Enabling Secure Trustworthiness Assessment and Privacy Protection in Integrating Data for Trading Person-Specific Information

Rashid Hussain Khokhar, Farkhund Iqbal, Benjamin C. M. Fung, *Senior Member, IEEE,* and Jamal Bentahar, *Member, IEEE*

***Abstract*—With the increasing adoption of cloud services in the e-market, collaboration between stakeholders is easier than ever. Consumer stakeholders demand data from various sources to analyze trends and improve customer services. *Data-as-a-Service* (*DaaS*) enables data integration to serve the demands of data consumers. However, the data must be of good quality and trustful for accurate analysis and effective decision-making. In addition, a data custodian or provider must conform to privacy policies to avoid potential penalties for privacy breaches. To address these challenges, we propose a two-fold solution: (1) We present the first information entropy-based trust computation algorithm, *IEB_Trust*, that allows a semi-trusted arbitrator to detect the covert behavior of a dishonest data provider and chooses the qualified providers for data mashup. (2) We incorporate the *Vickrey-Clarke-Groves (VCG) auction mechanism* for the valuation of data providers' attributes into the data mashup process. Experiments on real-life data demonstrate the robustness of our approach in restricting dishonest providers from participation in the data mashup and improving the efficiency in comparison to provenance-based approaches. Furthermore, we derive the monetary shares for the chosen providers from their information utility and trust scores over the differentially-private release of the integrated dataset under their joint privacy requirements.**

***Index Terms*—Cloud computing, data trustworthiness, data privacy, data mashup, monetary valuation.**

I. INTRODUCTION

DATA are the fuel of today's digital economy. Yet, data coming from a single source often fail to provide a complete picture for big data analytics. To answer complex queries, companies usually have to seek additional data from multiple sources. The emerging cloud paradigm *Data-as-a-Service* (*DaaS*) provides an ideal platform for data integration in order to serve data consumers' demands. However, business data often contain person-specific information. Mashing up personal data from different sources raises concerns on security, privacy, and data reliability. In the past decade, many trust models [6], [67] and frameworks [15], [57] have been proposed to evaluate and measure the security strength of cloud environments, but limited research considers the aspect of data reliability. In this article, we propose a cloud-based data integration solution that considers privacy protection, data trustworthiness, and fairness of profit distribution among data providers.

According to a recent survey [24], organizations in the U.S. estimate that 33% of their customer data is inaccurate. This skepticism about data elicits the increased risk of non-compliance and regulatory penalties. The study by IBM estimated that $3.1 trillion of the U.S.'s GDP is lost due to poor quality data [64]. Organizations may mitigate these potential risks by taking appropriate measures regarding the quality of their data, leading to more reliable analysis and decision-making. There is a line of research [13], [42] that focuses on exchanging data between multiple parties from the perspective of ensuring confidentiality and integrity. These works aim to provide prevention from unauthorized use and modification when data is in transit but do not verify data if any party provides false data. Our research perspective is to determine the trustfulness of private data held by dishonest data providers who may arbitrarily attempt to provide false data when trading person-specific information in the e-market for monetary benefits. Our proposed method can detect such behavior from dishonest data providers, who resemble adversaries under the covert security model [7]. In literature [3], [17], [26] two protocols are discussed, namely *Private Set Intersection* (*PSI*) and *Private Set Intersection Cardinality* (*PSI-CA*) for privacy and data quality assessment. Freudiger et al. [27] claimed that these protocols are incurred from computational overhead and thus are not applicable to real-world scenarios. They proposed some protocols that operate on reduced dimensionality descriptions and so can be scalable to large datasets. It is a challenging problem to evaluate the trustfulness of private data held by untrusted data providers. In this article, we study the problem of untrusted data providers holding overlapping attributes on a person-specific dataset. We illustrate the problem in the following example.

Example 1. Suppose there is a cloud-based data market, where data consumers can place their data mining requests and data providers compete with each other to contribute their data with the goal of fulfilling the requests for monetary reward. Consider the 12 raw data records in Table I, where each record corresponds to the personal information of an individual. The three data providers own different yet overlapping sets of

R. H. Khokhar is with the Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada (email:r_khokh@ciise.concordia.ca).

F. Iqbal is with the College of Technological Innovation, Zayed University, Abu Dhabi, United Arab Emirates (email:farkhund.iqbal@zu.ac.ae).

B. C. M. Fung is with the School of Information Studies, McGill University, Montreal, QC, Canada (email:ben.fung@mcgill.ca).

J. Bentahar is with the Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada (email:bentahar@ciise.concordia.ca).

attributes over the 12 records.

Since the data providers collect data from different channels, it is quite possible that their data conflict with each other as illustrated in Table I. According to the predefined generalization hierarchy of the attributes in Fig. 1, the individuals in the table can be generalized to two groups: *Non-Technical* and *Technical*. Suppose a data consumer wants to perform a data analysis that depends on the *Non-Technical* and *Technical* groups. Yet, the inconsistent, conflicting, or even inaccurate data may mislead the analysis result. For example, DP_1 and DP_3 state that the individuals in $\{Rec\#3, 5\}$ are $Cleaner$, while DP_2 states that they are $Technician$. A similar conflict can be seen in the $Rec\#9$, where DP_1 and DP_3 provide the Job as $Painter$, and DP_2 provides the Job as $Welder$. In this example, the Job attribute on $\{Rec\#3, 5, 9\}$ has two different values that are categorized as *Non-Technical* and *Technical*, respectively. These inconsistencies significantly impact the quality of data analysis. ■

Presumably the data providers would have missing values on some attributes, although the same set of records is identified by executing the secure set intersection protocol [3] on the globally unique identifiers [53], [54]. Instead of avoiding participating in the data mashup process, they would prefer to impute missing values by using the machine learning methods appropriate for their datasets. The properties of a dataset such as low dimensional or high dimensional data, single-type or mixed-type data, or linearly separable or non-linearly separable data are a crucial factor before choosing the imputation method. The data providers' decision whether to use a single imputation method or multiple imputation methods is conditional on their missing data. We evaluate the robustness of our approach when an acquisitive data provider employs a machine learning method for imputation of missing data.

In the context of quantifying monetary value through sharing person-specific data, the data providers first must do the valuation of personal data, but there is no determined market price [56], [62] for person-specific data that can be taken as a proxy for the valuation. It is also well-acknowledged from existing literature [25], [58] that there is no commonly agreed methodology for valuing personal data. However, in the e-market, many companies actively collect personal information by providing monetary rewards to their customers. In this article, we incorporate the Vickrey-Clarke-Groves (VCG) auction mechanism for the valuation of data providers' attributes. We reason that it is a dominant strategy, where no data provider has an incentive to lie about his true valuations. In addition, private data often encode privacy-sensitive information related to individuals that need to be protected when integrating data from the competing data providers. In this article we adopt differential privacy [22] because it provides strong privacy guarantees to an individual independently of an adversary's background knowledge, in contrast to underlying assumptions in syntactic privacy models [47], [51], [66] about an adversary's knowledge.

Contributions. We propose a novel solution to address the critical issues of data trustworthiness, privacy protection, and profit distribution for cloud-based data integration services. The data trustworthiness problem has been studied in [49], [50], [69] applications of sensor networks. The provenance-based approach has been used in [16], [50] to evaluate the trustworthiness of network nodes and data items. This approach is primarily used to collect evidence about where the data originates and how the data generates. In this article we are not concerned about the high degree of the instrumentation of customers' private data, which is collected by data providers. However, our proposed approach makes novel use of information entropy to verify the correctness of data from untrusted data providers and also to preserve the privacy of customers' data held by data providers when evaluating the trustworthiness of the providers.

PSI-based approaches allow multiple parties to jointly compute the intersection of their private data without revealing any additional information to either side [75]. These approaches are suitable for *privacy-preserving distributed data mining* (*PPDDM*), in which multiple data custodians compute a function based on their inputs without sharing their data with others. In this article, we focus on *privacy-preserving data publishing* (*PPDP*) in a distributed setting, where the data providers wish to integrate their data for better information utility. However, the data integration necessitates that under the specified privacy constraints, no data provider should learn any additional information other than necessary information. We summarize our contributions as follows:

- Our proposed method, *IEB_Trust*, is the first entropy-based trust computation method that enables secure trustworthiness assessment and incorporates fairness in the verification process to restrict dishonest data providers from participation in the next phase for integrating data.
- We compare our proposed method with a closely related method. Results suggest that our entropy-based trust computation algorithm is capable of significantly improving runtime efficiency.
- We evaluate the robustness of our method when an acquisitive data provider adopts machine learning techniques to substitute missing values on their own data and claim them as original data collected from customers to compete with the other participating data providers.
- We define the procedure for setting the price on person-specific attributes in trading personal information from data providers based on the VCG mechanism.
- We integrate data from chosen data providers using *Differentially-private anonymization based on Generalization* (*DistDiffGen*) [53] and analyze the impacts of privacy protections and trust scores on data providers' monetary value.

The rest of the article is organized as follows: In Section II, we provide an overview of the trust mechanism and the problem statement. In Section III, we review the related work. In Section IV, we discuss the trust aspects, imputation methods, and privacy models. In Section V, we present our proposed solution. In Section VI, we compare our proposed method and provide empirical study to analyze the trustworthiness of each data provider and further analyze its impact along with the

TABLE I: Raw data owned by three data providers

	Data Provider DP_1			*Data Provider DP_2*			*Data Provider DP_3*		
RecID	**Age**	**Sex**	**Job**	**Sex**	**Education**	**Job**	**Age**	**Education**	**Job**
1	39	M	Lawyer	M	Bachelors	Lawyer	45	Doctorate	Lawyer
2	50	M	Lawyer	M	Masters	Lawyer	50	Doctorate	Lawyer
3	38	M	Cleaner	M	12th	Technician	35	12th	Cleaner
4	53	M	Lawyer	M	Doctorate	Doctor	57	Masters	Lawyer
5	28	F	Cleaner	F	11th	Technician	28	11th	Cleaner
6	37	F	Welder	F	12th	Welder	37	11th	Welder
7	49	F	Painter	F	12th	Cleaner	49	12th	Painter
8	59	M	Doctor	F	Doctorate	Doctor	66	Doctorate	Doctor
9	31	F	Painter	M	12th	Welder	27	12th	Painter
10	42	M	Technician	M	Bachelors	Technician	42	Bachelors	Technician
11	37	M	Lawyer	M	Masters	Lawyer	38	Masters	Lawyer
12	30	M	Lawyer	M	Masters	Lawyer	28	Bachelors	Lawyer

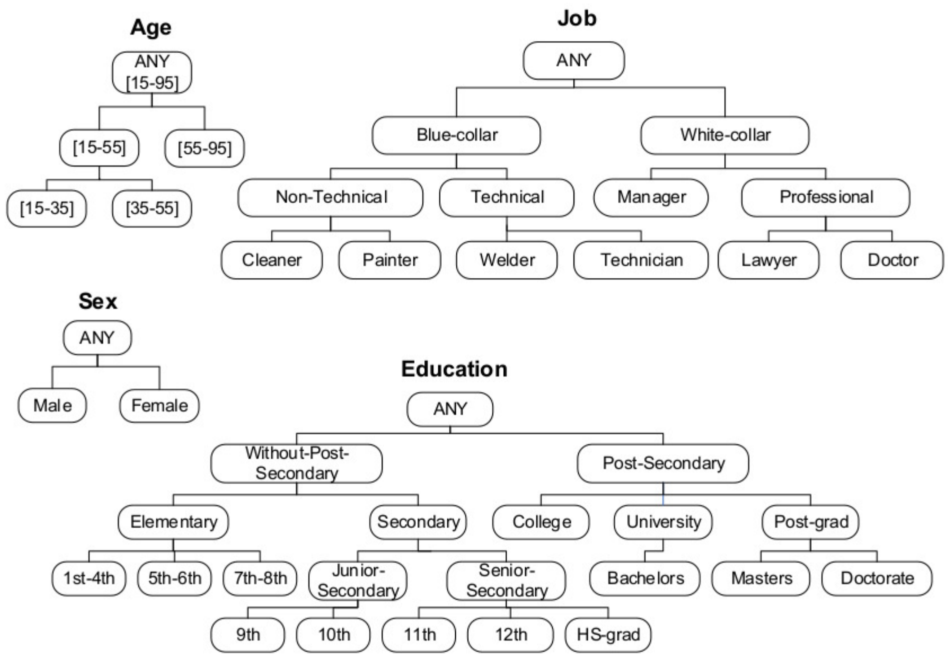


Fig. 1: Taxonomy trees

ϵ*-differential privacy* protection on a data provider's monetary value. Finally, we provide the conclusion in Section VII.

II. Trust Mechanism

In this section, we first provide an overview of our trust mechanism and then formally define the research problem.

A. Overview of trust mechanism

Fig. 2 provides an overview of our trust mechanism in which data providers, data consumers, and cloud service providers are the main entities. Data providers collect person-specific information from customers and intend to participate in the data mashup for generating more profit by competing with peer data providers, data consumers perform data analysis on the received data, and the cloud service provider (CSP) is a semi-trusted arbitrator between data providers and data consumers. The CSP manages three key services: authentication, mashup coordination, and data verification. These services are run on a cloud server by the CSP. First, each data provider has to pass the authentication phase to prove their identity. Second, data consumers submit their data requests to the CSP. In this article, we assume that a data consumer runs a classification analysis on its requested attributes by a supervised machine learning method. A resource queue is built by the mashup service to manage data requests from a data consumer, which is accessible only to authenticated data providers. Third, data providers register their available data attributes on the registry hosted by the mashup service; each data attribute is assigned a sequence number based on its arrival. Fourth, the verification process is run to detect false or incorrect data and to determine the trustworthiness of each data provider. Fifth, this process results in determining the accepted data providers. Sixth, the CSP connects the group of accepted data providers with the data consumer to serve its demand. This is done by the mashup service that determines the group of data providers whose data can collectively fulfill the demand of a data consumer. Seventh, the data providers quantify their costs and benefits using joint privacy requirements and integrate their data over the cloud. Finally, the anonymous integrated data is released to the data consumer.

