BIG DATA

Big Data

Data Asset Valuation on Data Lineage Graph

Journal:	*Big Data*
Manuscript ID	BIG-2024-0115
Manuscript Type:	Original Article
Date Submitted by the Author:	28-Jun-2024
Complete List of Authors:	Chen, Yunpeng; Central South University, Li, Xuanjing; Central South University Ma, Chenrui; Central South University Zhao, Xin; Central South University Zhao, Ying; Central South University Zhou, Fangfang; Central South University
Keywords:	Big data analytics, Data mining, Structured data
Manuscript Keywords (Search Terms):	data asset valuation, data lineage, data asset, visualization

·Data Asset Valuation on Data Lineage Graph

Yunpeng Chen[1], Xuanjing Li[1], Chenrui Ma[1], Xin Zhao[1], Ying Zhao[1], Fangfang Zhou[1,*]

Abstract—Core data assets are high-value data tables critical for enterprise business activities. Data asset valuation is the process of evaluating the economic value of data assets within an enterprise. This can be used to identify core data assets. However, efficient, accurate, and generalizable core data asset valuation methods are lacking. Data lineage graphs (DLGs) effectively assist in facilitating core data asset identification. In this work, we propose a method named graph-based data asset valuation (GBDAV). A set of important criteria are designed to guide the valuation of data assets in DLGs. A formula is established to quantify the value of data assets in DLGs. A heat diffusion model simulates data transformations in DLGs, defining the dynamic value for each table based on heat input and output. Two heat transfer matrices and iteratively update dynamic value until thermal equilibrium is reached. A series of experiments are conducted to evaluate the effectiveness of the proposed GBDAV.

Index Terms—Data asset, Data asset valuation, data lineage, visualization

1. Introduction

Data assets have emerged as new assets in enterprises[1,2]. They are valuable information resources owned by enterprises capable of generating economic value, typically existing in the form of data tables. Core data assets are high-value data assets critical for product innovation, management empowerment, and data exchange profitability within enterprises. They are prerequisites for certain enterprise activities, such as formulating data security levels and data exchange values. As enterprise-owned data assets continue to surge, core data assets are often hidden within large volumes of data assets, posing a challenge for enterprises to use them. Data assets originate from business activities, thereby the value of data assets is highly scenario-dependent. Enterprises rely on data analysts' subjective experience for manual identification, resulting in low efficiency and non-generalizable methods.

Data asset valuation is the process of evaluating the value of data assets, which can be used to identify core data assets. Current data asset valuation methods often use market, income, and cost methods, designing data asset valuation models based on industry demands and enterprise characteristics[3]. These methods have two shortcomings. First, market, income, and cost methods are intangible asset valuation techniques[4] and don't fully account for the unique characteristics of data assets. Next, existing data asset valuation methods are limited in scope and can't be applied across different scenarios. There is a lack of a universal method for data asset valuation.

The data within tables frequently undergoes various processes through user-defined data jobs driven by business activities in enterprises. This processing often results in the generation of new data tables and establishes data flows among existing tables. Such data transformations are known as data lineages[5]. A data lineage graph (DLG) is a data model that graphically represents data lineages among data table assets (Section 3.1). As illustrated in Figure 1, DLG can be abstracted as a directed node-link graph, where each node represents a data asset, and each edge signifies a transformation between data assets. The direction of an edge indicates the flow of data transformation. Both the overall data transformation structure and the detailed local data transformations between assets are clearly depicted in a DLG. For example, the data table node T1 stores customer feedback records for a certain enterprise's products. To enhance product innovation and customer retention, data jobs J1 and J2 use different sets of Structured Query Language (SQL) codes to extract feedback information (e.g., product drawbacks, advantages, suggestions, etc.) from different customers from T1, and save it to tables T2 and T3. To ensure information comprehensiveness and authenticity, data analysts retrieve relevant information (e.g., enterprise scale, business area, service level, etc.) from tables T4 and T5, in conjunction with T1 data, to generate T2 and T3.

DLGs provide effective assistance in facilitating data asset valuation. First, DLGs establish a bridge between the data analysts' subjective experiences and the objective topological features of data assets, enabling the study of universal features of data assets. Chen et al.[6] conducted an empirical study in which data analysts were invited to manually identify core data assets using their subjective experiences and then analyzed the topological features of these assets within DLGs. The study found intrinsic connections between core data assets identified from the subjective experiences and the topological structure features (Section 3.2). This supports our research. Second, data assets can be considered as nodes in DLGs. The task of data asset valuation thus can be translated into the problem of ranking nodes in a graph using centrality measures. Many centrality measures have been proposed, such as degree centrality, closeness centrality, and betweenness centrality. These approaches can offer valuable inspiration for our research.

However, few methods exist for data asset valuation via DLGs. The primary reason is that data asset valuation in DLGs requires assessment from both static and dynamic perspectives (Section 3.3). First, data asset valuation has a static nature. Data assets originate from enterprise activities and are abstracted as data table nodes in DLGs. The value of data assets can be assessed by their business and structural characteristics. Most current centrality measures, assessing nodes from a single perspective, cannot capture complex characteristics of data assets in DLGs. Five features of data table assets have been proposed based on these characteristics but lacked a direct method for quantitative valuation. Second, data asset identification has a dynamic nature. The value of data table assets changes with data transformations.

[1]School of Computer Science and Engineering, Central South University, Changsha, China.
*Address correspondence to: Fangfang Zhou, School of Computer Science and Engineering, Central South University, Changsha, China, E-mail: zff@csu.edu.cn

Current centrality measures ignore the dynamic nature of data asset value and cannot accurately identify core data assets in DLGs. Our pilot experiments showed current centrality measures perform unsatisfactorily in data asset valuation.

This work presents the first attempt to investigate relatively universal data asset valuation based on data lineage graphs. We propose a method called graph-based data asset valuation (GBDAV). First, we quantify five features of data assets to calculate their static value. Second, we use a heat diffusion model to simulate data transformation, defining a dynamic value for each table based on heat input and output. Finally, we construct two heat transfer matrices and iteratively update dynamic values until thermal equilibrium is reached. A series of experiments are conducted to evaluate the effectiveness of the proposed GBDAV. The results reveal that GBDAV performs the best among the six reference centrality measures in identifying data assets with high value and nodes with high structural impact in DLGs. GBDAV also achieves highly satisfactory performance distinguishing data assets' ability in DLGs.

2. Related Work

2.1 Data Asset Valuation

Data assets are similar to intangible assets as they lack a tangible form[7]. Most scholars use methods for valuing intangible assets to evaluate the value of data assets: market-based, cost-based, and income-based methods[8-10]. Market-based valuation[11] is conducted by seeking comparable assets in a public trading market, comparing them from both similarities and differences and ultimately determining the value of the asset. This method can effectively demonstrate the current value of the asset. Due to the lack of mature data asset trading markets, it is challenging to obtain historical transaction cases, which makes it difficult to implement market-based valuation. Cost-based valuation[12] evaluates data assets based on their purchase price or development cost, focusing on cost analysis. The calculation process is relatively straightforward. Since data assets are generated in various business activity stages, accurately obtaining cost data is challenging. Income-based valuation[13] determines value by predicting the economic benefits an asset will generate and applying an appropriate rate to those benefits. Compared to market and cost approaches, the scope of income-based valuation is broader and emphasizes a data asset's ability to create future benefits. This is the closest approximation of a data asset's economic value. However, estimating future economic benefits is challenging, and converting them to monetary equivalents is difficult.

