless, our proposed domain ontology method successfully overcomes that challenge and it has achieved the highest *precision rate* among these four methods. Thus, our mission to discover fine-grained services for effective discovery has been achieved successfully.

TABLE 3: BALANCE F-MEASURE TABLE

| Service | GSSN | Onto | LSSN | Domain |
|---|---|---|---|---|
| **20** | 0.526316 | 0.684211 | 0.736842 | 0.842105 |
| **50** | 0.461538 | 0.512821 | 0.589744 | 0.666667 |
| **100** | 0.461538 | 0.512821 | 0.589744 | 0.666667 |
| **150** | 0.384615 | 0.333333 | 0.538462 | 0.615385 |
| **200** | 0.307692 | 0.333333 | 0.512821 | 0.589744 |
| **250** | 0.282051 | 0.333333 | 0.538462 | 0.589744 |
| **300** | 0.282051 | 0.333333 | 0.538462 | 0.589744 |

Next, we evaluate the trade-off between precision and recall as a further test of our experimental results. Table 3 shows the results. We measure the *balance F-measure*; F = 2PR/ (P+R). Here P = precision and R = recall. We can see that the results have not deviated from the precision or recall rates already obtained. We can therefore affirm that our results are accurate and can be assured based on the F-measure table.

LSSN and GSSN are the methods that we have used for workflow discovery. Therefore, we compute the *success rate* of the workflow discoveries of LSSN vs GSSN with increasing numbers of services. Figure 10 shows the results. We apply *thick-dots-lead-you* in both cases.

When we compare Figures 8 and 9, the LSSN method consistently comes second. The perspective behind our proposed domain ontology method and the domain ontology were designed to address only fine-grained concerns.

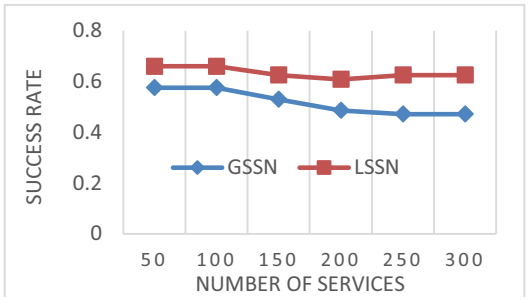


Figure 10: Success Rate between LSSN vs GSSN

Nevertheless, we considered an additional constraint, the sociability within the LSSN method. Because LSSN is in second place and performing better than the heterogeneous ontology and GSSN methods, even with this additional constraint it performs better than the other two methods. Moreover, it is clear that LSSN takes first place in Figure 10. That means that the workflow perspective is the best method because it has achieved the highest precision between the two workflow discovery methods (LSSN and GSSN) and is also successful over LSSN and GSSN for workflow discovery. It can be considered a breakthrough when we consider the near optimization of conventional workflow discoveries: the LSSN method finds a precise optimum rather than a near optimum.

Finally, we compute the *scalability* of both approaches. It is one of the most important factors for both GSSN and LSSN, because both work in networks. Figure 11 shows that LSSN's link average is less than that of GSSN and it proves that our LSSN maintains good scalability with increasing numbers of services in the network. This is a good sign, because LSSN can provide satisfactory services to peer users for their needs (such as service discovery, recommendation, and composition) than the existing GSSN.

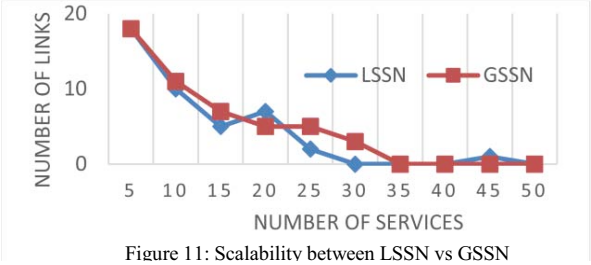


Figure 11: Scalability between LSSN vs GSSN



Authorized licensed use limited to: SOUTHERN METHODIST UNIV. Downloaded on December 23,2020 at 05:07:39 UTC from IEEE Xplore. Restrictions apply.

## V. Related Work

The literature on Web service discovery for the BDA domain is scarce. Alarcon et al. [14] presented the REST Web service description for graph-based service discovery in the Big Data domain. They proposed the graph-based method to overcome the limitations of REST services in certain ways. An approach by Akila et al. [15] was based on ontology for the BDA domain. An agent-based approach was proposed by Rajendran et al. [16]. They proposed a discovery framework consisting of separate agents for ranking services based on QoS certificates obtained from publishers. Johnson et al. [17] proposed a service discovery method designed to be used in heterogeneous networks. This method was specifically proposed for use in military domains because networks in that domain are heterogeneous. Rong and Liu [18] proposed a context-aware approach to overcome information loss during the transformation from the user's requests to a formalized one. Hai-Cheng and Hong [19] proposed a layer-based semantic method to improve efficiency of matching within a repository.

Interesting hybrid approaches have been proposed by several researchers. Tsai et al. [20] proposed a keyword- and ontology-based method. A multiple-criteria decision-making method was proposed by Saaty [21] based on text and ontology. It considers all associated attributes between query and services. It seems noisy because it considers functional and nonfunctional aspects. Wen et al. [22] presented a Web service discovery method based on semantics and clustering. An ontology-based workflow composition and discovery method was proposed by Karakoe et al. [23]. Their proposed model allows users to select relevant service types.