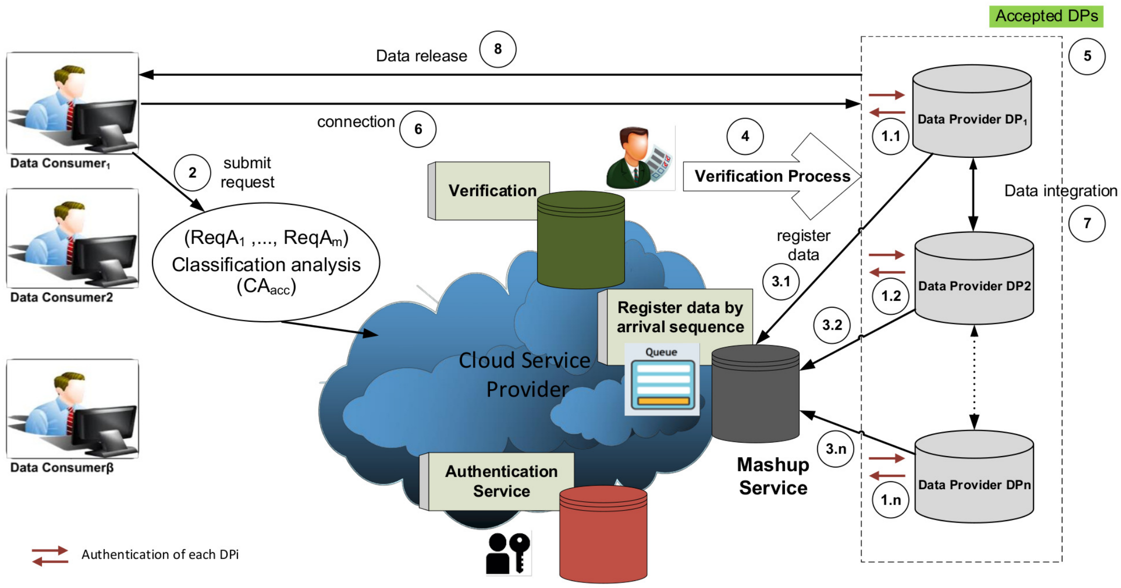


Fig. 2: Trust mechanism

B. Problem statement

We describe our problem as follows. There are three main entities discussed in our trust mechanism: data providers, data consumers, and a cloud service provider (CSP). *Data verification* service runs on a cloud server CS, which is managed by the CSP. The purpose of this service is to verify the correctness of data. The CSP is a semi-trusted arbitrator who would not have access to customers' private data, which is held by the data providers. Data providers are considered to be dishonest, meaning that they may arbitrarily attempt to provide false data because they are acquisitive in competing with others in the e-market. The behavior of such data providers is similar to adversaries in the covert security model.

Suppose data providers $DP_1,\ldots,DP_n$ own private data tables $D_1,\ldots,D_n$, respectively. Each record in the data table belongs to a unique individual. All explicit identifiers of an individual, such as name, social security number (SSN), and account number, have been removed. Each D_i is defined over a set of attributes $\mathcal{PA}_i = \{A_1,\ldots,A_d\}$. We assume that the data providers hold overlapping attributes for the same set of records identified by executing the secure set intersection protocol [3], [54] on the globally unique identifiers $RecID$. We require $\forall\mathcal{PA}_i\ \exists\mathcal{PA}_j$ such that $\mathcal{PA}_i \cap \mathcal{PA}_j \neq \emptyset$, where $i \neq j$, and $\mathcal{PA} = \{\mathcal{PA}_1,\ldots,\mathcal{PA}_n\}$. In addition, each D_i contains a A^{cls} attribute for classification analysis, which is shared among all the data providers. Each $A_{\mathcal{J}}$ is either a categorical or a numerical attribute, but A^{cls} is required to be categorical. A data consumer submits a data request $ReqA = \{ReqA_1,\ldots,ReqA_m\}$ for classification analysis. We assume that each data provider has $\mathcal{PA}_i \subseteq ReqA$ to serve the demand of a data consumer. The goal of this trust computation is to restrict dishonest data providers from participation in the data mashup process when their trust scores drop below a certain threshold.

Problem 1 (Trust computation). Given multiple person-specific raw data tables $D_1,\ldots,D_n$ from data providers $DP_1,\ldots,DP_n$ and a set of requested attributes $ReqA = \{ReqA_1,\ldots,ReqA_m\}$ for classification analysis from a data consumer, the research problem is to verify the correctness of data on the submissions of the overlapping set of attributes $\mathcal{PA}_i = \{A_1,\ldots,A_d\}$ on the same set of records from each data provider DP_i, where $\mathcal{PA}_i \cap \mathcal{PA}_j \neq \emptyset\ \forall\mathcal{PA}_i\ \exists\mathcal{PA}_j$ and $i \neq j$ and to compute the trust score TS_{DP_i} of each data provider.

In the context of data privacy, the data providers want to integrate their data in a way such that no data provider should learn any additional information about the others as a result of data integration. After the completion of trust computation, the data providers $DP_1,\ldots,DP_n$ attain a mutually exclusive set of attributes $\mathcal{PA}_i = \{A_1,\ldots,A_d\}$ over the same set of records for data integration. That is, $\mathcal{PA}_i \cap \mathcal{PA}_j = \emptyset$ for any $1 \leq i,j \leq n$. We assume that for each attribute $A_{\mathcal{J}} \in \mathcal{PA}_i$, a taxonomy tree is provided that defines the hierarchy of values in $\Omega(A_{\mathcal{J}})$, where $\Omega(A_{\mathcal{J}})$ represents the domain of $A_{\mathcal{J}}$. Data providers require doing their attributes' valuations for price setting and jointly setting up the privacy requirements, such as privacy budget ϵ and specialization level h for a ϵ-*differential privacy* model, before data integration. They wish to derive their monetary shares from the information utility of anonymous integrated data $\hat{D}$ for classification analysis and their trust scores.

Problem 2 (Monetary share under ϵ-differential privacy mechanism). Given multiple raw data tables $D_1,\ldots,D_n$ containing mutually exclusive sets of attributes $\mathcal{PA}_i = \{A_1,\ldots,A_d\}$, i.e., $\mathcal{PA}_i \cap \mathcal{PA}_j = \emptyset$ for any $1 \leq i,j \leq n$ over the same set of records, and a data request $ReqA = \{ReqA_1,\ldots,ReqA_m\}$ from a data consumer for classification analysis, the research problem is to derive the monetary share of each DP_i from their information utility and trust scores over the differentially-private release of integrated dataset $\hat{D}$ under the joint privacy requirements and attributes' valuations.

Several companies, such as Acxiom, AnalyticsIQ, Dataline, and Expedia, collect user data including demographic, financial, retail, social, and travel information from multiple sources with the goal of serving different market needs [1]. Our research problem can be generalized to other similar companies who face trustworthy or quality data issues [24] and whose business models are primarily based on sharing person-specific information.

III. Related work

In this section, we summarize the literature of the following related areas: data trustworthiness and auction-based pricing, cryptographic primitives, and differentially-private anonymization techniques.

A. Data trustworthiness and auction-based pricing

Different trust models, frameworks, and techniques have been proposed to address the problem of data trustworthiness. Bertino and Lim [11] proposed a framework that consists of two key components. The first component is based on the concept of data provenance in which information relies on the origin of data for computation of trust scores. The second component undertakes the notion of confidence policy in which query results are filtered based on the specified confidence range for use in certain tasks. Dai et al. [16] proposed a provenance-based model in which they evaluated the trustworthiness of data items based on the aspects of data similarity, path similarity, data conflict, and data deduction. Benjelloun et al. [8] introduced ULDBs in which they combined the concept of *lineage* and *uncertainty* for querying in probabilistic databases.

There are studies related to data trustworthiness in mission-critical applications [49], [69]. Tang et al. [69] proposed trustworthiness analysis for sensor networks in cyber-physical systems to eliminate false alarms that occur due to random noise or defective sensors. They validated events by using a graph-based filtering approach. However, their method does not deal with coordinated attacks where a fraction of sensing nodes are compromised by malicious attackers. Lim et al. [49] addressed this challenge by adopting a game-theoretic approach based on the Stackelberg competition for defending the network against false data injection. They assessed trust scores for both data items and network nodes using the cyclic framework proposed in [50]. This framework is based on the interdependency property between data items and their associated network nodes in which trust scores are computed using two types of similarity functions. First, *value similarity* is derived from the principle that the more that similar values refer to the same event, the higher the trust scores. Second, *provenance similarity* is based on the principle that the more that different data sources are with similar data values, the higher the trust scores. Mainly, the approaches presented in the above works fall under the category of workflow provenance. In contrast, we are not concerned about the higher level of instrumentation at the data collection phase by data providers because it is not practically efficient to determine the data provenance in the e-market. Furthermore, the above works mainly focus on similarity functions for trust computation but do not consider privacy protection for data trustworthiness. We propose an approach that makes novel use of information entropy to verify the correctness of data in a multiple data providers scenario where a semi-trusted arbitrator cannot derive any customers' private data when evaluating the trustworthiness of the participating data providers.

Karabati et al. [41] studied the challenge of pricing with short-term capacity allocation decisions for multiple products in a single-supplier and multiple-buyers scenario. They proposed an iterative auction mechanism with monotonically increasing prices to maximize the profit of a supplier. Li et al. [48] presented dynamic pricing strategies for resources allocations in cloud workflow systems. Their proposed reverse auction-based mechanism allows resource providers to change the prices during the auction, depending upon their trading situation, to improve the efficiency of resource utilization as well as the competitiveness. Wu et al. [72] employed a VCG auction to implement a dynamic pricing scheme for multi-granularity service composition. They considered both coarse-grained and fine-grained services for composition. In their approach, service providers bid for services of different granularities in the composite service, whereas a recipient of the bids decides a composition that minimizes the overall cost while satisfying quality constraints. They solved the problem of winner determination by an integer programming model. In this article, we define the procedure for the valuations of data providers attributes based on the VCG mechanism.

B. Cryptographic primitives

Private set intersection (PSI) is a cryptographic primitive that was first formally defined in [26]. The protocols for PSI allow two parties, holding sets A and B, to compute the private intersection without revealing to each other any additional information from their respective sets. At the end of the protocol, either one or both parties may learn the size of the intersection, depending on the application. Since its inception, many variants have been proposed in an attempt to speed up PSI computation, including garbled Bloom filters [20], [33], server-aided computations [19], [39], [40], and computational optimizations [46], [59], [61].

Oblivious Transfer (OT) is one of the fundamental primitives in cryptography and has been extensively used for secure multi-party computation. Particularly, the most efficient OTs were introduced by Pinkas et al. [61] and further strengthened in [46], [59], [60]. Kolesnikov et al. [46] proposed a batched related-key oblivious pseudo-random function (BaRK-OPRF) protocol to improve the performance of semi-honest secure PSI. They achieved a 1-out-of-n OT of random messages for an arbitrarily large n at nearly the same cost as 1-out-of-2 in Ishai et al. [35]. The new OPRF construction of Pinkas et al. [59] is similar to Kolesnikov et al. [46] except in handling error correcting code. Kolesnikov et al. [46] demonstrated that their protocol outperforms Pinkas et al. [60] in almost as many settings, particularly for the long bit length of input and large values of the input size.

In practice, the OT-based protocols are much faster than the random garbled Bloom filter-based protocols for larger set sizes, yet these protocols do not have the lowest communication cost [46]. One desirable property is to achieve the fairness that ensures either all the parties of a group learn the output of the computation or none do [39]. This is not the case with standard approaches to PSI. Our solution to the problem is different from several PSI-based approaches in which the intention is to achieve both privacy and security simultaneously. These approaches are suitable for different motivating applications in private data mining, online recommendation services, and genomic computations. In our approach, we maintain confidentiality and integrity by exchanging only an encrypted information gain message and its keyed hash between a data provider and the cloud server, based on a random challenge (i.e., attribute request) of the cloud server, instead of exchanging encrypted individual data items. This apparently reduces the overhead of communication. In addition, we do not rely on the server to perform the computation on clients' private data. In the context of privacy, PSI protocols enable parties to privately know the result from their intersection, but the total information is not published for data analysis [75]. However, we intend to securely integrate person-specific data from multiple data providers and to release differentially-private data for classification analysis.