Based on the characteristics of data assets, scholars have introduced various adjustment factors to enhance the adaptability of traditional intangible asset valuation methods[14-16]. With numerous factors influencing data asset value, scholars propose using the Analytic Hierarchy Process (AHP) to determine each factor's weight and establish a value assessment model[17,18]. Moreover, machine learning methods, such as classification, clustering and regression, are used to calculate data asset value[19-22]. Artificial neural networks objectively evaluate and predict data application value due to their high self-organization and adaptability[23,24]. This overcomes human factors and subjective evaluation fuzziness. However, all the aforementioned methods for valuing data assets often assess the overall data assets owned by an enterprise, making it difficult to identify specific important data assets at a micro level. Furthermore, these methods are designed to assess data assets in specific enterprises or scenarios and may not apply to different scenarios or enterprises.

2.2 Graph Centrality Measure

The current centrality measures can be classified into three groups, namely, node neighbors-, path- and feature vectors-based[25,26]. In the node neighbors-based group, Degree Centrality (DC) and K-Shell Decomposition (KSD) are typical node neighbors-based algorithms[27,28]. These algorithms define the importance of nodes based on their neighboring node[29]. DC measures the direct influence of a node, it believes that the larger the degree of a node, the more neighbors it can directly influence, and therefore the more important it is[30]. Its drawback is that it only considers the most local information of the node, and does not delve deeper into the environment around the node, leading to inaccuracies in many cases. KSD not only considers the information of the node's neighbors but also the node's location information. It believes that if a node is located in the core of the network, even if its degree is small, it often has a high influence[31]. The number of layers divided by the k-shell decomposition method is significantly fewer than the number of layers divided by the degree centrality method, making it difficult to compare the importance of nodes at different levels.

Closeness Centrality (CC) and Betweenness Centrality (BC) are typical path-based algorithms[32,33]. They define the importance of nodes based on their control over network information flow[34]. CC measures a node's importance in a network based on its average distance to other nodes. The shorter the average distance, the more important the node. The algorithm is widely applied in studies, but has high time complexity[35]. BC evaluates a node's importance by the number of shortest paths passing through it. It characterizes a node's control over network traffic along the shortest paths. Traffic Load Centrality (TLC) is a variant of BC that utilize a mechanism similar to information packet transmission in a network[36]. These algorithms have similar computational complexity limitations[37,38].

The feature vectors-based group includes many algorithms. PageRank and LeaderRank algorithms consider that the importance of a page on the World Wide Web depends on the quantity and quality of other pages pointing to it[39,40]. PageRank algorithm assumes that if a page is linked to by many high-quality pages, then its quality is also high. As the most classic ranking algorithm for directed network, PageRank and its improved algorithms are widely applied in various fields. However, the random "hopping" probability for each node is the same in PageRank, that is, the probability of visiting other pages from any web page using the input URL is equal[41,42].

LeaderRank algorithm replaces the random hopping probability c in the PageRank algorithm with the probability of visiting other pages from the background node using the input URL, thereby obtaining an algorithm with no parameters and a simpler form[43,44]. Different types of nodes in a network have varying functions, and the importance of each node cannot be accurately determined by a single metric. HITS[45] algorithms assign two metrics to each node: authorities and hubs. Authorities measure the originality of the node's information, while hubs reflect the node's role in information propagation. HITS algorithms were the first to simultaneously rank nodes in a network using different metrics, which holds significant implications.

It should be noted that special network structures can influence the performance of algorithms such as LeaderRank and HITs, which utilize the neighbor's scores to rank nodes[46]. For instance, tightly connected communities have nodes with extremely close inter-node links, which can mutually reinforce the authorities and hubs of these nodes, thereby pushing the ranking results to favor nodes in communities over the correct ones. The data lineages within a data lineage graph is quite complex, forming a tightly connected local structure between data table nodes. This is also why these algorithms are unable to accurately identify core data assets on DLGs.

3. Overview

3.1 Data Description

Each DLG has three data asset types (i.e., data table, data field, and data job) and two relation types (i.e., PARENT_CHILD and DATA_FLOW), which can be used to describe extract, transform, and load (ETL) processes in application scenarios.

As described in Table 1, data tables are the primary form of data assets, storing business-related data within an enterprise. This work focuses on the valuation of data table assets. Data fields represent the fundamental units for storing data within data tables. These two types of data assets form PARENT-CHILD relations, indicating the affiliations between data tables and data fields. In Figure 1, the data table asset T1 has many PARENT_CHILD relations, indicating that these data field assets are constituent parts of T1. Data jobs typically involve SQL codes to manipulate data from data tables, establishing DATA_FLOW relations among them. These relations represent data transformation directions between data tables and data jobs. In Figure 1, a DATA_FLOW relation from the data table asset T1 to the data job asset J1 signifies that the information in T1 is extracted and manipulated by J1. Likewise, a DATA_FLOW relation from the data job asset J1 to the data table asset T2 implies that the processed information is loaded into T2 by J1.

3.2 Concept Definition

Five universal features of core data assets in DLGs are derived from the combined business and structural characteristics of data assets.

(1) Data richness

Data experts noted that the number of data fields would influences the business value of data assets. If the number is small, a data asset has difficulty playing a significant role in the data transformation process.

(2) Transformation influence

This universal feature refers to the extent of disruption to the data transformation process when a data asset is removed. There are multiple data lineages within a DLG, connected through specific data tables. Disrupting these tables can cause the entire data transformation process to break down.

(3) Data transfer volume

This universal feature refers to the total amount of data that is transferred from the data asset. The greater the amount of data that the data asset transfers through data transformations, the greater its value. This reflects its fundamental role in providing data.

(4) The number of data sources

This universal feature refers to the number of data assets directly transmitting data to the target data asset. Data transformations between data assets is driven by business activities, and accepting data from a number of data assets implies that the data asset has significant value.

(5) The value of data sources

This universal feature refers to the value of data assets that directly transmit data to the target data asset. If a data asset requires invoking other high-value data assets, it also likely has high value.

3.3 Design Consideration

Data asset valuation in DLGs should be assessed from static and dynamic perspectives.

(1) The value of data assets has a static nature

Data assets arise from enterprise business activities and possess business value. Data assets in DLGs are abstracted as data table nodes with structural value. It suggests that data assets in DLGs can be valued from both a business and a structural perspective. A data lineage graph (DLG) is denotated with $G = (V, E)$, where V represents the nodes and E represents the edges. We use $\{DR, TI, DTV, NS, VS\}$ to represent the five features of a data table node $v_T \in V$ in G, where DC represents the data richness of v_T, TI represents the transformation influence of v_T, DTV represents the data transfer volume experienced by v_T, NS represents the number of data sources that directly transfer data to v_T, and VS represents the value of data sources that directly transfer data to v_T. By quantifying the $\{DR, TI, DTV, NS, VS\}$, we can comprehensively assess the value of data assets at business and structural levels.

(2) The value of data table assets has a dynamic nature

The value of data assets changes with data transformations. Data transformations refer to the process in which data of data tables are used by data jobs to generate new data tables. They are performed based on specific business requirements, including complex data input and output. The more data input and output a data asset has during the transformation process, the greater its dynamic value, indicating its increasing importance within that particular business requirement.

(3) Data transformations are dynamically iterative

DLGs are composed of data tables and data transformations among them, forming a static graph model. But DLGs only indicate the direction of data transformation and do not provide details about the transformation process or its outcomes. Therefore, by repeatedly executing the data transformations within DLGs, it is possible to iteratively track the data transformation process and its outcomes for data table nodes until the data transformation results stabilize and no longer undergo significant changes.