## VI. Conclusion

The BDA domain is still evolving and BDA as a Service (BDAaaS) has just started its legendary journey. BDA service publishing and composition are two very critical factors in providing comprehensive services for the BDA domain based on Web services. BDA service discovery plays a crucial role, because it is considered the first and most important step of successful composition and publishing. We have considered two critical factors that are affected by BDA Web service discovery. Experimental results show that our approaches perform better than other approaches in the missions conferred on them. Thus, we have successfully completed the second stage of our architecture to automate BDA [3]. In future work, we plan to move to selection and complete the full cycle of the ASC for the BDA process. We hope this will be a breakthrough for the BDA domain and for industry, which is suffering most at present because BDA is a heavy-duty task requiring multiple resources and much time.

## References


[1] http://www.bdva.eu/sites/default/files/europeanbigdatavaluepartnership_sria__v1_0_final.pdf

[2] F. Castanedo, "Data Preparation in the Big Data Era", O'Reilly-2015, http://www.tamr.com/wp-content/uploads/2015/09/Data_Preparation_in_the_Big_Data_Era_Tamr.pdf

[3] T. H. Akila, S. Siriweera, I. Paik, B. T. G. S. Kumara, K. R. C. Koswatta: "Architecture for Intelligent Big Data Analysis based on Automatic Service Composition." International Journal of Big Data (ISSN 2326-442X), Accepted August 2015.

[4] I. Paik, W. Chen, and M. N. Huhns, "A Scalable Architecture for Automatic Service Composition," IEEE Transactions on Services Computing, vol. 7, no. 1, pp. 82–95, Jan.–Mar. 2014.

[5] Banage, T. G. S. Kumara, Incheon Paik, Jia Zhang, T. H. A. S. Siriweera, K R. C. Koswatte: "Ontology-Based Workflow Generation for Intelligent Big Data Analytics." ICWS 2015: pp. 495–502.

[6] W. Chen, I. Paik, P C. K. Hung: "Constructing a Global Social Service Network for Better Quality of Web Service Discovery." IEEE Transactions on Services Computing vol. 8, no. 2, pp. 284–298 (2015).

[7] T. Yu , Y. Zhang , K. Lin, Efficient algorithms for Web services selection with end-to-end QoS constraints, ACM Transactions on the Web (TWEB), v.1 n.1, p.6-es, May 2007

[8] M. K. Bergman, "Domain Ontologies Development Methodology," http://wiki.opensemanticframework.org/index.php/Lightweight,_Domain_Ontologies_Development_Methodology, 2010

[9] Y. Wang, R. Zhang, J.Wang. "Ontology-based heterogeneous messages matching in web services." Wireless Communications, Networking and Mobile Computing, 2008. WiCOM'08. 4th International Conference on. IEEE, 2008.

[10] S. Dustdar, R. Gombotz, "Discovering web service workflows using web services interaction mining," Int. J. Business Process Integration and Management, vol. 1, no. 4, pp. 256–266, 2006.

[11] A. Musen, "Domain Ontologies in Software Engineering: Use of Protégé with the EON Architecture," Methods of Information in Medicine, vol. 37, pp. 540–550, 1998

[12] W. Chen, I. Paik: "Toward Better Quality of Service Composition Based on a Global Social Service Network." IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 5, pp. 1466–1476 (2015).

[13] B. T. G. S. Kumara, I. Paik, W. Chen, K. H. Ryu: "Web Service Clustering using a Hybrid Term-Similarity Measure with Ontology Learning." Int. J. Web Service Res., vol. 11, no. 2, pp. 24–45 (2014).

[14] R. Alarcon, R. Saffie, N. Kolas, J. Cabello, "REST Web Service Description for Graph-Based Service Discovery," Engineering the Web in the Big Data Era. Lecture Notes in Computer Science, Vol. 9114, pp 461–478. 10 June 2015.

[15] T. H. Akila, S. Siriweera, I. Paik, B. T. G. S. Kumara, "Ontology-based Service Discovery for Intelligent Big Data Analytics," IEEE International Conference on Awareness Science and Technology (iCAST 2015), Qinhuangdao, China, Sep. 2015

[16] T. Rajendran, P. Balasubramanie, "An Optimal Agent-Based Architecture for Dynamic Web Service Discovery with QoS", Second International conference on Computing, Communication and Networking Technologies, IEEEXplore 2010.

[17] F.Johnsen, T. Hafsøe, M.Skjegstad, "Web services and service discovery in military networks", 14th ICCRTS, Washington D.C, US, June 2009

[18] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language (WSDL) 1.1", W3C Note, 2001.

[19] G. Wen-Yue, Q. Hai-Cheng, C. Hong, "Semantic Web Service Discovery Algorithm and Its Application on the Intelligent Automotive Manufacturing System", International Conference on Information Management and Engineering, IEEEXplore, 2010.

[20] Y. H. Tsai, S.-Y. Hwang, Y. Tang, "A Hybrid Approach to Automatic Web Services Discovery," In International Joint Conference on Service Sciences, IEEEXplore, 2011.

[21] T. Saaty,"Decision Making with Analytic Hierarchy Process", International Journal of Services Sciences, vol. 1, pp. 83–98, 2008.

[22] T. Wen, G. Sheng, Y. Li, and Q. Guo, "Research on Web Service Discovery with Semantics and Clustering," in Proc. 6th IEEE Joint International IT and AI Conference, pp. 62–67, Aug. 2011.

[23] E. Karakoc, K. Kardas, K. P. Senkul, "A Workflow-Based Web Service Composition System" Web Intelligence and Intelligent Agent Technology Workshops, 2006 (WI-IAT 2006). Workshops 2006 IEEE/WIC/ACM International Conference, Dec. 2006




Authorized licensed use limited to: SOUTHERN METHODIST UNIV. Downloaded on December 23,2020 at 05:07:39 UTC from IEEE Xplore. Restrictions apply.