C. *Differentially-private anonymization techniques*

Differential privacy is increasingly being accepted as the cornerstone of privacy protection by domain experts due to its robustness and rigorous mathematical definition. In literature, two settings, namely *interactive* and *non-interactive*, are mainly discussed regarding utilization of the privacy budget ϵ. The primary difference is that in the interactive setting [22], [28], [73], [74] the data custodian holds the raw data and a data analyst poses a set of queries in real time for which the data custodian provides differentially-private answers. Each query would utilize a fraction of ϵ incrementally to produce a noisy answer. When the entire privacy budget has been depleted, a data analyst would not be able to get the answer by querying the database. On the other hand, in the non-interactive setting, the data custodian first anonymizes its raw data by utilizing the entire privacy budget. Later, the anonymous (*ϵ-differentially private*) data releases to the data analyst, who would perform an analysis without any constraints on the data usage. This approach is widely known as privacy-preserving data publishing (PPDP) [30], which is more appropriate in many real-life data sharing scenarios because of the flexibility for a data analyst to perform an analysis without back and forth querying of the database. In this article, we focus on the non-interactive setting for a differentially-private release of data in a distributed setup.

The group of works [4], [53] based on distributed approaches are suitable for multiple parties whose prime concern is to integrate their data in a way that no party could learn any additional information about the other party as a result of data integration. Mohammed et al. [53] proposed an algorithm, called *DistDiffGen*, in which data is vertically partitioned among multiple parties in a distributed setup. It allows two parties to securely integrate their person-specific data while maintaining necessary information to support data utility. Each party in this setup owns a mutually exclusive set of attributes over the same set of records. A similar problem has also been studied by Alhadidi et al. [4] where data is horizontally partitioned among two parties. Each party in this setup owns a disjoint set of records over the same set of attributes. In this article, we employ *DistDiffGen* [53] for a distributed setup with an extension for multiple data providers to achieve *ϵ-differential privacy*. There are existing works that allow data integration for horizontally partitioned databases [37], [55] and vertically partitioned databases [29], [36], [54] under the privacy constraints in a distributed setup. These works are based on syntactic privacy models, which are vulnerable to certain attacks such as minimality attack [71], composition attack [32], and deFinetti attack [44]. Therefore, we adopt differential privacy [22] because it provides strong privacy guarantees against such attacks. Whereas existing work [43] proposed a privacy-preserving data mashup model that allows the collaboration of multiple data providers for integrating their data and derives the contribution of each data provider by evaluating the incorporated cost factors, in our work we derive the monetary shares for the chosen data providers from their contribution to information utility over the differentially-private integrated data for classification analysis and their trust scores.

IV. PRELIMINARIES

In this section, we first present the principles that are crucial for establishing trust. Next, we discuss methods for imputation of missing data, and finally, we discuss privacy models.

A. *Trust aspects*

Trust is a critical aspect of decision making in e-commerce. Trust principles are a part of many service-oriented architectures (SOA)-based models where participants in the system want to do interactions for service delivery and use [2]. We review the principles that are crucial for trust establishment. First, entities should be identified [38] as they have claimed. In the world of the Internet, where entities are physically isolated, they may have real identities or may use fake identities to show their existences in their interactions. Authentication is a way of validating entities by the use of usernames and passwords, tokens, or digital certificates before granting them access to the resources or applications [12]. Second, it is crucial for trust formation to initialize new entities with trust rates. This process is called *trust bootstrapping*. Third, when one entity trusts another entity's decision there is a risk of an undesirable outcome due to some degree of uncertainty and dependency [45]. The risk is considered to be a prerequisite before trusting the trustee's behavior. The entities who are involved in an interaction should comply with the norms and rules of trust to avoid penalties for violation. Fourth, trust rates are of two types: local and global [70]. Local trust rating refers to a personalized score in which each trustee would have different scores from the trustors. Global trust rating provides a unique score about the entity (trustee) independently of who are the entities (trustors) participating in the evaluation. Global trust rating often requires the trusted third party (TTP) services to collect feedback from the trustors about trustees and compute the trust rates. Last, security and privacy are the main components for trust establishment. Trust is required when there is uncertainty; it has widely been accepted that perfect security does not exist, even though security measures are necessary to gain trust in many circumstances [10]. Customers who place their orders online and submit private information in the form of their name, address, and credit details necessitate that their private information should not be disclosed or shared by any means with untrusted parties. Building a trust relationship requires protection of customers' privacy in online transactions. We pay attention to some of the aforementioned principles for establishing trust on the data providers in the context of our trust mechanism.

B. Methods for imputation of missing data

There are different types of missing data [34], such as Missing at Random (MAR), Missing Completely at Random (MCAR), and Missing Not at Random (MNAR). MAR refers to the probability of missing data of an attribute on other present observations of attributes in the dataset, but not on the attribute's own value. Whereas, MCAR occurs when there is no dependency on the attribute value itself or any other attribute in the dataset. And the special case MNAR occurs when the missing data meets neither the condition defined in MAR nor MCAR. In this special case, missing values in MNAR cannot be imputed by using other present observations of attributes.

There is extensive research [5], [9], [76], [77] done on machine learning methods such as hot-deck imputation, mean imputation, regression imputation, k-nearest neighbors imputation, and random forest imputation. Hot-deck imputation is a technique for replacing missing values of a non-respondent on one or more attributes with the most similar characteristics to a respondent [5]. This method has been used in practice, but the theory is not as well developed. Mean imputation is a technique used for replacing missing values of a numerical attribute by the average value, and for a categorical attribute by the mode, i.e., most frequent value. This method is quite simple, but it is not suitable for multivariate analysis. Regression imputation first builds a model from the observed data, then predictions for the incomplete cases are calculated under the fitted model to replace the missing data [77]. The drawback of the regression model is that all predicted values fall directly on the regression line, which decreases variability. Random forest is a type of ensemble learning method [76]. It is used widely for classification and regression tasks. The learning process of a random forest algorithm is based upon the bootstrap aggregation technique, in which a specified number of trees are trained on a given dataset. As the random forest is built upon multiple decision trees, intrinsically it uses the same approach for attribute selection measures such as information gain, gini index, and gain ratio of decision trees. Random forest can deal with missing values with different types of variables. k-nearest neighbors (kNN) imputation is an efficient approach for replacing missing values on some records by computing another value from similar examples in the given dataset [9]. kNN computes the similarity by using a distance metric, such as Euclidean distance. k is a positive integer, when $k = 1$ the object is simply assigned to the class of that single nearest neighbor. When $k > 1$ the object is assigned to the class that appears most frequently within the k-subset. kNN generally produces good quality predictions, but the computation cost is high because of computing distances.

C. Privacy models

In the literature, there are two types of models apprehended: $syntactic$ and $semantic$. Syntactic models, such as K-anonymity [66] protects against identity disclosure, l-diversity [51] protects from homogeneity attacks, and t-closeness [47] is an extension of l-diversity in which the distribution of sensitive attribute values for privacy protection is further refined. Differential privacy [22] is a semantic model that is more robust against the aforementioned attacks. It provides strong privacy guarantees to an individual independently of an adversary's background knowledge. The intuition of differential privacy is that individual information is not revealed from the output of the analysis in the anonymized data. In other words, it is insensitive whether an individual record is present in the input dataset or not. It is mathematically defined as follows.

Definition IV.1 (*ϵ-differential privacy*). [22]. A sanitization mechanism M provides *ϵ-differential privacy*, if for any neighboring datasets D and D' differing by at most one record (i.e., symmetric difference $|D \triangle D'| \leq 1$), and for any possible sanitized dataset $\hat{D}$,

$$\Pr[M(D) = \hat{D}] \leq e^{\epsilon} \times \Pr[M(D') = \hat{D}],$$

where the probability is taken over the randomness of the M. ■

ϵ is the privacy budget that is specified by the data custodian. A smaller value of ϵ results in stronger privacy protection but produces lower data utility. Conversely, a larger value of ϵ results in weaker privacy protection but yields higher data utility.

The *Laplace mechanism* and *exponential mechanism* are the canonical examples of a differentially-private mechanism. A standard mechanism to achieve differential privacy is to add random noise to the outcome of the analysis for providing privacy protection. The calibration of noise is done according to the *sensitivity* of the function f.

Definition IV.2 (*Sensitivity*). For any function $f : D \to R^d$, the sensitivity of f is

$$\Delta f = \max_{D,D'} ||f(D) - f(D')||_1 \tag{1}$$

for all D, D' differing at most by one record. ■

The sensitivity of a function does not depend on the data but instead produces an upper bound to how much noise we must add to the true output to preserve privacy. Suppose function f answers count queries over a dataset D. Then, the Δf is 1 because $f(D)$ can differ at most by 1, due to the addition or removal of a single record.

Laplace mechanism. Dwork et al. [22] proposed the Laplace mechanism. It is appropriate when the output of function f is a real value, and f should perturb its output with a noisy answer to preserve privacy. The noise is calibrated based on the privacy parameter ϵ and the sensitivity of the utility function Δf. Formally, the Laplace mechanism takes as inputs a data set D, the privacy parameter ϵ, and a function f and outputs $\hat{f(D)} = f(D) + \mathtt{Lap}(\lambda)$, where $\mathtt{Lap}(\lambda)$ is a noise drawn from the Laplace distribution with probability density function $\Pr(x|\lambda) = \frac{1}{2\lambda}\mathtt{exp}(-|x|/\lambda)$. The variance of this distribution is $2\lambda^2$, and the mean is 0.

Exponential mechanism. McSherry and Talwar [52] proposed the exponential mechanism. It is appropriate for situations in which it is desirable to choose the best response, because adding noise directly to the count can eradicate its value. Given an arbitrary range $\mathcal{T}$, the exponential mechanism is defined with respect to a utility function $u : (D \times \mathcal{T}) \rightarrow R$ that assigns a real valued score to every output $t \in \mathcal{T}$, where a higher score means better utility. The exponential mechanism induces a probability distribution over the range $\mathcal{T}$ and then samples an output t. Suppose $\Delta u = \max_{\forall t, D, D'} |u(D,t) - u(D',t)|$ to be the sensitivity of the utility function. The probability associated with each output t is proportional to $\mathtt{exp}(\frac{\epsilon u(D,t)}{2\Delta u})$.

V. Proposed solution

In this section, we provide a solution to address the concerns of stakeholders on data trustworthiness, privacy protection, and profit distribution in the online market for trading person-specific data. Section V-A presents our proposed ***IEB_Trust***, an information entropy-based trust computation algorithm to restrict dishonest data providers from participation in the data mashup process and to assess the trustworthiness of each data provider. Section V-B discusses security properties. Section V-C provides an analysis of ***IEB_Trust*** algorithm. Section V-D provides an evaluation of learner models. Section V-E provides an auction mechanism for price-setting among data providers who own multiple attributes. Section V-F presents an algorithm for privacy protection by which data providers can determine the impact of anonymization on data utility for classification analysis. Section V-G discusses how the chosen data providers can quantify their monetary value.

A. Trust computation

In Section II-B, we state the problem where the challenge is to verify the correctness of data from untrusted multiple data providers who own overlapping attributes for the same set of records. We assume that the data providers are competitors who intend to maximize their profits. The data providers consider as dishonest anyone who may arbitrarily attempt to provide false data to get a larger monetary share from their participation. To address this problem, we propose a novel algorithm that adopts information entropy for secure trustworthiness assessment of acquisitive data providers. Information entropy has been widely used in machine learning tools and decision-making systems. Compared to the existing work on data trustworthiness [49], [50], [69], our proposed algorithm not only detects false or incorrect data from a dishonest data provider during the verification process, but also preserves the privacy of customers' data owned by a data provider. Furthermore, our method provides better runtime efficiency over provenance-based approaches [16], [50].

Algorithm 1 presents our approach in more detail. A cloud service provider (CSP) runs this algorithm on a cloud server(CS). Consider multiple data providers $DP_1, \ldots, DP_n$, who own private data tables $D_1, \ldots, D_n$ having overlapping attributes for the same set of records identified by the common record identifier $RecID$ [3], [54]. First, the CS and each DP_i mutually authenticate each other and derive ks_i symmetric keys for all $i \in I$ by the mutual authentication protocol [18] for the secure exchange of messages. Each DP_i has its own ks_i to answer the CS's queries. Second, a data consumer submits a data request $ReqA = \{ReqA_1, \ldots, ReqA_m\}$ to the CS. Third, each data provider DP_i submits an available set of attributes $\mathcal{PA}_i = \{A_1, \ldots, A_d\}$, where $\mathcal{PA}_i \subseteq ReqA$, to the CS. We assume that initially all the participating data providers have "zero" in their trust scores (Line 3). ϵ' is the allocated privacy budget to consume for each requested attribute. A resource queue is created by the mashup service for m requested attributes, where each attribute $A_{\mathcal{J}} \in \mathcal{PA}_i$ of a corresponding data provider is registered with its arrival sequence (Line 9).

Fourth, the verification process is run to determine the trustworthiness of each data provider. In the first round, CS successively selects one attribute $ReqA_x{}'$ uniformly at random without replacement over a domain of m requested attributes and sends an encrypted challenge $E(ks_i, ReqA_x{}')$ to the corresponding data providers $DP_1, \ldots, DP_n$, who own common attribute $A_{\mathcal{J}}$. Prior to responding to this challenge, each DP_i decrypts to retrieve $ReqA_x{}'$, computes information gain on the challenge attribute in Line 16, denoted by $\mathcal{G}^{(1)}_{A_{\mathcal{J}}}$ (refer to Section V-A1 for details), according to Eq. (4) [63] and then adds noise to a true output. Then DP_i encrypts the message $\psi^{(1)} \leftarrow E(ks_i, \mathcal{G}^{(1)'}_{A_{\mathcal{J}}})$ and computes tags $\Upsilon^{(1)} \leftarrow \mathcal{S}(k_h, \psi^{(1)})$ by using *keyed hash-based message authentication code* (*HMAC*) in Line 17. CS receives the concatenated message, tag, and identity $\psi^{(1)} \| \Upsilon^{(1)} \| DP_i$ on his challenge from each data provider. Then CS computes the comparison to determine the majority candidates by invoking procedure findMajCand($\psi^{(1)} \| \Upsilon^{(1)}$, $size$) in Line 19, where $size$ indicates the number of data providers who own the requested attribute. This procedure returns majority candidate $Maj^{R(1)}_{Cand}$. In the second round, CS generates $\mathcal{K}$ random IDs for the requested challenge $ReqA_x{}'$, i.e., picked in the first round, from $|D_i|$ records, then generates $\mathcal{P}$ pairs of values for $ReqA_x{}'$ and A^{cls} attributes. CS sends another challenge to each DP_i by concatenating the encrypted $\mathcal{K}$ random IDs and $\mathcal{P}$ pairs of values as $E(ks_i, \mathcal{K}, ReqA_x{}') \| E(ks_i, v_{x'}, v_{cls})$. DP_i decrypts to retrieve $\mathcal{K}$ record IDs and $\mathcal{P}$ pairs of values.