4. Algorithm Design

We introduce a new algorithm called GBDAV for data asset valuation on DLGs. The proposed GBDAV algorithm consists of three steps, as shown below:

4.1 Static Value Definition

STEP 1 is to define and calculate the static value of data assets in G. We propose a method to assess the static value of data assets, as the five proposed features cannot be directly quantified to determine this value. This step outputs the static value of data table nodes in G. For each data table node $v_T \in V$, we reference five features of core data assets $\{DR, TI, NT, NS, VS\}$, namely, data capacity, structural influence, number of transformations, number of sources, and value of sources, to propose five quantitative methods as follows:

(1) We use the number of data field $v_F \in V$ connected to v_T to quantify the $data\,richness$, noted as:

$$DR = |v_F|$$

(2) We use the shortest path[47] after deleting v_T to quantify the $transformation\,influence$, noted as:

$$TI = \sum_{(i,j)\in U} \frac{1}{d(i,j)},$$

where the set U denotes the collection of nodes in G that become disconnected after deleting node v_T. $d(i,j)$ represents the shortest distance between nodes i and j before deleting node v_T.

(3) We use the information indices[48] to measure the total amount of information transmitted by v_T throughout the data transformation process, noted as:

$$DTV = \left[\frac{1}{n}\sum_j \frac{1}{r_{ij}}\right]^{-1},$$

$$R = (r_{ij}) = (D - A + F)^{-1}$$

where D is an n-order diagonal matrix, the diagonal elements are the degree values of the corresponding nodes; A is the adjacency matrix of the graph G and F is a n-order square matrix in which all elements are equal to 1. The DTV metric of node v_T is defined using the harmonic mean method[49].

(4) We use the number of data tables that directly transfer data to v_T to quantify the $number\ of\ sources$, noted as:

$$NS = |v'_T|,$$

where v'_T are the data tables that directly transfer data to v_T.

(5) We use the static value of data tables that directly transfer data to v_T to quantify the $value\ of\ sources$, noted as:

$$VS = \sum SV(v'_T),$$

where $SV(v'_T)$ is the static value of data tables that directly transfer data to v_T.

There are two types of data table nodes in DLGs: source nodes and non-source nodes. Source nodes only have outgoing edges to data jobs and transmit data outward. Their static value considers only three features: DR, TI, and DTV. Non-source nodes must be assessed using all five features for their static value. The static value of data table node v_T in G is as follows:

$$SV(v_T) = \begin{cases} \frac{DR}{\sum DR} + \frac{TI}{\sum TI} + \frac{DTV}{\sum DTV}, if\ v_T \text{ is source node} \\ \frac{DR}{\sum DR} + \frac{TI}{\sum TI} + \frac{DTV}{\sum DTV} + \frac{NS}{\sum NS} + VS, otherwise \end{cases}$$

4.2 Dynamic value definition

STEP 2 is to define the dynamic value of data assets in data transformation. The dynamic nature of data assets cause changes to their static value as data transformations occur in DLGs. Data transformation involve both input and output, like heat input and output in heat diffusion[50]. We apply an iterative refinement process analogous to heat diffusion on DLGs to propagate the dynamic value over data table assets. Data tables can be analogized as heat points, data as heat, and data transformation as heat diffusion.

In the heat diffusion, the greater the heat input and output of a heat point, the more important it is within the entire system. We utilize the dynamic heat status of heat points to assess their dynamic changes in value. The dynamic heat status of heat point i at time t can be represented as:

$$H^t(i) = H^t_{in}(i) + H^t_{out}(i),$$

where $H^t_{in}(i)$ represents the heat input of heat point i and H^t_{out} (i) represents the heat output of heat point i at time t.

When $t = 0$, there is no heat diffusion between heat points, that is, $H^0_{in}(i) = H^0_{out}(i) = 0$. At $t = 1$, heat propagates directionally along edges between heat points, following strict heat diffusion paths corresponding to data lineages in DLGs. We define heat output $H^1_{out}(i)$ and heat input $H^1_{in}(i)$ as influenced by two factors: the point's heat (i.e., the static value) and heat propagate along connecting edges. Therefore, we trace forward heat diffusion paths of any point i, recording points in paths as j; then, we trace backward, recording points in paths as k. The heat output and heat input of any heat point i at $t = 1$ is denoted as:

$$\begin{cases} H^1_{out}(i) = \sum(SV(j) * \prod HE(i,j)\,) \\ H^1_{in}(i) = \sum(SV(k) * \prod HE(i,k)\,) \end{cases},$$

where SV represents the heat of heat point and HE represents the heat transferred along the edge. Heat is uniformly propagated within the system, the heat propagated along the edges can be calculated based on the number of edges. For example, if a heat point extends outwards with 2 edges, then the heat of each edge is 1/2.

As shown in Figure 2, the heat point T1 has two heat diffusion paths, that is, $T1 \rightarrow J1 \rightarrow T2$ and $T1 \rightarrow J2 \rightarrow T3$. We trace forward two heat diffusion paths of the heat point $T1$, the heat output of T 1 at $t = 1$ can be represented as: $H^1_{out}(T1) = SV(T2) *$

$Edge_{T1\to J1} * Edge_{J1\to T2} + SV(T3) * Edge_{T1\to J2} * Edge_{J2\to T3}$. The heat point T2 has one heat diffusion path, that is, $T1\to J1\to T2$. We trace backward the heat diffusion path of $T2$, the heat input of $T2$ at $t = 1$ can be represented as: $H_{in}^{1}(T2) = SV(T1) * Edge_{T1\to J1} * Edge_{J1\to T2}$.

4.3 Dynamic value iteration

STEP 3 is to literately update the dynamic value of data assets in G. Heat transformations between heat points are dynamically iterative. We construct two heat transfer matrices, named TM_1 and TM_2, representing the heat input and heat output state among heat points. These two matrices can be used to update the heat input and output of the heat points.

$$\begin{cases} TM_1 = D_r^{-1}W \\ TM_2 = D_c^{-1}W^T \end{cases},$$

where W is the adjacency matrix of G. If node i points to node j, then the element in the i^{th} row and j^{th} column of W is 1. D_r is the diagonal matrix of W, where the element in the i^{th} row and i^{th} column equals the total sum of elements in the i^{th} row of W, reflecting the out-degree of each node in G. D_c is the diagonal matrix of W, where the element in the i^{th} row and i^{th} column equals the total sum of elements in the i^{th} column of W, reflecting the in-degree of each node in G. D_r^{-1} and D_c^{-1} are the inverse matrices of the diagonal matrices $D_r$and D_c, with diagonal elements being the reciprocal of the original elements.

To facilitate iteration of $H_{in}^{t}(i)$ and $H_{out}^{t}(i)$, we calculate heat input and output for all heat points in G at $t = 1$, forming two $n * 1$ matrices, denoted as:

$$IM^1 = \begin{bmatrix} H_{in}^1(1) \\ H_{in}^1(2) \\ \cdots \\ H_{in}^1(n) \end{bmatrix}, OM^1 = \begin{bmatrix} H_{out}^1(1) \\ H_{out}^1(2) \\ \cdots \\ H_{out}^1(n) \end{bmatrix},$$

where n is the number of heat points in G. Each heat point updates its heat state based on other connected points' heat states until system thermal equilibrium is reached. At time t, any heat point's heat input and output matrices are:

$$\begin{cases} OM^t = \alpha * TM_1 * IM^{t-1} + (1-\alpha) * OM^1 \\ IM^t = \beta * TM_2 * OM^t + (1-\beta) * IM^1 \end{cases},$$

where α and β are the weights which range from 0 to 1. The greater value of α and β is chosen to show more confidence on the initial heat value for heat points. After experimentation, we set the two parameter values as follows: α=0.77 and β=0.23.

We show the sequences OM^t and IM^t in the iterative refinement process converge. Considering the sequence OM^t firstly, by the iteration in the above equation, we have

$$OM^t = \alpha(1-\beta)TM_1IM^1 + (1-\alpha)OM^1\sum_{k=0}^{t-1}(\alpha\beta TM_1TM_2)^k + (\alpha\beta TM_1TM_2)^tOM^1$$

Since $0 < \alpha,\ \beta < 1$, and the eigenvalues of TM_1TM_2 are in $[-1, 1]$, we have

$$\begin{cases} \lim_{t\to\infty}(\alpha\beta TM_1TM_2)^t = 0 \\ \lim_{t\to\infty}\sum_{k=0}^{t-1}(\alpha\beta TM_1TM_2)^k = (I-\alpha\beta TM_1TM_2)^{-1} \end{cases}$$

Hence, we obtain the heat input matrix at time t:

$$OM^* = \lim_{t\to\infty} OM^t = (I-\alpha\beta TM_1TM_2)^{-1}[\alpha(1-\beta)TM_1IM^1 + (1-\alpha)OM^1]$$

Similarly, we can get

$$IM^* = \lim_{t\to\infty} IM^t = (I-\alpha\beta TM_2TM_1)^{-1}[\beta(1-\alpha)TM_2OM^1 + (1-\beta)IM^1]$$

Consequently, by adding the elements in OM^* and IM^* row-wise, we obtain n values corresponding to the importance estimation of data table nodes in G.

5. Evaluation

We evaluated the proposed GBDAV algorithm through an objective performance analysis and a field assessment.

5.1 Objective Performance Analysis

The performance analysis had three experiments with different perspective. (1) Ability to distinguish nodes[51]. A high frequency of nodes with the same centrality value indicates poor performance for a centrality measure. (2) Ability to identify nodes with high structural impact[52]. This centers on studying structural changes in a network after removing a node subset. The greater the change, the more important the removed nodes are considered. (3) Ability to identify core data assets. The proposed GBDAV algorithm is designed to help data analysts improve efficiency in identifying core data assets. The key to evaluating its performance is whether it can accurately identify core data assets in DLGs.

We selected the proposed GBDAV and six reference algorithms, namely, Closeness, Betweenness, Eigenvector, LeaderRank, HITs and H_index, most of which performed relativelywell in the previous pilot experiments. The experimental dataset is an open dataset of Data Lineage Graphs for Data Governance Research (DLG-DG-23)[53]. This dataset is a real-world dataset sourced from Huawei Cloud Computing Technology Company Limited (referred to as "Huawei Cloud"), which is the fifth-largest global cloud service provider. The raw data in the dataset have been subjected to a set of data cleansing and rigorous anonymization processes to ensure high quality and eliminate privacy and security risks. As shown in Table 2, the dataset contains 18 DLGs with diverse sizes from 278 nodes to 17,085 nodes. These DLGs were sourced from three application scenarios, namely, cloud infrastructure, customer service, and operation analysis. These application scenarios frequently appear in enterprise management. Anonymization operations were performed on the names of data assets and their relations to prevent the disclosure of private information.

5.1.1 Ability to Distinguish Nodes

In the process of determining the significance or ranking nodes in a network, it's not uncommon to come across instances where certain nodes exhibit identical centrality values. This occurrence can be attributed to the inherent nature of the network or the methodology employed to assess the centrality values. In such scenarios, these nodes appear virtually indistinguishable from one another, posing a challenge in discerning their relative significance within the network. The prevalence of nodes manifesting identical centrality values serves as an indicator of the overall efficacy of the centrality measure adopted. A noticeably high frequency of nodes displaying the same centrality value could potentially signal subpar performance from the centrality measure, suggesting that it might not be adept at capturing subtle nuances or distinguishing features among nodes. This can be used as an indicator to evaluate the performance of a centrality measure, denoted as:

$$Repetition\ Frequency = \frac{n'}{n},$$

where n' denotes the total number of nodes with the same centrality value and n denotes the total number of data table nodes in DLGs. The minimum value $\text{Repetition Frequency} = 0$ represents all data table nodes assigned different centrality values, while the maximum value $\text{Repetition Frequency} = 1$ indicates all data table nodes have the same centrality value.

The Repetition Frequency of GBDAV and HITs are the smallest, while the Repetition Frequency of H_index and Betweenness are almost the greatest. LeaderRank and Eigenvector exhibit comparable performances; their Repetition Frequency is almost indistinguishable. Specifically, the GBDAV demonstrates a lower Repetition Frequency than the HITs algorithm for small- and Medium-scale DLGs, suggestive of its superior ability to discern nodes' varying competence. As clearly demonstrated in Figure 3, the horizontal axis represents the ranking order of nodes, while the vertical axis represents the Repetition Frequency. The difference in the ability of an algorithm to differentiate between nodes is reflected in the Repetition Frequency of the same sorted result. The stronger the algorithm's ability to distinguish between nodes, the smaller the Repetition Frequency. H_index and Betweenness exhibit a higher frequency than other centrality measures, indicating that a significant number of nodes share the same centrality value when using these two methods. It's evident that they possess a notably diminished capacity to adequately distinguish individual nodes. In comparison, GBDAV and HITs have the least number of nodes with the same centrality value across all cases that were studied. Unlike other centrality measures, these two better detect differences between nodes and can more accurately distinguish between them.

5.1.2 Ability to identify structural impact nodes

In network analysis, a critical measure of a node's significance is its connectivity with other nodes. This can be quantified by calculating the change in overall network connectivity when the node is removed. Initial studies used various centrality measures, each assessing a node's importance differently. The concept of the giant component is particularly relevant in directed networks. The giant component can be divided into two distinct segments: the strongly connected component (SCC) and the weakly connected component (WCC). In this evaluation, we consider the "giant component" as weakly connected components primarily due to the following two considerations. First, from the point of view of graph theory, a strongly connected component refers to a subgraph where any two nodes have a bidirectional path. However, our data lineage graph is a clear unidirectional graph without bidirectional paths. Secondly, the term "giant component" is often used in many studies, particularly in the research of directed networks, to denote weakly connected components. Its meaning is to represent a subgraph where there exists at least one directed path between any two nodes (directly or indirectly). This definition aligns more closely with our research object and its characteristics, thus we choose to consider the "giant component" as weakly connected components.

The evaluation experiment primarily comprises three key steps. First, we start by obtaining the maximum weakly connected component G' from the weakly connected components in the graph G; next, for each DLG, we calculate the node ranking results using the algorithms we proposed and six comparison algorithms; finally, for any algorithm's node ranking results on different DLGs, we remove the $top - N\%$ nodes from the graph G each time, and then recalculate the size of the maximum weakly connected component G' in the graph G. We initially set N to 5, with a maximum value of 30, increasing by 5 each time. This is because our research found that when N reaches 30%, the size of the maximum connected component G'already shows a tendency to stabilize. When N exceeds a threshold, all remaining nodes become island nodes.

Figure 4 clearly illustrates that the size of the largest weakly connected component, playing a significant role in network analysis, initially declines before reaching a stable state as N increases. This can be attributed to the fragmentation of the largest component into multiple smaller ones as a certain percentage of nodes are removed from the network. However, as N continues to increase, the size of the largest component stabilizes. The red curve representing the GBDAV algorithm notably demonstrates a faster convergence rate. Particularly when the value of N reaches 10%, it has already shown signs of approaching stability. This phenomenon implies that the GBDAV algorithm performs better than other centrality algorithms in accurately identifying nodes that exert a significant impact on graph structure. In addition, in two large-scale DLGs, namely DLG-6 and DLG-18, the performance of Betweenness is comparable to that of the GBDAV algorithm. This is because the Betweenness evaluates the importance of a node by counting the number of shortest paths that pass through it. Essentially, this means that Betweenness can effectively identify nodes that play a key role in graph structure.

5.1.3 Ability to identify core data assets

Identifying core data assets within the DLG is a key objective in the design of this algorithm. Due to the complex nature of data assets, the identification of core data assets is typically implemented manually by data experts within enterprises based on their own experience. To validate the ability of this algorithm to identify core data assets, we invited three data experts from Huawei Cloud to manually label the core data assets on DLGs in the DLG-DG-23 dataset. Manual annotation is time-consuming and labor-intensive, we only annotated small- and medium-scale DLGs. These annotated information will serve as the ground truth for this evaluation.