DP_i concatenates $\mathcal{K}$ records and $\mathcal{P}$ pairs of values received from the CS. DP_i computes $\mathcal{G}^{(2)}_{A_{\mathcal{J}}}$ on the concatenated version and then adds noise to a true output, encrypts it as $\psi^{(2)} \leftarrow E(ks_i, \mathcal{G}^{(2)'}_{A_{\mathcal{J}}})$, and computes the tag as $\Upsilon^{(2)} \leftarrow \mathcal{S}(k_h, \psi^{(2)})$. CS receives $\psi^{(2)} \| \Upsilon^{(2)} \| DP_i$ on the second round challenge from the corresponding data providers in Line 28. CS again invokes procedure findMajCand($\psi^{(2)} \| \Upsilon^{(2)}$, $size$) to determine the majority candidates in Line 29. This process repeats α times. In Line 33 an intersection of both the rounds is computed to determine Maj_{Cand}.

Candidates whose scores match on the majority are considered as Qualified, denoted by $Qual_{DP_i}$, who gain a positive weight γ in their trust scores TS_{DP_i}. Alternatively, candidates whose scores do not match are considered as Non-Qualified, denoted by $UnQual_{DP_i}$. Subsequently, $UnQual_{DP_i}$ is penalized with a negative weight -γ in their trust scores TS_{DP_i}. When only a single data provider responds to the CS challenge of $ReqA_x{}'$, it is accepted based on his existing trust score $TS_{DP_i} \geq 0$. However, in this case, the trust score does not increase for that data provider. When a data consumer request for an attribute, which is not fulfilled by the participating data providers, then that attribute is excluded from the verification process, and the data providers gain no monetary value from it. The comparison is performed (Line 45) to select one candidate (or data provider) on each attribute from the qualified data providers $Qual_{DP_n}$ based on their arrival sequences (using first-come first-served (FCFS) rule). If the final aggregated trust score of any data provider becomes < 0 that data provider drops from the final selection for the data mashup and the attributes initially belonging to him are subsequently reassigned to other qualified data providers that appear next in the arrival sequences. The algorithm terminates when there is no more attribute for verification.

1) Computation of information gain: We use information gain as a criterion for splitting attributes [63] based on the concept introduced by Claude Shannon on information theory [68]. We compute information gain on an individual attribute $A_{\mathcal{J}} \in \mathcal{PA}_i$ of each data provider in the presence of a shared class attribute A^{cls} on raw data. Let $D^{\tau} \subseteq D_i$ denote a subset of the data table D_i. Suppose the attribute A^{cls} has $\mathcal{C}$ distinct values. Let $A^{cls}_{i,D^{\tau}}$ be the set of records of class A^{cls}_i in D^{τ}. Let $|D^{\tau}|$ and $|A^{cls}_{i,D^{\tau}}|$ denote the number of records in D^{τ} and $A^{cls}_{i,D^{\tau}}$, respectively. The entropy on the data table D^{τ} is computed as follows.

$$E(D^{\tau}) = -\sum_{i=1}^{\mathcal{C}} \ \mathrm{Pr}_i \times log_2 \ \mathrm{Pr}_i \tag{2}$$

where Pr_i is the probability that an arbitrary record in D^{τ} belongs to class A^{cls}_i. It is estimated by $\frac{|A^{cls}_{i,D^{\tau}}|}{|D^{\tau}|}$.

We can further partition the records in D^{τ} on the attribute $A_{\mathcal{J}}$. If $A_{\mathcal{J}}$ is discrete-valued, then one branch is grown for each known value of $A_{\mathcal{J}}$. On the other side, if $A_{\mathcal{J}}$ is continuous-valued, then two branches are grown, corresponding to $A_{\mathcal{J}} \leq splitpoint$ and $A_{\mathcal{J}} > splitpoint$. It is calculated by the following equation.

$$E_{A_{\mathcal{J}}}(D^{\tau}) = \sum_{j=1}^{\mathcal{V}} \ \frac{|D^{\tau}_j|}{|D^{\tau}|} \times E(D^{\tau}_j) \tag{3}$$

Finally, we can compute the information gain $\mathcal{G}_{A_{\mathcal{J}}}$ on the chosen attribute $A_{\mathcal{J}}$ of each data provider DP_i as follows.

$$\mathcal{G}_{A_{\mathcal{J}}} = E(D^{\tau}) - E_{A_{\mathcal{J}}}(D^{\tau}) \tag{4}$$

2) Differentially-private $\mathcal{G}_{A_{\mathcal{J}}}$: Given a privacy budget ϵ', the sensitivity of the utility function (Δf) is 1, and a true computed $\mathcal{G}_{A_{\mathcal{J}}}$. We add independently generated noise from the Laplace distribution $\texttt{Lap}(1/\epsilon')$ to a true computed $\mathcal{G}_{A_{\mathcal{J}}}$ to have a differentially-private version of Eq. (4).

$$\mathcal{G}'_{A_{\mathcal{J}}} = \mathcal{G}_{A_{\mathcal{J}}} + \texttt{Lap}(1/\epsilon') \tag{5}$$

3) Discretization: We use *equal-width* method to discretize a continuous-valued attribute $A_{\mathcal{J}}$ into K intervals of equal size. The min_{val} and max_{val} parameters are used for defining the boundaries of the range, whereas arity K is used to determine the number of bins. Each bin is associated with a distinct discrete value. The width of interval is computed by

$$Int_{width} = \frac{max_{val} - min_{val}}{K} \tag{6}$$

Example 2. We continue from Example 1. Consider the example data of numerical type attribute in Table II. In this table *Age* is a numerical attribute, whereas *Loan approval* is an A^{cls} attribute. Data providers DP_1 and DP_3 own raw data tables Table II.(a) and Table II.(b), respectively. DP_3 has somewhat different values on the *Age* attribute in contrast to DP_1 on records $\{ID\#1, 3, 4, 8, 9, 11, 12\}$. They discretize their data on the *Age* attribute, as shown in Table II.(c), according to the parameters of equal width binning. A boundary is defined as $min_{val} = 10.0$ and $max_{val} = 70.0$, whereas arity $K = 5$. Though they have differences in their raw data, the produced discrete version is the same for both since the data values occurred in the specified range. Therefore, the computed information gain 0.34573 is also the same. ■

Example 3. We continue from Example 1. Consider the raw data tables of two data providers who own common attribute, e.g., *Sex* (which has two values, M or F) as shown in the compressed Table III. The class attribute *Loan approval* shared between the data providers has two values, Y or N, indicating whether or not the loan is approved. Both DP_1 and DP_2 have the same number of records and the same count on their records, i.e., $M = 8$, and $F = 4$, but they have different information gain $DP_1 = 0.011580$ and $DP_2 = 0.251629$ on the Sex attribute. Since the data providers are not consistent in providing the same information on the common $RecIDs$, this results in a change in the count for class label values. For instance, DP_1 indicates that there is 1 female whose loan is approved, whereas DP_2 indicates 0 females. ■

4) Computation of trust score: Intuitively, the trust score is a metric for assessing the trustworthiness of each data provider. We compute the trust score TS_{DP_i} locally for each data provider in an iterative manner on each attribute $ReqA_x$ from

TABLE II: Example data of numerical type attribute

Data Provider DP_1		
ID	**Age**	**Loan approval**
1	39	N
2	50	N
3	38	N
4	53	N
5	28	N
6	37	N
7	49	N
8	59	N
9	31	Y
10	42	Y
11	37	Y
12	30	Y

Raw data table (a)

Data Provider DP_3		
ID	**Age**	**Loan approval**
1	45	N
2	50	N
3	35	N
4	57	N
5	28	N
6	37	N
7	49	N
8	66	N
9	27	Y
10	42	Y
11	38	Y
12	28	Y

Raw data table (b)

Discretization		
ID	**Age**	**Loan approval**
1	[34.0 - 46.0]	N
2	[46.0 - 58.0]	N
3	[34.0 - 46.0]	N
4	[46.0 - 58.0]	N
5	[22.0 - 34.0]	N
6	[34.0 - 46.0]	N
7	[46.0 - 58.0]	N
8	[58.0 - 70.0]	N
9	[22.0 - 34.0]	Y
10	[34.0 - 46.0]	Y
11	[34.0 - 46.0]	Y
12	[22.0 - 34.0]	Y

Raw data table (c)

TABLE III: Compressed data table for categorical type attribute

Data Provider DP_1		
Sex	**Loan approval**	**#of Recs.**
M	3Y5N	8
F	1Y3N	4
	Total	12

Raw data table (a)

Data Provider DP_2		
Sex	**Loan approval**	**# of Recs.**
M	4Y4N	8
F	0Y4N	4
	Total	12

Raw data table (b)

the CS. γ is a user-defined weight. A data provider qualifying on the majority gains a positive γ weight in the trust score. On the other hand, a disqualified data provider is penalized with a negative -γ weight in the trust score. We aggregate on both positive and negative weights at each iteration to determine the final trust score for each data provider.

$$TS_{DP_i} = \sum_{ReqA_x \in ReqA} \gamma \begin{cases} if(Cand \in Maj_{Cand}) & +\gamma \\ if(Cand \notin Maj_{Cand}) & -\gamma \end{cases} \tag{7}$$

B. Security properties

In this section, we discuss the security properties of our proposed algorithm.

1) Security against covert adversaries: In the context of our problem, a dishonest data provider is a kind of covert adversary who may arbitrarily provide false data on his attribute $A_{\mathcal{J}} \in \mathcal{PA}_i$. The probability of detecting this cheat by our proposed trust computation algorithm is $1 - \xi$ (refer to the Section V-C1 for details). Each DP_i who has committed to, when registering, the available attributes $\mathcal{PA}_i = \{A_1, \ldots, A_d\}$ is responsible to answer the CS's challenge request, where $\exists ReqA_x{}' \in \mathcal{PA}_i$. When the CS detects a data provider cheating, the provider is penalized with a negative -γ weight in the trust score.

2) Mutual authentication: Before the verification process, each DP_i and the CS mutually authenticate each other by the TLS 1.2 protocol or higher [18], [65]. It is indispensable for the CS to negotiate on the latest stable version of the TLS protocol and stronger cipher suite to prevent against different forms of deception. After successful authentication of each DP_i, they are granted access to the resource queue, where they can register their data attributes.

3) Minimal access for outsourcing verification: The data providers who own customers' private data outsource the verification on their data to the CS. Each DP_i computes locally the information gain function $\mathcal{G}$ on an available attribute $A_{\mathcal{J}} \in \mathcal{PA}_i$, whereas the CS can have access to only an encrypted $\mathcal{G}'_{A_{\mathcal{J}}}$ message, i.e., ψ, and its keyed hash, i.e., Υ for the verification. It benefits the data providers to restrict the CS from accessing the customers' private data. Since encrypted individual data records are not exchanged during the verification, the overhead of computation on the CS is also reduced.