The Receiver Operating Characteristic (ROC) curve is used to assess the algorithm's ability to identify core data assets. The ROC curve shows the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR) at different thresholds, assessing the algorithm's performance on actual positive and negative samples, noted as

$$TPR = \frac{TP}{TP+FN}, FPR = \frac{FP}{FP+TN},$$

where TP is the number of correctly identified core data assets, FP is the number of non-core data assets identified as core data assets, FN is the number of core data assets identified as non-core data assets, and TN is the number of correctly identified non-core data assets.

The process is as follows: First, we set an initial threshold p, with a value of 10%, and calculate the average true rate (TPR) and false positive rate (FPR) of each algorithm on all DLGs. Then, the threshold p will be adjusted according to a 10% increment each time, until reaching its maximum value of 90%. Through this process, each algorithm will ultimately generate a curve composed of 9 key points. The horizontal axis represents the algorithm's average false positive rate (FPR) on all DLGs, while the vertical axis represents the algorithm's average true rate (TPR). As shown in Figure 5, the red curve representing GBDAV algorithm is consistently leading, surpassing all other curves, indicating a significant performance superiority over other algorithms. This enables it to accurately identify core data assets from DLGs with greater precision. The Area Under the Curve (AUC) is a numerical value representing the area beneath the ROC curve, used to quantify an algorithm's overall performance. Its values range from 0 to 1, with higher values indicating better performance. The AUC value of GBDAV is significantly higher than other centrality measures, indicating it's better at identifying core data asset nodes in DLGs. The AUC values for Closeness and Betweenness are virtually identical, which is in consonance with their definitions. For centrality measures with an AUC value below 0.5, its performance is worse than random guessing.

Moreover, Infectious disease models, such as the SIS and SIR models, are commonly used to assess a node's information dissemination capability. However, information propagation in networks significantly differs from virus transmission[54]. Data transformation between core data asset nodes in DLGs differs from virus spread, making infectious disease models inappropriate for assessing their information dissemination.

5.2 Field Assessment

In order to demonstrate that GBDAV algorithm can effectively assist data users in improving the identification efficiency of core data assets in practical application scenarios, we collaborated with Huawei Cloud and conducted a month-long field trial with their data users. The data users in the daily workflow typically search for data assets using keywords and verify whether the data assets conform to the expected objectives by business descriptions. Then, they evaluate the value of the data in these tables using the business knowledge they possess. Moreover, they examine the data conversion process and determine its importance in the business chain. Finally, they review other data tables in the conversion process and repeat the first two steps, continuing until most core data assets are identified. We encouraged three users to use our method to identify core data assets in their daily work. The commonly used data asset manual identification method by Huawei Cloud was used as a comparison. As shown in Table 3, we conducted interviews with the users to evaluate usability, effectiveness, and user satisfaction. They were asked to rate each question on a five-point Likert scale, ranging from 5 (strongly agree) to 1 (strongly disagree).

In summary, the proposed GBDAV algorithm consistently scored higher (3.5 vs 3.1) than the reference method across all questions during the field study, indicating it outperformed the reference method.

For Question 1, GBDAV algorithm has a lower score than the comparison method. Due to Huawei Cloud's current use of a directory-style display for data assets and their transformation relationships, there is no promotion of data lineage graphs. This necessitates additional graph abstraction processing for data asset transformation relationships, making the process slightly cumbersome. For Question 2, most data users stated that the reference method typically requires extensive training to identify core data assets while GBDAV algorithm is an automated algorithm that can be quickly grasped.

In terms of effectiveness (Quesitons 3-4), GBDAV can obtain the ranking results of potential core data assets with only the input of data lineage graphs. This reduces the screening time for data users and significantly enhances their efficiency in identifying core data assets. Additionally, data users rely on business scenario information for manual identification of core data assets, but efficiency and accuracy suffer in unfamiliar scenarios In contrast, GBDAV achieves cross-scenario identification of core data assets.

In terms of effectiveness (Quesitons 5-6), GBDAV received slightly higher average ratings than the reference method. However, not all data users are fully satisfied with either method. They noted that GBDAV is convenient, but lacks identification process and detailed contextual information. After the algorithm provides potential core data asset ranking results, data users still need to verify their authenticity. One data user suggested that if the identification process and contextual information of potential core data assets could be visualized, it would greatly enhance the algorithm's credibility.

6. Discussion

We mainly used data lineage graphs in this work derived from three business scenarios of Huawei Cloud. Whether our GBDAV applies to data lineage graphs from other scenarios or enterprises, needs to be deliberated.

In our algorithm design, we assumed the five core data asset features had equal weights. In practice, feature weights vary. Each feature's weight can be adjusted as needed. Similarly, we assumed data was transmitted uniformly along edges, but in reality, the value of edge transmission varies. We should collect more real data to assign appropriate weights to the edges.

In our subjective assessment experiment, we evaluated algorithms using core data assets annotated by three data experts from Huawei Cloud. However, expert annotations can be erroneous or lacking, so involving more experts in the data annotation process is advisable. While we tested our algorithm on large-scale DLGs in the first two analyses, but lack of expert-annotated data prevented us from assessing our algorithm's ability to identify core data assets in such large-scale DLGs. We should consider if the proposed GBDAV algorithm applies to large-scale data lineage graphs.

In future work, we plan to properly modify GBDAV to extend its applied scope from the DLG-DG-23 dataset to data lineage graphs in other scenarios or enterprises. We plan to add new data asset features into the static value definition process. Moreover, a layout that can present data assets distinctly is worth further exploration.

7. Conclusion

The identification of core data assets is an essential and open issue, which is of great significance to the data asset valuation and data governance of enterprises. In this paper, a novel method, called GBDAV, considering core data assets as important nodes in data lineage graphs is proposed for core data asset identification in data lineage graphs. GBDAV comprehensively considers three aspects of data table assets in data lineage graphs, including the static value, the dynamic value, and the iteration of the dynamic value. Here, we use five core data asset features to measure the static value of data table assets, considering both business and structural characteristics; then, we analogize the data transformation process using a heat diffusion model, measuring the dynamic value from aspects of heat input and heat output; finally, we iteratively update the heat input and output of the data table assets through heat transfer relationships (data lineages) until the system reaches thermal equilibrium.

This work is the first investigation of data asset valuation in data lineage graphs. We hope this work will be conducive to the research and application of graph analysis oriented to data asset valuation. We also expect that this work will inspire other researchers to further study data asset valuation.

Authors' Contributions

Investigation and Writing-Original Draft Preparation, Yunpeng Chen; Writing, Reviewing and Editing, Ying Zhao; Scrub Data and Maintain Research Data, Xuanjing Li, Chenrui Ma; Resources-Provision of Study Data, Xin Zhao; Conceptualization and Supervision, Ying Zhao and Fangfang Zhou. All authors have read and agreed to the published version of the manuscript.

Availability of Data and Materials

The data underlying this article can be found at: https://github.com/csuvis/DataAssetGraphData

Author Disclosure Statement

All authors declare that they have no potential conflicts of interest with respectto the research, authorship, and/or publication of this article.

Funding Information

This research received no external funding.

Figure Legends

Figure 1. Example of a DLG that contains 105 data assets and 104 relations.

Figure 2. Example of heat propagation in heat diffusion paths.

Figure 3. The frequency of nodes with the same ranking value in six DLGs of the dataset. The horizontal axis represents the ordering of nodes, while the vertical axis represents the number of nodes with the same ordering.

Figure 4. Variation on the size of the largest weakly connected component when removing different proportions of nodes in six DLGs of the dataset.

Figure 5. The average ROC curves of seven algorithms on small and medium-scale DLGs of the dataset.

References

[1] Birch K, Cochrane DT, Ward C. Data as asset? The measurement, governance, and valuation of digital personal data by Big Tech. Big Data & Society 2021; 8(1); doi: 10.1177/20539517211017308

[2] Telenti A, Jiang X. Treating Medical Data as A Durable Asset. Nature Genetics 2020; 52:1005-1010; doi: 10.1038/s41588-020-0698-y

[3] Kim O, Park J, Cho WS, et al. Data Asset Valuation Model Review. The Korea Journal of BigData 2021; 6(1):153-160; doi: 10.36498/KBIGDT.2021.6.1.153

[4] Van Criekingen K, Bloch CW, Eklund C. Measuring intangible assets-A Review of the State of the Art. Journal of Economic Surveys 2021; 36(5):1539-1558; doi: 10.1111/joes.12475

[5] Tang M, Shao S, Yang W, et al. SAC: A System for Big Data Lineage Tracking IEEE 35th International Conference on Data Engineering (ICDE) 2019; 1964-1967. doi: 10.1109/ICDE.2019.00215

[6] Chen Y, Zhao Y, Xie W, et al. An Empirical Study on Core Data Asset Identification in Data Governance. Big Data and Cognitive Computing 2023; 7(4):161; doi: 10.3390/bdcc7040161

[7] Xiong F, Xie M, Zhao L, et al. Recognition and Evaluation of Data as Intangible Assets. Sage Open 2022; 12(2); doi: 10.1177/21582440221094600

[8] Fleckenstein M, Obaidi A, Tryfona N. A Review of Data Valuation Approaches and Building and Scoring a Data Valuation Model. Harvard Data Science Review 2023; 5(1); doi: 10.1162/99608f92.c18db966

[9] Li Z, Ni Y, Gao X, et al. Value Evaluation of Data Assets: Progress and Enlightenment. IEEE 4th International Conference on Big Data Analytics (ICBDA) 2019; 88-93; doi: 10.1109/ICBDA.2019.8713240

[10] Vasarhelyi M.A, Kogan A, Tuttle B.M; Big Data in Accounting: An Overview. Accounting Horizons 2015; 29(2):381–396; doi: 10.2308/acch-51071

[11] Banker R.D, Huang R, Natarajan R, et al. Market Valuation of Intangible Asset: Evidence on SG&A Expenditure. The Accounting Review 2019; 94(6); doi: 10.2308/accr-52468

[12] Pastor D, Glova J, Lipták F, et al. Intangibles and Methods for Their Valuation in Financial Terms: Literature Review. Intangible Capital 2017; 13(2):387-410; doi: 10.3926/ic.752

[13] Lopes I.T. The Boundaries of Intellectual Property Valuation: Cost, Market, Income Based Approaches and Innovation Turnover. Intellectual Economics 2011; 5(1):99-116. Avaliable from: http://hdl.handle.net/10400.15/442. [Last accessed: June 28, 2024].

[14] Rodov I, Leliaert P. FiMIAM: Financial Method of Intangible Assets Measurement. Journal of Intellectual Capital 2002; 3(3):323-336; doi: 10.1108/14691930210435642

[15] Wirtz H. Valuation of Intellectual Property: A Review of Approaches and Methods. International Journal of Biometrics 2012; 7(9); doi: 10.5539/ijbm.v7n9p40

[16] Gu F, Lev B. Intangible Assets: Measurement, Drivers, and Usefulness. IGI Global 2011; 110-124; doi: 10.4018/978-1-60960-071-6.ch007

[17] Lin Z, Wu Y. Research on the Method of Evaluating the Value of Data Assets. International Conference on Education, E-learning and Management Technology 2016; doi: 10.2991/iceemt-16.2016.95

[18] Fu J, Xiao B, Wang F. Evaluation of Data Asset Value Based on Hierarchical Analysis Method. Frontiers in Business Economics and Management 2024; 12(3):240-244; doi: 10.54097/920f9215

[19] Teng H.W, Li Y.H, Chang S.W. Machine Learning in Empirical Asset Pricing Models. International Conference on Pervasive Artificial Intelligence (ICPAI) 2020; 123-129; doi: 10.1109/ICPAI51961.2020.00030

[20] Tsai C-F, Lu Y-H, Hung Y-C, et al. Intangible Assets Evaluation: The Machine Learning Perspective. Neurocomputing 2016; 175(A):110-120; doi: 10.1016/j.neucom.2015.10.041

[21] Gu S, Kelly B, Xiu D. Empirical Asset Pricing Via Machine Learning.The Review of Financial Studies 2020; 33(5): 2223–2273; doi: 10.1093/rfs/hhaa009

[22] Bagnara M. Asset Pricing and Machine Learning: A Critical Review. Journal of Economic Surveys 2024; 38:27–56; doi: 10.1111/joes.12532

[23] Abidoye R.B, Chan A.P.C. Artificial Neural Network in Property Valuation: Application Framework and Research Trend. Property Management 2017; 35(5):554-571; doi: 10.1108/PM-06-2016-0027

[24] Salvador B, Oosterlee C.W, Van der Meer R. Financial Option Valuation By Unsupervised Learning with Artificial Neural Networks. Mathematics 2020; 9(1): 1-20; doi: 10.48550/arXiv.2005.12059

[25] Liu J, Li X, Dong J. A Survey on Network Node Ranking Algorithms: Representative Methods, Extensions, and Applications. Science China Technological Sciences 2021; 64:451-461; doi: 10.1007/s11431-020-1683-2

[26] Wu S, Yin H, Cao H, et al. Node Ranking Strategy in Virtual Network Embedding: An Overview. China Communications 2021; 18(6):114-136; doi: 10.23919/JCC.2021.06.010

[27] Zhang J, Luo Y. Degree Centrality, Betweenness Centrality, and Closeness Centrality in Social Network. International Conference on Modelling, Simulation and Applied Mathematics 2017; 300-303; doi: 10.2991/msam-17.2017.68

[28] Peng G, Qiguang M, Steffen S. Community-based K-shell Decomposition for Identifying Influential Spreaders. Pattern Recognition 2021; 120:108130; doi: 10.1016/j.patcog.2021.108130

[29] Freeman L.C. Centrality in Social Networks Conceptual Clarification. Social Networks 1978; 1:215-239; doi: 10.1016/0378-8733(78)90021-7

[30] Butts C. T. Social Network Analysis: A Methodological Introduction. Asian Journal of Social Psychology 2008; 11(1):13–41; doi: 10.1111/j.1467-839X.2007.00241.x

[31] Garas A, Schweitzer F, Havlin S. A K-shell Decomposition Method for Weighted Networks. New Journal of Physics 2012; 14(8); doi: 10.1088/1367-2630/14/8/083030

[32] Okamoto K, Chen W, Li X. Ranking of Closeness Centrality for Large-Scale Social Networks. Frontiers in Algorithmics 2008; 186-195; doi: 10.1007/978-3-540-69311-6_21

[33] Barthélemy M. Betweenness Centrality in Large Complex Networks. The European Physical Journal 2004; 38(2):163-168; doi: 10.1140/epjb/e2004-00111-4

[34] Crescenzi P, D'angelo G, Severini L, et al. Greedily Improving Our Own Closeness Centrality in a Network. ACM Transactions on Knowledge Discovery from Data 2016; 11(1):1-32; doi: 10.1145/2953882

[35] Salavati C, Abdollahpouri A, Manbari Z. Ranking Nodes in Complex Networks Based on Local Structure and Improving Closeness Centrality. Neurocomputing. 2019; 336:36-45; doi: 10.1016/j.neucom.2018.04.086

[36] Uoh K.-I, Kahng B, Kim D. Universal Behavior of Load Distribution in Scale-Free Networks. Physical Review Letters; 91(18); doi: 10.1515/9781400841356.368

[37] Brandes U. A Faster Algorithm for Betweenness Centrality. The Journal of Mathematical Sociology 2001; 25(2):163–177; doi: 10.1080/0022250X.2001.9990249