4) Authentication and integrity: HMAC enforces integrity and authenticity. It depends on what underlying hashing function has been used. There are some collision-related vulnerabilities of MD5; however, HMAC-MD5 is not as affected by those vulnerabilities. Regardless, SHA-2 is cryptographically stronger than MD5 and SHA-1. HMAC is constructed by using two nested keys, say k_{in} and k_{out}. These nested keys are not independent; instead they are derived from a single k_h. Let $\mathcal{M}$ bytes be assumed to be the message blocks for the underlying Merkle-Damgard hash. To derive the keys k_{in} and k_{out}, which are byte strings of length $\mathcal{M}$, we first construct k_h exactly $\mathcal{M}$ bytes long. If the length of $k_h \leq \mathcal{M}$, we pad it out with zero bytes; otherwise, we replace it with $\mathcal{H}(k_h)$ padded with zero bytes. Then we compute

$$k_{in} \leftarrow k_h \oplus ipad$$
$$k_{out} \leftarrow k_h \oplus opad$$

The ipad denotes the *inner* pad and the opad denotes the *outer* pad. These pads are 512 bit constants that never change and are embedded in the implementation of HMAC. HMAC is assumed to be a secure PRF [14]. It provides better protection against length extension attacks. It is built as follows:

$$\mathcal{S}(k_h, \psi) = \mathcal{H}\Big(k_h \oplus opad, \mathcal{H}\Big(k_h \oplus ipad \| \psi\Big)\Big)$$

One of the properties of a cryptographic hash function is that if there is a minor change in an input message, it changes the message digest so extensively that the new message digest appears uncorrelated with the old computed message digest. In our case, we do not apply cryptographic hash functions directly on the input data for data integrity because we allow

Algorithm 1 *IEB_Trust*

Key Setup: CS and DP_i derive n symmetric keys by mutual authentication protocol
Input: Data consumer attributes request $ReqA_1, \ldots, ReqA_m$, privacy budget ϵ
Input: Data provider DP_n's attributes $A_1, \ldots, A_d$
Output: Accepted DP_n

$DP_1, \ldots, DP_n$ own private data tables $D_1, \ldots, D_n$ $\forall i \in I$, where $I = 1, \ldots, n$;
Each DP_i holds set of attributes $\mathcal{PA}_i = \{A_1, \ldots, A_d\}$, over a domain of attributes request $ReqA = \{ReqA_1, \ldots, ReqA_m\}$;

$TS_{DP_i} \leftarrow 0$; /* Initially, trust score is set to 0 for each data provider */
$s_t \leftarrow 0$; /* Initially, arrival sequence is set to 0 for all data providers' attributes */
$\epsilon' \leftarrow \frac{\epsilon}{|ReqA|}$;
while $\exists ReqA_x \in ReqA$ **do**
  for $i \in I$ **do**
    if $\exists ReqA_x \in \mathcal{PA}_i$ **then**
      register arrival sequence s_t on each attribute;
    end if
  end for
end while
Round 1
while $\exists ReqA_x \in ReqA$ **do**
  CS randomly picks $ReqA_x{}'$ over a range of $ReqA_1, \ldots, ReqA_m$ without replacement;
  CS sends challenge $E(ks_i, ReqA_x{}')$ to each DP_i where $\exists ReqA_x{}' \in \mathcal{PA}_i$;
  Each DP_i computes $\mathcal{G}^{(1)}_{A_\mathcal{J}}$ according to Eq. (4) and then adds $\mathtt{Lap}(1/\epsilon')$, to have $\mathcal{G}^{(1)'}_{A_\mathcal{J}}$;
  Each DP_i encrypts the message $\psi^{(1)} \leftarrow E(ks_i, \mathcal{G}^{(1)'}_{A_\mathcal{J}})$ and then computes tag $\Upsilon^{(1)} \leftarrow \mathcal{S}(k_h, \psi^{(1)})$;
  CS receives $\psi^{(1)} \| \Upsilon^{(1)} \| DP_i$ on his challenge from the corresponding data providers;
  CS computes comparison to determine $Maj^{R(1)}_{Cand} \leftarrow$ findMajCand($\psi^{(1)} \| \Upsilon^{(1)}$, $size$);
end while
Round 2
while $\exists ReqA_x \in ReqA$ **do**
  for $\ell = 1$ to α **do**
    CS generates $\mathcal{K}$ random IDs for $ReqA_x{}'$ (pick in Round 1) from $|D_i|$ records, where $5 \leq \mathcal{K} \leq 10$;
    CS generates $\mathcal{P}$ pairs of values for $ReqA_x{}'$ and A^{cls} attributes, where $5 \leq \mathcal{P} \leq 10$;
    CS sends challenge $E(ks_i, \mathcal{K}, ReqA_x{}') \| E(ks_i, v_{x'}, v_{cls})$ to each DP_i where $\exists ReqA_x{}' \in \mathcal{PA}_i$;
    Each DP_i computes $\mathcal{G}^{(2)}_{A_\mathcal{J}}$ on the concatenated $\mathcal{K}$ specified records and $\mathcal{P}$ pairs of values and then adds $\mathtt{Lap}(1/\epsilon')$, to have $\mathcal{G}^{(2)'}_{A_\mathcal{J}}$;
    Each DP_i encrypts the message $\psi^{(2)} \leftarrow E(ks_i, \mathcal{G}^{(2)'}_{A_\mathcal{J}})$ and then computes tag $\Upsilon^{(2)} \leftarrow \mathcal{S}(k_h, \psi^{(2)})$;
    CS receives $\psi^{(2)} \| \Upsilon^{(2)} \| DP_i$ on his challenge from the corresponding data providers;
    CS computes comparison to determine $Maj^{R(2)}_{Cand_\ell} \leftarrow$ findMajCand($\psi^{(2)} \| \Upsilon^{(2)}$, $size$);
  end for
  CS computes $\bigcap_{\ell=1}^{\alpha} Maj^{R(2)}_{Cand_\ell}$;
end while
CS computes $Maj^{R(1)}_{Cand} \bigcap Maj^{R(2)}_{Cand}$ to determine Maj_{Cand};
for all $Cand \in Maj_{Cand}$ **do**
  set $Cand$ as $Qual_{DP_i}$;
  $TS_{DP_i} = TS_{DP_i} + \gamma$;
end for
for all $Cand \notin Maj_{Cand}$ **do**
  set $Cand$ as $UnQual_{DP_i}$;
  $TS_{DP_i} = TS_{DP_i} - \gamma$;
end for
if $size == 1 \wedge TS_{DP_i} \geq 0$ **then**
  set DP_i as $Qual_{DP_i}$;
end if
Pick one $Cand$ by comparison on the arrival sequences of the $Qual_{DP_n}$ on each attribute;
return Data providers whose final aggregated trust score ≥ 0

parties to have minor inaccuracies on numerical attributes for a specified threshold.

C. Analysis

In this section, we analyze the correctness and security of Algorithm 1.

Proposition V.1. ***(Correctness)*** *Assuming multiple data providers are dishonest, Algorithm 1 correctly computes the trust scores among them, as stated in Problem 1 in Section II-B, to evaluate the trustworthiness of each data provider.*

Proof: Algorithm 1 selects an attribute uniformly at random without replacement from a list $ReqA = \{ReqA_1, \ldots, ReqA_m\}$ of m requested attributes. Each DP_i computes $\mathcal{G}_{A_\mathcal{J}}$ according to Eq. (4) for its matching attribute in the presence of a shared class attribute A^{cls}. For a continuous-valued attribute, each provider follows equal-width method for discretization into intervals of equal size. Consider $A_\mathcal{J}$ is discrete-valued, owned by two providers, where $\Omega(A_\mathcal{J}) = \{v_1, v_2\}$ is in its domain of data values. Assume there is a single record between two providers where they have different values. Algorithm 1 computes $\mathcal{G}^{(1)'}_{A_\mathcal{J}}$ in the first round for both the data providers and returns different scores. This suggests that they are not the same.

Now, we consider an extended case where two data providers (say DP_1, DP_2) would have different sets of records but the computation of $\mathcal{G}^{(1)'}_{A_\mathcal{J}}$ in the first round on the full dataset for both data providers returns the same score, so we have $Maj^{R(1)}_{Cand} = \{DP_1, DP_2\}$. Algorithm 1 verifies further by running the process α times in the second round. During each iteration data providers have to select records over $\mathcal{K}$ random IDs for $A_\mathcal{J}$, and they also have to add $\mathcal{P}$ pairs of values $v_{x'}$ and v_{cls} for $A_\mathcal{J}$ and a class attribute A^{cls}, respectively, from the CS before computation of $\mathcal{G}^{(2)'}_{A_\mathcal{J}}$. Algorithm 1 computes $Maj^{R(1)}_{Cand} \cap (\bigcap_{\ell=1}^{\alpha} Maj^{R(2)}_{Cand_\ell})$ to determine Maj_{Cand}. This determines whether or not the data providers are holding the same data values over the common attribute $A_{\mathcal{J}}$. Data providers are required to match in both the rounds to prove that they have the same score. Since data providers are holding a different set of records, it is not possible for them to match because of the randomness introduced in the second round. ∎

Proposition V.2. *(Security) Algorithm 1 is secure against covert adversaries as described in Section V-B1 by the probabilistic bound of* $1 - \xi$.

Proof: The security of Algorithm 1 depends on the keys derivation in the mutual authentication protocol and the communication of the cloud server CS and data providers DP_n in the verification process.

- A random challenge $E(ks_i, ReqA_x{}')$ is secure because of symmetric keys derivation by [18], [65].
- On a given challenge request, if $\exists ReqA_x{}' \in \mathcal{PA}_i$, each data provider first computes the information gain function on its matching attribute $\mathcal{G}_{A_{\mathcal{J}}} \in \mathcal{PA}_i$, and then perturbs the output by adding noise. This returns a noisy score $\mathcal{G}'_{A_{\mathcal{J}}}$ for which data providers should agree on the scale for digits after the decimal point. It is secured for privacy protection because each DP_i only exchanges an encrypted $\mathcal{G}'_{A_{\mathcal{J}}}$ message, i.e., ψ, and its keyed hash, i.e., Υ, with the CS in both rounds of the protocol, instead of exchanging encrypted individual data records on their attributes $A_{\mathcal{J}}$.
- Keyed hash-based message authentication code $\mathcal{S}(k_h, \psi)$ is a secure PRF according to [14]. It is computationally infeasible for an adversary to find distinct inputs ψ_1, ψ_2 such that $\mathcal{S}(k_h, \psi_1) = \mathcal{S}(k_h, \psi_2)$.
- Dishonest data providers cannot modify the outputs, i.e., $\psi \| \Upsilon$, of the honest providers in any round of the protocol. They may compute $\mathcal{G}^*_{A_{\mathcal{J}}}$ on their false data and can send their $\psi^* \| \Upsilon^*$ to the CS. The CS computes a comparison and detects cheating from a dishonest data provider with the probability of $1 - \xi$.

∎

1) Adversary's inferences: In the following, we estimate the probability of an adversary, i.e., a dishonest data provider, to correctly guess $\mathcal{G}^*_{A_{\mathcal{J}}}$ on a random challenge attribute $ReqA'_x$. An adversary knows $|D^\tau|$, the number of records in D^τ, and $|A^{cls}_{i,D^\tau}|$, the number of records of class A^{cls}_i in D^τ, and computes the entropy of D^τ by Eq. (2). Next, the adversary may try to compute entropy on $A_{\mathcal{J}}$ by the following equation because he knows $|\Omega(A_{\mathcal{J}})|$, the domain size of $A_{\mathcal{J}}$, and $|D^\tau|$, the number of records in D^τ.

$$E^*_{A_{\mathcal{J}}}(D^\tau) = \sum_{j'=1}^{\mathcal{V}'} \frac{|D^\tau_{j'}|}{|D^\tau|} \times - \sum_{i=1}^{\mathcal{C}} \frac{|A^{cls}_{i,D^\tau_{j'}}|}{|D^\tau_{j'}|} \times log_2 \frac{|A^{cls}_{i,D^\tau_{j'}}|}{|D^\tau_{j'}|} \quad (8)$$

There are $|\Omega(A_{\mathcal{J}})|^{|D^\tau|}$ possible arrangements in which an adversary may try to compute $E^*_{A_{\mathcal{J}}}(D^\tau)$. Finally, he computes $\mathcal{G}^*_{A_{\mathcal{J}}}$ having all distinct values by the following equation.

$$\mathcal{G}^*_{A_{\mathcal{J}}} = E(D^\tau) - E^*_{A_{\mathcal{J}}}(D^\tau) \quad (9)$$

This results in ϑ distinct values of $\mathcal{G}^*_{A_{\mathcal{J}}}$, with the lower bound of $\vartheta \approx |D^\tau|$. The probability of correctly guessing $\mathcal{G}^*_{A_{\mathcal{J}}}$ for an adversary in our verification process is

$$\xi = \frac{1}{\vartheta} \times (\frac{1}{\vartheta})^\alpha \quad (10)$$

2) Detecting cheat against varying dishonest providers: Let n denote the number of participating data providers, and let b denote an upper bound on the number of dishonest data providers who may arbitrarily provide incorrect data in responding to the CS's challenge.

- When $b < n/2$, the verification process guarantees fairness and no honest data providers are negatively affected by their trust levels.
- When $b \leq n - 2$, the verification process guarantees fairness under the arbitrary behavior of dishonest data providers, where the chance of detecting them is $1 - \xi$. It is a type of covert adversarial behavior when the dishonest data providers arbitrarily provide false data on their data inputs, i.e., they neither would be able to appear in the majority nor would be able to undermine the reputations of the honest data providers.
- When $b > n/2$, the verification process does not guarantee fairness on the flip side, i.e., when the behavior of dishonest data providers is not arbitrary. This would be the case when the dishonest data providers not only appear in the majority but also organize in a way to undermine the reputation of the honest data providers. We assume that if a secure set intersection is carried out by using a trusted mediator (e.g., by computing the function on the data providers input) between data providers, then the dishonest providers would not be able to determine the total number of participating data providers in advance. This would restrict them from developing the organized group; still, there is no remedy if they would try by guessing at random.

D. Evaluation of learner models

We provide an example of a sample data to evaluate the quality of linear regression, k-nearest neighbors (kNN), and random forest learner models.

Example 4. We retrieve the top 1000 records from a real-life *Adult*[1] dataset on attributes *age, education-num, race, sex, income*. The attributes *age, education-num* are of continuous types, whereas *race, sex, income* are of categorical types. We develop learner models in *RapidMiner*[2] to compare the predictive accuracy of linear regression, k-nearest neighbors (kNN), and random forest methods.

For the linear regression model, we set *education-num* as a label, which is considered as a dependent attribute (or variable), and the remaining are considered as independent attributes. We convert non-numeric type attributes to the numeric type. After running 10-fold cross-validation, the *Root Mean Square Error (RMSE)* is found to be 2.438 ± 0.165, which

[1] Available at: http://archive.ics.uci.edu/ml/datasets/Adult.
[2] Available at: https://rapidminer.com/products/studio/

indicates the standard deviation of the residuals. Furthermore, R^2 is found to be 0.127 ± 0.055, which indicates the goodness of fit of this regression model. Its value is close to 0, indicating a weak linear correlation.