[38] Zhou T, Liu J, Wang B. Notes on the Algorithm for Calculating Betweenness. Chinese Physics Letters 2006; 23(8):23-27; doi: 10.1088/0256-307X/23/8/099

[39] Berkhin P. A Survey on PageRank Computing. Internet Mathematics 2005; 2(1):73-120; doi: 10.1080/15427951.2005.10129098

[40] Xu S, Wang P. Identifying Important Nodes by Adaptive LeaderRank. Physica A: Statistical Mechanics and its Applications 2017; 469(C):654-664; doi: 10.1016/j.physa.2016.11.034

[41] Sung J.K, Sang H.L. An Improved Computation of the PageRank Algorithm. BCS-IRSG European Colloquium on IR Research 2002; 73-85; doi: 10.1007/3-540-45886-7_5

[42] Avrachenkov K, Litvak N. The Effect of New Links on Google Pagerank. Stochastic Models 2006; 22:319-331; doi: 10.1080/15326340600649052

[43] Li Q, Zhou T, Lü L, et al. dentifying Influential Spreaders by Weighted LeaderRank. Physica A: Statistical Mechanics and its Applications 2014; 404:47-55; doi: 10.1016/j.physa.2014.02.041

[44] Lü L, Zhang Y-C, Yeung CH, et al. Leaders in Social Networks, the Delicious Case. PLoS ONE 2011; 6(6):1-9; doi: 10.1371/journal.pone.0021202

[45] Zhang X, Yu H, Zhang C, et al. An Improved Weighted HITS Algorithm Based on Similarity and Popularity. International Multi-Symposiums on Computer and Computational Sciences (IMSCCS) 2007; 477-480; doi: 10.1109/IMSCCS.2007.67

[46] Lü L, Chen D, Ren X-L, et al. Vital Nodes Identification in Complex Networks. Physics Reports 2016; 650:1-63; doi: 10.1016/j.physrep.2016.06.007

[47] Restrepo J.G, Ott E, Hunt B.R. Characterizing the Dynamical Importance of Network Nodes and Links. Physical Review Letters 2006; 97(9):094102; doi: 10.1103/PhysRevLett.97.094102

[48] Stephenson K, Zelen M. Rethinking Centrality: Methods and Examples. Social Networks 1989; 11(1):1-37; doi: 10.1016/0378-8733(89)90016-6

[49] Poulin R, Boily M.C, Mâsse B.R. Dynamical Systems to Define Centrality in Social Networks. Social Networks 2000; 22(3):187-220; doi: 10.1016/S0378-8733(00)00020-4

[50] Zhou T, Kuscsik Z, Liu J-G, et al. Solving the Apparent Diversity-accuracy Dilemma of Recommender Systems. Applied Physical Sciences 2010; 107(10):4511-4515; doi: 10.1073/pnas.1000488107

[51] Yang Y, Wang X, Chen Y, et al. A Novel Centrality of Influential Nodes Identification in Complex Networks. IEEE Access 2020; 8:58742-58751; doi: 10.1109/ACCESS.2020.2983053

[52] Jia P, Liu J, Huang C, et al. An Improvement Method for Degree and its Extending Centralities in Directed Networks. Physica A: Statistical Mechanics and its Applications 2019; 532(15):121891; doi: 10.1016/j.physa.2019.121891

[53] Chen Y, Zhao Y, Li X, et al. An Open Dataset of Data Lineage Graphs for Data Governance Research. Visual Informatics 2024; 8(1):1-5; doi: 10.1016/j.visinf.2024.01.001

[54] Lü L, Chen D-B, Zhou T. The Small World Yields the Most Effective Information Spreading. New Journal of Physics 2011; 13(12):123005; doi: 10.1088/1367-2630/13/12/123005

Table 1 Descriptions of data asset types and relation types

	Type	Description
Data Asset	Data Table	A relational data table that stores information
	Data Job	A segment of SQL codes used to process data in data table assets
	Data Field	A column in a data table asset
Relation	DATA_FLOW	Data transformation between data table assets and data job assets
	PARENT_CHILD	Affiliation between data table assets and data field assets

Table 2. Basic information of DLGs in the DLG-DG-23 dataset

Scenarios	ID	Nodes	Edges	Scale
Cloud Infrastructure	DLG1	298	298	Small
	DLG2	464	467	Small
	DLG3	603	610	Small
	DLG4	415	414	Small
	DLG5	1,299	1,351	Medium
	DLG6	13,840	14,325	Large
Customer Service	DLG7	574	574	Small
	DLG8	546	546	Small
	DLG9	756	771	Small
	DLG10	5,378	5,555	Medium
	DLG11	2,024	2,032	Medium
	DLG12	5,645	5,802	Medium
Operation Analysis	DLG13	2,157	2,211	Medium
	DLG14	453	452	Small
	DLG15	558	564	Small
	DLG16	5,574	5,876	Medium
	DLG17	651	652	Small
	DLG18	17,085	17,720	Large

Table 3. Questions of the subjective questionnaire used in the field study

Usability	1. How easy is this method to use?
	2. Can you quickly master this method?
Effectiveness	3. How effective is this method in identifying core data assets?
	4. Can this method support you in identifying core data assets across business scenarios?
Satisfaction	5. Does this method support your daily work?
	6. How satisfied are you with this method overall?

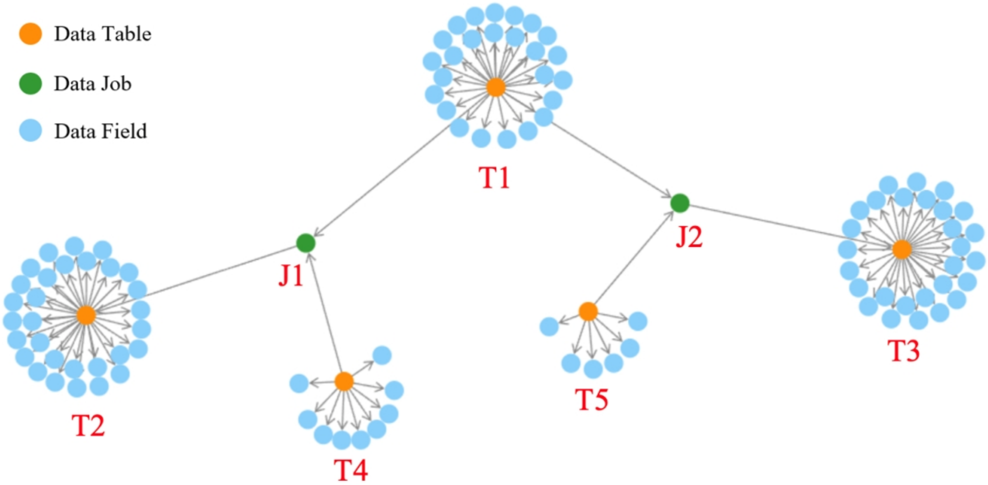


Figure 1. Example of a DLG that contains 105 data assets and 104 relations.