For the k-nearest neighbors (kNN) model, we set all attributes as nominal and *education-num* as a label. After running 10-fold cross-validation when $k = 20$, the *accuracy* is found to be $33.90\% \pm 5.59\%$, which indicates the percentage of correct predictions.

For the random forest model, we set the *education-num* attribute as nominal and specify the role as a label. The key parameter 'number of trees' is specified as 10, and the 'gain ratio' is chosen as a criterion for splitting attributes. After running 10-fold cross-validation, the *accuracy* is found to be $32.90\% \pm 0.30\%$, which indicates the percentage of correct predictions. ■

There are no significant performance differences found on running these learner models on the sample dataset. Data providers would use any one or multiple learning methods for missing data imputation.

E. Price setting using auction mechanism

An auction mechanism can be defined in many different ways depending upon the design requirements. The two variants of 2nd price sealed-bid auctions [23] have widely used, namely Vickrey-Clarke-Groves (VCG) and Generalized Second Price (GSP) mechanisms for multiple items.

The reason for employing the VCG mechanism for determining the pricing on data providers' attributes is that truthful bidding is a dominant strategy, and there is no incentive to lie or deviate from reporting true valuations for a data provider. It maximizes the total valuation obtained by data providers. One nice property of the VCG mechanism is that it provides a unique outcome, which is socially optimal, whereas, in the GSP there would be multiple outcomes in terms of Nash equilibrium. One Nash equilibrium would maximize social welfare but not all of them.

We intend to design an auction mechanism for multiple items. It is assumed that the data providers intend to set up a matching market using a 2nd price sealed-bid auction for valuation of their attributes. We formally define the procedure for setting the price as follows:

1) Data providers: Let $DP_1, \ldots, DP_n$ (where $i = 1, \ldots, n$) be the set of data providers who set up a matching market for valuations of their attributes.

2) Positions: Let $P_1, \ldots, P_n$ (where $j = 1, \ldots, n$) be the set of positions for which data providers compete. The higher the position P_j, the more will be its demand rate. The positions should be equal to the number of data providers. If there are more data providers than positions, we simply add fictitious positions of demand rate 0. Similarly, if there are more positions than data providers, we add fictitious data providers of revenue per demand 0.

3) Revenue per demand: Revenue per demand is the expected amount of money that a data provider DP_i receives, denoted by Rev_i, for every demand on its attribute. The monetary values of Rev_i are sorted in descending order.

4) Demand rate: Demand rate is defined as the number of demands requested by a consumer over a period of time, denoted by Q_j. Demand rate varies as per the position P_j. Q_j enumerates in descending order.

5) Data providers' valuations: Data providers' valuations are defined as the data provider DP_i's valuation of the position P_j. It is the product of the revenue per demand Rev_i and the demand rate Q_j, denoted by $Val_{i,j}$. It is computed as follows:

$$Val_{i,j} = Rev_i \times Q_j \tag{11}$$

6) VCG price: VCG price is defined as the harm or externality caused by data provider DP_i to other data providers in terms of reduction of their valuations due to his presence. It is called VCG price, denoted by $ExPrc_{i,j}$, which is paid by data provider DP_i for position P_j. Formally, it is defined by

$$ExPrc_{i,j} = \bigvee_{DP_n - DP_i}^{P_n} - \bigvee_{DP_n - DP_i}^{P_n - P_j} \tag{12}$$

where

- $DP_n - DP_i$ is the set of data providers excluding data provider DP_i;
- $P_n - P_j$ is the set of positions excluding position P_j;
- $\bigvee_{DP_n - DP_i}^{P_n}$ is the sum of data provider values of an optimal matching between sets $DP_n - DP_i$ and P_n; and
- $\bigvee_{DP_n - DP_i}^{P_n - P_j}$ is the sum of data provider values of an optimal matching between sets $DP_n - DP_i$ and $P_n - P_j$.

7) Data providers' valuations after payoff: Data providers' valuations after payoff is defined as the data provider DP_i's valuation on position P_j after paying off harm to other data providers. It is calculated using the following equation.

$$Val_{DP_i} = \max Val_{i,j} - ExPrc_{i,j} \tag{13}$$

8) Valuation of an attribute: Valuation of an attribute can be assessed once a data provider DP_i acquires a certain position P_j. The value of each data provider's attribute per single demand is calculated using the following equation.

$$ValAttr_{DP_i} = \frac{Val_{DP_i}}{Q_j} \tag{14}$$

9) Attribute count: The attribute count $CntAttr_{DP_i}$ of a data provider DP_i represents the number of attributes in a single record. Each DP_i owns a mutually exclusive set of attributes.

10) Price per record: The price per record $PrcRec_{DP_i}$ of a data provider DP_i represents the unit price of a record. Naturally, it is the product of the value per attribute $ValAttr_{DP_i}$ and the attribute count $CntAttr_{DP_i}$ in a single record. That is,

$$PrcRec_{DP_i} = ValAttr_{DP_i} \times CntAttr_{DP_i} \tag{15}$$

11) Size of dataset: The dataset of each data provider DP_i consists of a collection of records, denoted by $|D_i|$. The size of a dataset grows as the number of records in the dataset increases.

12) Price of raw dataset: The price of a raw dataset $PrcRawDS_{DP_i}$ represents the data provider DP_i's selling price of a raw dataset in the e-market. The overall pricing of a raw dataset increases as the number of records or the unit *price per record* increases. It is computed as follows:

$$PrcRawDS_{DP_i} = |D_i| \times PrcRec_{DP_i} \tag{16}$$

13) Total price of raw dataset: The total price of the raw dataset $TPrc_{RawDS}$ is the sum of the pricing of all the contributing data providers' raw datasets. It is computed as follows:

$$TPrc_{RawDS} = \sum_{i=1}^{n} PrcRawDS_{DP_i} \tag{17}$$

14) Total monetary value of raw dataset: First, data providers compute baseline accuracy (BA) for classification analysis using the secure multiple party classifier [21] by maintaining the confidentiality of their raw data. Then they use the information utility of classifying raw data to derive the monetary value of the raw dataset, denoted by $TMValue_{RawDS}$. It is calculated using the following equation:

$$TMValue_{RawDS} = TPrc_{RawDS} \times BA \tag{18}$$

F. Anonymization method

In this section, we provide an extension of the two-party ***Diff***erentially private anonymization in Algorithm 2, which is based on ***Gen***eralization [53] to differentially integrate multiple private data tables. This algorithm guarantees ϵ-differential privacy and security definition under the semi-honest adversary model (readers may refer to the detailed analysis in [53], Section 6.3). The two major extensions over the TDS algorithm [31] include: (1) ***DistDiffGen*** selects the $Best$ specialization based on the exponential mechanism, and (2) ***DistDiffGen*** perturbs the generalized contingency table by adding the Laplacian noise to the count of each equivalence group.

Generally, there is no incentive for any data provider who executes the algorithm as the purpose is merely to synchronize the anonymization process. We assume a trusted data provider, who attains the highest trust score after running the Algorithm 1, starts the anonymization process. The accepted data providers, as a result of trust computation by Algorithm 1, attain a mutually exclusive set of attributes, i.e., $\mathcal{PA}_i \cap \mathcal{PA}_j = \emptyset$ for any $1 \leq i, j \leq n$ over the same set of records for integrating data.

Initially, all values in the set of attributes $\mathcal{PA}_i = \{A_1, \ldots, A_d\}$ of each data provider are generalized to the topmost value in their taxonomy trees (Line 1), as illustrated in Fig. 1, and $Mark_\kappa$ contains the topmost value for each attribute $A_\mathcal{J} \in \mathcal{PA}_i$ (Line 2). Each data provider keeps a copy of the $\cup Mark_\kappa$ and a generalized data table D_g. The attribute $A_\mathcal{J}$ can be either categorical or numerical, but the class attribute is required to be categorical. The split value of a categorical attribute v_c is a generalized value drawn from a pre-defined taxonomy tree of the attribute, whereas the split value of a numerical attribute v_{num} is determined by using the exponential mechanism (Line 4). It partitions the domain range of a numerical attribute into successive intervals $\mathcal{I}_1, \ldots, \mathcal{I}_k$. Line 4 preserves $\epsilon'|A_{num}|$-differential privacy since the cost of each exponential mechanism is ϵ'. In Line 5, a score $IGScore$ is computed for all candidates $v \in \cup Mark_\kappa$. At each iteration, the algorithm uses the secure distributed exponential mechanism (DistExp) as presented in [53] (readers may refer to the details of the DistExp algorithm) to select a winner candidate $w \in \cup Mark_\kappa$ for specialization (Line 7). Different utility functions (e.g., information gain) can be used to calculate the score. If the winner candidate w is local to DP_i, DP_i specializes w on D_g by splitting its records into child partitions, updates its local copy of $\cup Mark_\kappa$, and instructs all the other participating data providers to specialize and update their local copy of $\cup Mark_\kappa$ (Line 8-11). The information gain, denoted by $\widetilde{\mathcal{G}}_{DP_i}$, accumulates $IGScore(x)$ on the winner's attribute specializations (Line 12). DP_i further calculates the scores of the new candidates as a result of the specialization (Line 14). If the winner w is not one of DP_i's candidates, DP_i waits for instructions from the other winner data provider DP_j, where $i \neq j$, to specialize w and to update its local copy of $\cup Mark_\kappa$ (Lines 16 and 17). This process iterates until the specified number of the specializations h is reached. The algorithm perturbs the output by adding the noisy count at each leaf node (Line 21) using the Laplace mechanism. The contribution of each data provider is computed according to Eq. (22). Finally, the monetary share of each data provider is derived according to the Eq. (23).

G. Quantifying the monetary value

The rationality of quantifying the monetary value is that data providers are the business stakeholders who collaborate in the data integration process to maximize their profits. The profit generated by their collaboration is distributed based on each provider's contribution to information utility and its trustworthiness.

1) Cost of anonymization in integrated data: First, the data providers compute classification accuracy (CA) on the anonymized integrated data. Then, they quantify the cost of anonymization in integrated data, denoted by $Cost_{IntDS}$, on the difference between the baseline accuracy (BA) and the classification accuracy (CA). It is computed as follows:

$$Cost_{IntDS} = TPrc_{RawDS} \times (BA - CA) \tag{19}$$

2) Expected value in integrated data: An expected monetary value in integrated data is what the data providers earn from the information utility of classification analysis when trading an anonymized version of integrated data. The information utility varies with the valuations of data providers' attributes and joint privacy requirements, such as privacy budget ϵ and specialization level h, for a *ϵ-differential privacy* model in a distributed setup, between the data providers. It is calculated on the difference between the total monetary value of the raw dataset $TMValue_{RawDS}$ and the cost of anonymization in integrated data $Cost_{IntDS}$. It is computed as follows:

$$EValue_{IntDS} = TMValue_{RawDS} - Cost_{IntDS} \tag{20}$$

Algorithm 2 Monetary Shares for Data Providers using Dist-DiffGen

Input: Data providers' attributes valuations $ValAttr_{DP_n}$
Input: Private data tables $D_1, \ldots, D_n$, privacy budget ϵ, and number of specializations h
Output: Monetary shares $MShare_{DP_n}$

```
Initialize D_g with one record containing topmost generalized values in each data provider's taxonomy tree;
Initialize Mark_κ to include the topmost value;
ε' ← ε / (2(|A_num| + 2h));
Determine the split value for each v_num ∈ ∪Mark_κ with probability ∝ exp(ε'/(2Δu) u(D, v_num));
Compute the IGScore for ∀v ∈ ∪Mark_κ;
for iter = 1 to h do
    Determine the winner candidate w by using the DistExp Algorithm [53];
    if w is local then
        Specialize w on D_g;
        Replace w with child(w) in the local copy of ∪Mark_κ;
        Instruct all the other participating data providers to specialize and update ∪Mark_κ;
        G̃_{DP_i} = G̃_{DP_i} + IGScore(x);
        Determine the split value for each new v_num ∈ ∪Mark_κ with probability ∝ exp(ε'/(2Δu) u(D, v_num));
        Compute the IGScore for each new v ∈ ∪Mark_κ;
    else
        Wait for the instruction from the winner data provider;
        Specialize w and update ∪Mark_κ using the instruction;
        G̃_{DP_j} = G̃_{DP_j} + IGScore(x);
    end if
end for
Compute count (CT + Lap(2/ε)) for each leaf node;
Compute the contribution of each data provider according to Eq. (22);
Compute monetary share of each data provider according to Eq. (23);
return MShare_{DP_n}
```

3) Expected value of an individual data provider: The expected monetary value of an individual data provider, denoted by $EValueIndv_{DP_i}$, is determined by the ratio of the number of attributes $CntAttr_{DP_i}$ a data provider owns with the total count of attributes. It is computed as follows:

$$EValueIndv_{DP_i} = EValue_{IntDS} \times \frac{CntAttr_{DP_i}}{\sum_{i=1}^{n} CntAttr_{DP_i}} \tag{21}$$