655x323mm (57 x 57 DPI)

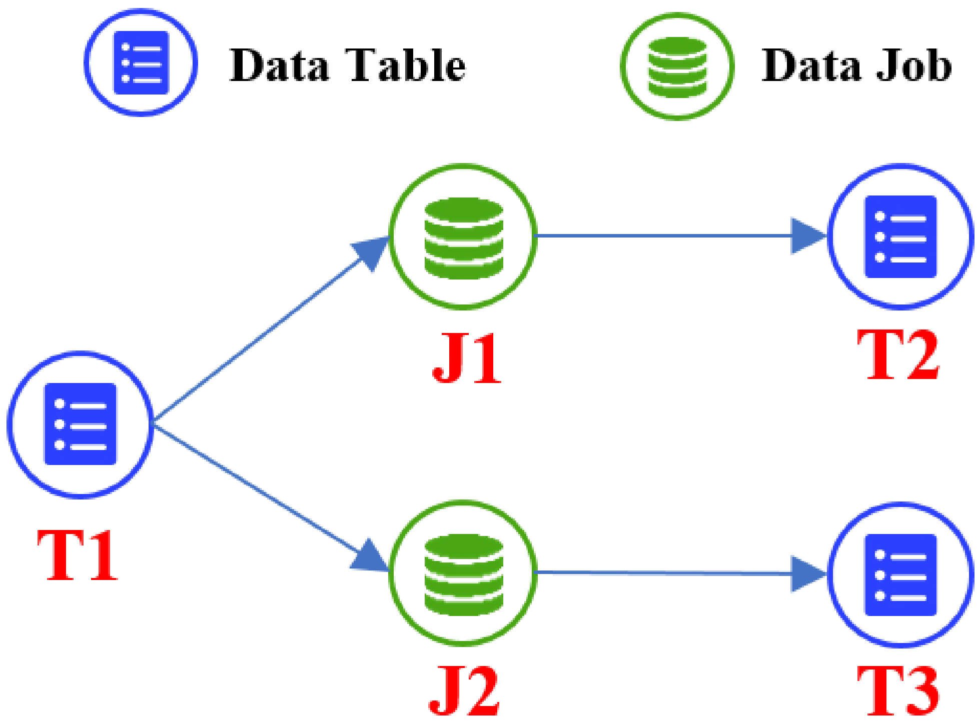


Figure 2. Example of heat propagation in heat diffusion paths.

232x169mm (57 x 57 DPI)

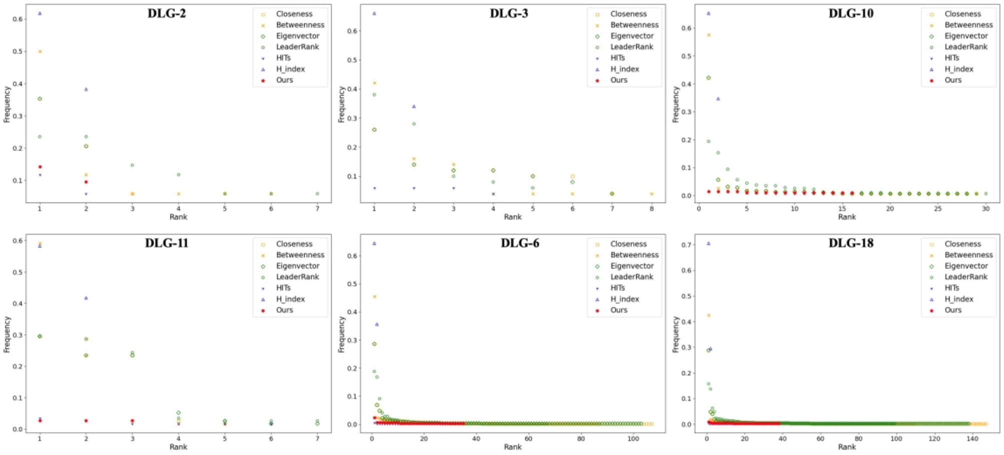


Figure 3. The frequency of nodes with the same ranking value in six DLGs of the dataset. The horizontal axis represents the ordering of nodes, while the vertical axis represents the number of nodes with the same ordering.

1385x622mm (57 x 57 DPI)

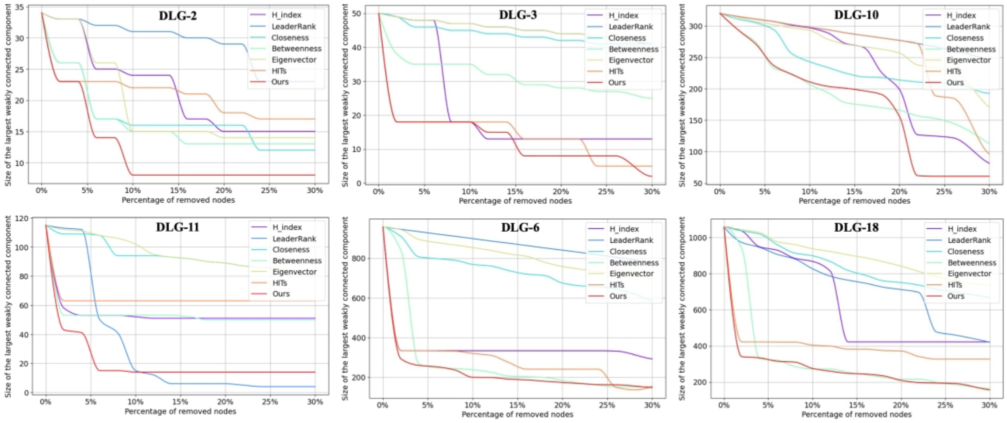
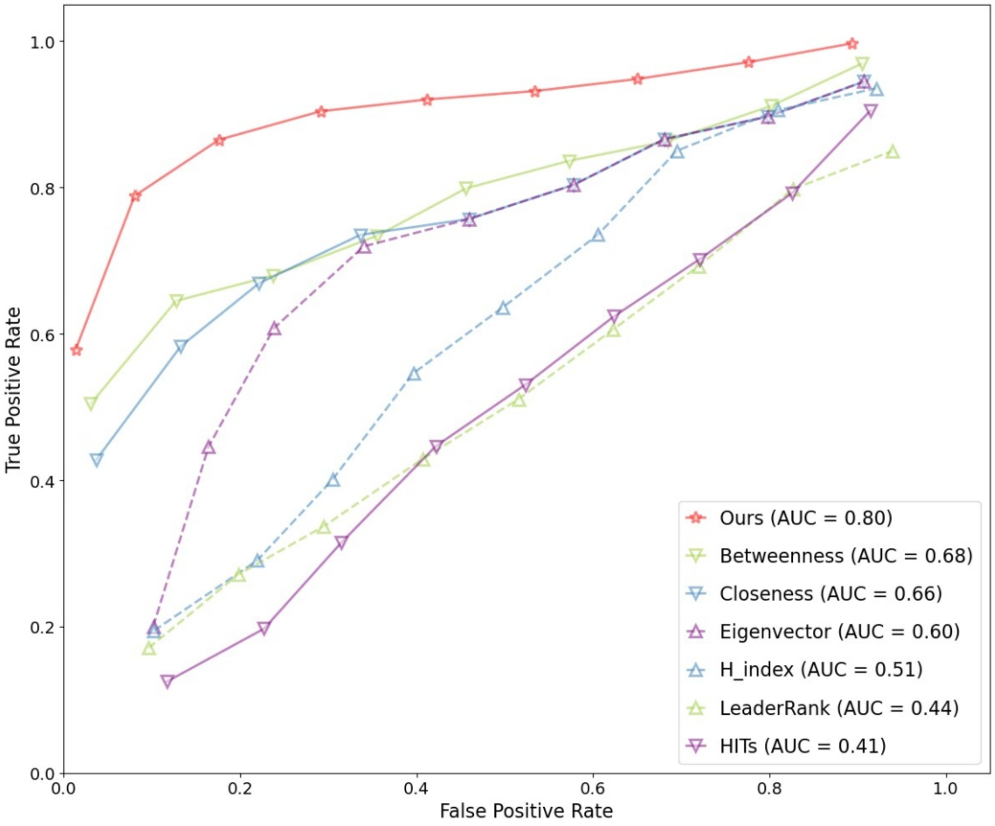


Figure 4. Variation on the size of the largest weakly connected component when removing different proportions of nodes in six DLGs of the dataset.

1400x581mm (57 x 57 DPI)



The average ROC curves of seven algorithms on small and medium-scale DLGs of the dataset.

794x661mm (38 x 38 DPI)