4) Derivation of monetary share: The derivation of a monetary share depends upon the contribution of each data provider and its trustworthiness. Intuitively, a data provider whose provided data on his attributes result in more information gain, and whose trust level is higher than the other competitors, can get a significantly larger share of the monetary value. The contribution of each data provider DP_i is derived from the expected monetary value $EValueIndv_{DP_i}$ by fairly computing first the accumulative information gain $\widetilde{\mathcal{G}}_{DP_i}$ of each data provider DP_i on the anonymized integrated dataset. The information gain $IGScore(x)$ of the winner candidate w data provider accumulates under the relevant winner w data provider at each iteration (refer to the Section V-F for details) for the specified specialization level h. The contribution of each data provider $Contrib_{DP_i}$ is calculated using the following equation:

$$Contrib_{DP_i} = \frac{\widetilde{\mathcal{G}}_{DP_i}}{\sum_{i=1}^{n} \widetilde{\mathcal{G}}_{DP_i}} \times EValueIndv_{DP_i} \tag{22}$$

Finally, the monetary share of each data provider $MShare_{DP_i}$ is derived according to Eq. (7), i.e., the aggregated trust score of each data provider, and Eq. (22), i.e., the contribution of each data provider. Therefore, $MShare_{DP_i}$ becomes:

$$MShare_{DP_i} = Contrib_{DP_i}\left(1 + \frac{TS_{DP_i}}{\sum_{i=1}^{n} TS_{DP_i}}\right) \tag{23}$$

VI. Comparative analysis and empirical study

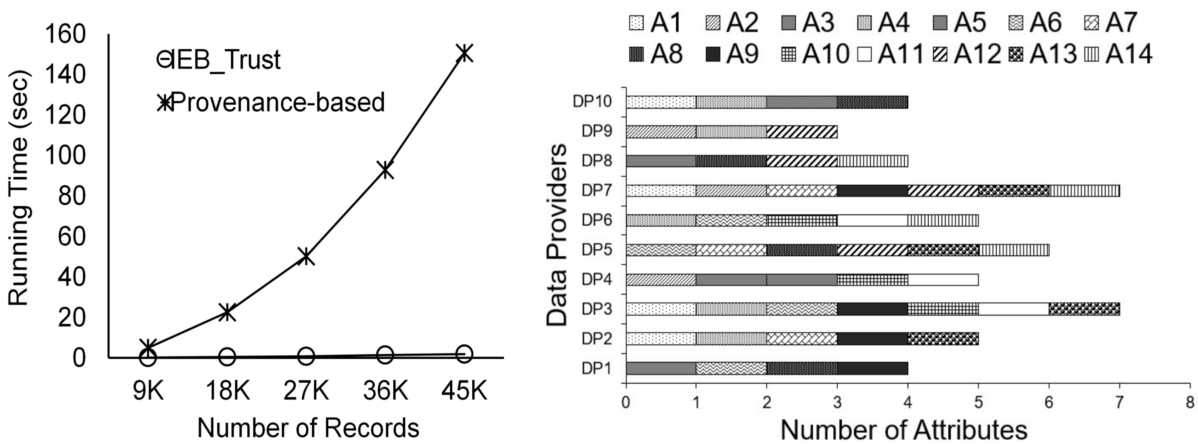
In this section, we first provide a comparison of our approach, followed by an empirical study.

A. Comparative analysis

We compare our proposed *IEB_Trust*, an entropy-based trust computation algorithm with the closely related provenance-based trust method [16]. The provenance-based method computes the trust scores for data and data providers using similarity functions, but do not consider privacy protection when evaluating trustworthiness. The fundamental idea of our approach is different. Our method enables secure trustworthiness assessment and preserves the privacy of the customers' data when evaluating the trustworthiness of the participating data providers. For this reason, we are limiting to the runtime comparison in Fig. 3a. We evaluate the performance of our proposed method on a real-life *Adult*[3] dataset. It contains $45,222$ records with 8 categorical attributes, 6 numerical attributes, and a binary class attribute $Income$ with two levels, $\leq 50K$ or $> 50K$. The distribution of attributes other than class attribute among 10 data providers is shown in Fig. 3b. We generate 10% of data conflicts over randomly chosen attributes. We vary the size of the datasets $|D_i|$ from $10K$ to $50K$ to study the runtime cost. All experiments are conducted on an Intel Core i7 3.4GHz PC with 8GB memory.

The running time includes time elapsed in both the initialization phase and the iteration phase. We observe that the initialization phase of the provenance-based method takes more time to compute data similarity and data conflict. It has worst-case complexity of $O(n^2)$. While the complexity of our proposed method at the initialization phase is $O(CntAttr_{DP_i} \cdot |D_i| \log |D_i|)$. Since each data provider computes $\mathcal{G}_{A_{\mathcal{J}}}$ in a distributed setup, the complexity remains the same in our method. The iteration phase to compute trust is much faster in both the methods. It takes less than one second to complete the trust computation. Fig. 3a shows that our method is more efficient in running time over the provenance-based method. Our method is scalable when we need to grow either the number of attributes, the number of data providers, or both on a dataset.

[3]Available at: http://archive.ics.uci.edu/ml/datasets/Adult.



(a) Runtime comparison (b) Distribution of attributes

Fig. 3: Our method improves the runtime efficiency compared to the provenance-based trust method.

B. Empirical study

We first analyze the trustworthiness of each data provider and assess the truthfulness of the provided data by a trust score metric. Second, we analyze the impact of ϵ*-differential privacy* requirements along with the aggregated trust score on each data provider's monetary value. We evaluate our proposed method, *IEB_Trust*, with the assumption of having 4 data providers who intend to verify the correctness of their data before participation in the data mashup. This assumption is reasonable because we have a limited number of attributes in the dataset to be shared among data providers.

1) Trust measurement: Our proposed method evaluates the trust of participating data providers based on the following conditions: (1) A data provider is found as honest and gains a positive score; (2) A data provider is found as dishonest and is penalized with a negative score; (3) A single data provider of an attribute that no others own is accepted based on the existing trust score $TS_{DP_i} \geq 0$ without an increase in the trust score; and (4) A data provider who does not register for an attribute has no effects on the trust score.

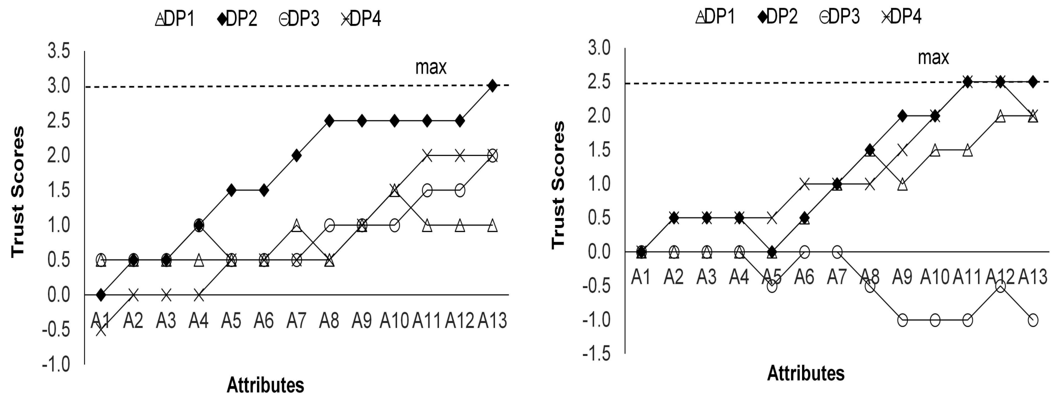
To demonstrate the effectiveness of our approach, we conduct two cases of experiments that are independent of each other. This means that for each case data providers hold different sets of overlapping attributes with their arrival sequences. In each case, we assume $\gamma = 0.5$, but it does not need to be fixed to a specific weight.

Consider the first case with the participating data providers' attributes and their arrival sequences. $DP_1 \mapsto A_1{:}s_{t_1}$, $A_7{:}s_{t_1}$, $A_8{:}s_{t_1}$, $A_9{:}s_{t_1}$, $A_{10}{:}s_{t_2}$, $A_{11}{:}s_{t_1}$; $DP_2 \mapsto A_2{:}s_{t_2}$, $A_3{:}s_{t_1}$, $A_4{:}s_{t_1}$, $A_5{:}s_{t_2}$, $A_7{:}s_{t_2}$, $A_8{:}s_{t_3}$, $A_{13}{:}s_{t_1}$; $DP_3 \mapsto A_1{:}s_{t_2}$, $A_4{:}s_{t_2}$, $A_5{:}s_{t_1}$, $A_6{:}s_{t_1}$, $A_8{:}s_{t_2}$, $A_{11}{:}s_{t_2}$, $A_{13}{:}s_{t_2}$; and $DP_4 \mapsto A_1{:}s_{t_3}$, $A_2{:}s_{t_1}$, $A_5{:}s_{t_3}$, $A_9{:}s_{t_2}$, $A_{10}{:}s_{t_1}$, $A_{11}{:}s_{t_3}$, $A_{12}{:}s_{t_1}$. Fig. 4a depicts the trust scores analysis for Case 1 based on the demand of a data consumer on attributes $A_1, \ldots, A_{13}$.

It is observed that the DP_2 trust score never drops during the verification process in contrast to the other competing data providers. The flat lines from A_2 to A_6 at trust score level 0.5, and A_9 to A_{12} at trust score level 2.5, indicate that those attributes are not submitted by DP_1 and DP_2, respectively. This is not always the case; for instance, there are flat lines from A_2 to A_3 at trust score level 0.5, A_5 to A_6 at trust score level 0.5, and A_{11} to A_{12} at trust score level 2.0, indicating that DP_2, DP_3, and DP_4 are the single data providers on those attributes. DP_2, DP_3, and DP_4 are accepted because they are maintaining an aggregated trust score ≥ 0 at that point of the verification. However, their trust scores do not increase because they own an attribute that no others own. It is assumed that DP_1 has 5% of missing data on A_8 and A_{11}, DP_3 has 5% of missing data on A_5, and DP_4 has 1% of missing data on A_1. They impute missing data by using the kNN imputation method in order to claim it as original data. Our trust verification approach restricts this dishonest behavior of data providers; for instance, DP_1 at A_8 and A_{11}, DP_3 at A_5, and DP_4 at A_1, by penalizing them with negative weight in their trust scores. Fig. 5a depicts the aggregated trust scores for Case 1. DP_2 attains the maximum trust score 3.0 in competing with the other data providers, whereas DP_1 ends up with the minimum trust score 1.0. There is a tie on aggregated trust scores between DP_3 and DP_4.

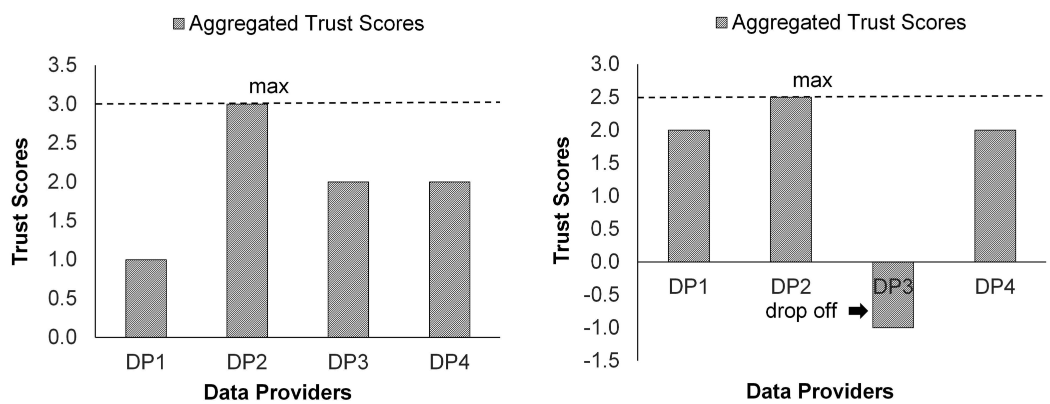
Consider the second case with the participating data providers' attributes and their arrival sequences. $DP_1 \mapsto A_1{:}s_{t_1}$, $A_6{:}s_{t_3}$, $A_7{:}s_{t_1}$, $A_8{:}s_{t_2}$, $A_9{:}s_{t_3}$, $A_{10}{:}s_{t_2}$, $A_{12}{:}s_{t_2}$; $DP_2 \mapsto A_2{:}s_{t_2}$, $A_5{:}s_{t_2}$, $A_6{:}s_{t_4}$, $A_7{:}s_{t_2}$, $A_8{:}s_{t_1}$, $A_9{:}s_{t_2}$, $A_{11}{:}s_{t_1}$; $DP_3 \mapsto A_3{:}s_{t_1}$, $A_5{:}s_{t_1}$, $A_6{:}s_{t_1}$, $A_8{:}s_{t_3}$, $A_9{:}s_{t_1}$, $A_{12}{:}s_{t_1}$, $A_{13}{:}s_{t_2}$; and $DP_4 \mapsto A_2{:}s_{t_1}$, $A_4{:}s_{t_1}$, $A_6{:}s_{t_2}$, $A_9{:}s_{t_4}$, $A_{10}{:}s_{t_1}$, $A_{11}{:}s_{t_2}$, $A_{13}{:}s_{t_1}$. Fig. 4b depicts the trust scores analysis for Case 2 based on the demand of a data consumer on attributes $A_1, \ldots, A_{13}$.

It is observed that DP_1, DP_2, and DP_4 maintain their trust scores quite well except for a fall of 0.5 in their trust scores at A_9, A_5, and A_{13}, respectively. The flat lines from A_1 to A_5 at trust score level 0.0, and A_3 to A_5 at trust score level 0.5, indicate that those attributes are not submitted by DP_1 and DP_4, except at A_1 and A_4, respectively. Since DP_1 and DP_4 are the single data providers on A_1 and A_4, their trust scores do not increase. However, they are accepted because they maintain an aggregated trust score ≥ 0. We observe that DP_3 is inconsistent in maintaining its trust level throughout



(a) *Case 1* (b) *Case 2*

Fig. 4: Trust scores analysis



(a) *Case 1* (b) *Case 2*

Fig. 5: Aggregated trust scores

the verification process. It is worthwhile to note that our trust verification process restricts the arbitrary behavior of dishonest DP_1 and DP_3 to undermine the trust levels of DP_2 and DP_4. Fig. 5b depicts the aggregated trust scores for Case 2. DP_2 attains the maximum trust score 2.5 in competing with the other data providers, whereas DP_3 ends up with a negative trust score of -1.0. This results in the rejection of DP_3 from the final selection in the data mashup.

2) Impact of privacy protection and trust score on DP's monetary value: In this section, we analyze the impact of ϵ-*differential privacy* requirements along with the aggregated trust score on each data provider's monetary value. Recall from Section V-E that both revenue per demand Rev_i and demand rate Q_j are enumerated in descending order. Suppose $Rev_i = \{\$0.6, \$0.5, \$0.4, \$0.3\}$ and $Q_j = \{9, 8, 7, 6\}$ for data providers DP_1, DP_2, DP_3, and DP_4, respectively. The inputs for Rev_i and Q_j do not need to be fixed to a particular value, it is just assumed here for simplicity.

Case 1 Table IV.(a) shows the selection of attributes from each accepted data provider. Baseline accuracy (BA) on the integrated data of accepted data providers is 85.3% using the secure multiple party classifier [21] without disclosing their raw data. We vertically partition the *Adult* dataset into four partitions VP_1, VP_2, VP_3, and VP_4 for data providers DP_1, DP_2, DP_3, and DP_4, respectively. Further, we split the dataset into $30,162$, and $15,060$ records for the training and testing set, respectively. The valuation of each data provider's attribute is \$0.47, \$0.41, \$0.36, and \$0.30, representing $ValAttr_{DP_1}$, $ValAttr_{DP_2}$, $ValAttr_{DP_3}$, and $ValAttr_{DP_4}$ by Eq. (14). The attribute count of each data provider is $CntAttr_{DP_1} = 3$, $CntAttr_{DP_2} = 4$, $CntAttr_{DP_3} = 3$, and $CntAttr_{DP_4} = 3$. The size of the dataset for each data provider $|D_i| = 45,222$.

TABLE IV: Selection of attributes from data providers

DP_1	DP_2	DP_3	DP_4
A_1	A_5	A_8	A_2
A_9	A_4	A_6	A_{12}
A_7	A_{13}	A_{11}	A_{10}
	A_3		

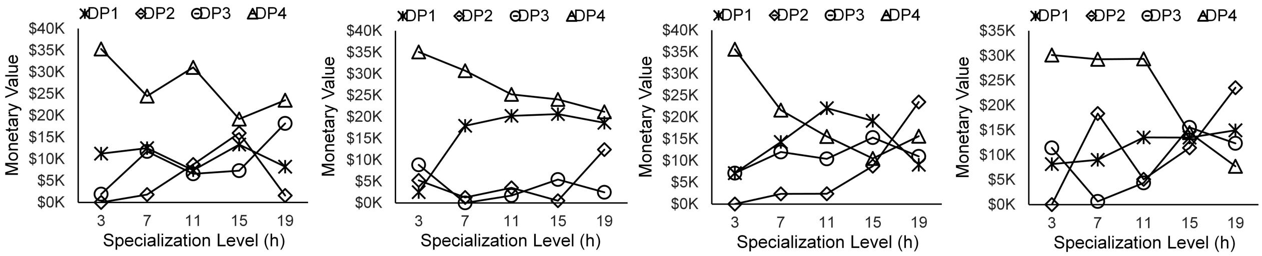
(a) Case 1

DP_1	DP_2	DP_4
A_1	A_8	A_2
A_7	A_9	A_4
A_{12}	A_{11}	A_{10}
		A_6

(b) Case 2

Fig. 6 depicts the impact of privacy protection and trust scores on DP_1, DP_2, DP_3, and DP_4's monetary value. ϵ-*differential privacy* is enforced with privacy parameters $\epsilon =$ 0.2, 0.4, 0.6, and 0.8 and specialization levels $3 \leq h \leq 19$.

Fig. 6a depicts the impact on DP_1, DP_2, DP_3, and DP_4's monetary value when the threshold is $\epsilon = 0.2$. We observe that DP_4 attains the highest monetary share due to more information utility and its aggregated trust score. When specialization



(a) $\epsilon = 0.2$ (b) $\epsilon = 0.4$ (c) $\epsilon = 0.6$ (d) $\epsilon = 0.8$

Fig. 6: Impact of ϵ-*differential privacy* requirements and Trust scores on DP_1, DP_2, DP_3, and DP_4 monetary value (Case 1)

level h increases from 3 to 7 and 11 to 15, DP_1, DP_2, and DP_3 get increases in their monetary shares, while DP_4's monetary share falls by approximately $\$11K$, though still achieving a higher share than other data providers. Initially, DP_2 has no monetary share when $h = 3$, but it increases with the increase in the specialization level h except when $h = 19$. DP_1, DP_2, and DP_3's monetary shares become closer to each other when $h = 11$.

Fig. 6b depicts the impact on DP_1, DP_2, DP_3, and DP_4's monetary value when the threshold is $\epsilon = 0.4$. We observe that DP_4 attains the highest monetary share because of greater information utility and its aggregated trust score. Though DP_1 does not get the highest share, its monetary share becomes closer to DP_4 at $h = 11$, 15, and 19 with the difference of approximately $\$3K$ to $\$5K$. Interestingly, DP_4's monetary share exhibits non-increasing monotonicity with the increase in specialization level h, while DP_1's monetary share increases with the increase in specialization level h except when $h = 19$. We notice that DP_3 has no monetary share when $h = 7$ because of a lack of information utility for classification analysis. The trust score does not add any monetary value if a data provider fails to contribute to information utility. The trend on DP_2 and DP_3's monetary share is not obvious with the increase in h.

Fig. 6c depicts the impact on DP_1, DP_2, DP_3, and DP_4's monetary value when the threshold is $\epsilon = 0.6$. We observe that DP_4 gains the maximum value of monetary share when $h = 3$ and $h = 7$, and DP_1 gains the maximum value of monetary share when $h = 11$ and $h = 15$, whereas DP_2 gains the maximum value of monetary share when $h = 19$. This is because it has greater information utility in competing with the other data providers at the indicated levels of specialization. We observe that DP_2's monetary share increases monotonically as the increase in specialization level h, whereas DP_4's monetary share falls with the increase in specialization level h, except when $h = 19$.

Fig. 6d depicts the impact on DP_1, DP_2, DP_3, and DP_4's monetary value when the threshold is $\epsilon = 0.8$. We observe that DP_4 achieves the highest monetary share because of greater information utility and its aggregated trust score. We observe that DP_1's monetary share generally increases as the specialization level h increases, whereas DP_4's monetary share falls with the increase in specialization level h, except when $h = 11$. We notice that when $h = 15$, all data providers' monetary shares become closer, with a difference of approximately $\$4K$.

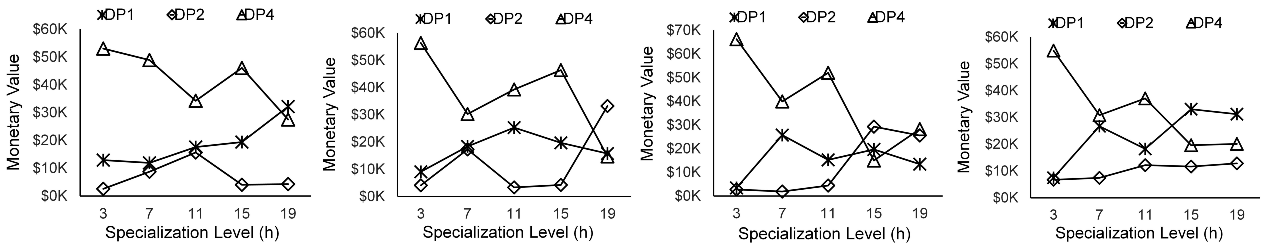
Case 2 Table IV.(b) shows the selection of attributes from each accepted data provider. Baseline accuracy (BA) on the integrated data of accepted data providers is 85.4%, using the secure multiple party classifier [21] without disclosing their raw data. We vertically partition the *Adult* dataset into three partitions VP_1, VP_2, and VP_3 for data providers DP_1, DP_2, and DP_4, respectively. Further, we split the dataset into $30,162$, and $15,060$ records for the training and testing set, respectively. Since DP_3 has dropped from the list of accepted data providers, DP_4 acquires the position of DP_3. Now, the valuation of each data provider's attribute is \$0.47, \$0.41, and \$0.36, representing $ValAttr_{DP_1}$, $ValAttr_{DP_2}$, and $ValAttr_{DP_4}$ by Eq. (14). The attribute count of each data provider is $CntAttr_{DP_1} = 3$, $CntAttr_{DP_2} = 3$, and $CntAttr_{DP_4} = 4$. The size of dataset for each data provider $|D_i| = 45,222$.

Fig. 7 depicts the impact of privacy protection and trust scores on DP_1, DP_2, and DP_4's monetary value. ϵ-*differential privacy* is enforced with privacy parameters $\epsilon = 0.2$, 0.4, 0.6, and 0.8, and specialization levels $3 \leq h \leq 19$.

Fig. 7a depicts the impact on DP_1, DP_2, and DP_4's monetary value when the threshold is $\epsilon = 0.2$. We observe that DP_4 attains the highest monetary share because of higher information utility and its trust level, except when $h = 19$. We observe that DP_1's monetary share increases as the specialization level h increases, except when $h = 7$, whereas DP_4's monetary share generally falls with the increase in specialization level h except when $h = 15$. DP_1 gains the maximum value of approximately $\$32K$ of his monetary share when $h = 19$.

Fig. 7b depicts the impact on DP_1, DP_2, and DP_4's monetary value when the threshold is $\epsilon = 0.4$. We observe that DP_4 attains the highest monetary share because of higher information utility and its trust level, except when $h = 19$. The trend on DP_1, DP_2, and DP_4's monetary share is not obvious with the increase in specialization level h. DP_2 gains the maximum value of approximately $\$33K$ of his monetary share when $h = 19$.

Fig. 7c depicts the impact on DP_1, DP_2, and DP_4's monetary value when the threshold is $\epsilon = 0.6$. We observe that DP_4 achieves the highest monetary share because of higher information utility and its trust level, except when $h = 15$.



(a) $\epsilon = 0.2$ (b) $\epsilon = 0.4$ (c) $\epsilon = 0.6$ (d) $\epsilon = 0.8$

Fig. 7: Impact of ϵ-*differential privacy* requirements and Trust scores on DP_1, DP_2, and DP_4 monetary value (Case 2)

DP_4's monetary share drops sharply when h increases from 3 to 7 and 11 to 15, while DP_1 and DP_2 have a significant increase in their monetary shares with this increase in h. DP_2 gains the maximum value of approximately $\$29K$ of monetary share when $h = 15$.

Fig. 7d depicts the impact on DP_1, DP_2, and DP_4's monetary value when the threshold is $\epsilon = 0.8$. We observe that DP_4 gains the maximum value of monetary share when h = 3, 7, and 11, whereas DP_1 gains the maximum value of monetary share when h = 15 and 19. This is because they have more information utility in competing with the other data providers at the indicated levels of specializations. DP_2's monetary share generally increases as the increase in specialization level h, except when $h = 15$. DP_1 and DP_4 do not exhibit monotonicity with the increase in h.

VII. Conclusion

In this article, we propose a novel entropy-based trust computation algorithm to verify the correctness of data from untrusted multiple data providers who own overlapping attributes over the same set of records. We achieve three main benefits in delegating the verification role to the semi-trusted cloud service provider. First, our method ensures that the cloud service provider cannot derive customers' private data from the information collected during the verification process. Second, the overhead of computation on the cloud server is also reduced because only an encrypted information gain message and its keyed hash are exchanged between a data provider and the cloud server, instead of exchanging encrypted individual data records during the verification process. Third, it also reduces the burden on data consumers to determine which data providers can serve their demands on requested attributes and what are their attained trust scores. Furthermore, we evaluate the robustness of our approach when a data provider employs machine learning method for imputation of missing values on its data. There is no significant difference in perspective to the performance of the imputation method. It is conditional to what proportion of data is missing and whether the data contains repeated patterns. If the prediction of a missing data happens to be as precise data, then it will be considered as true data. We incorporate the VCG auction mechanism to determine the pricing on data providers' attributes. It maximizes the total valuation obtained by data providers since there is no incentive to lie or deviate from truthful reporting for a data provider. From the perspective of privacy protection, the accepted data providers as a result of trust computation set up their joint privacy requirements for the data mashup. During the data mashup process, every data provider competes with the other participating data providers to produce more data utility. It is evident from the experiments that an accepted data provider whose data attributes result in more information gain, and whose trust level is higher than the other competitors, can get a significantly larger share of the monetary value.

Acknowledgment

This research is supported in part by the Research Cluster Award Fund (R16083) and Research Incentive Funds (R18055 and R19044) from Zayed University, the Discovery Grants (RGPIN-2018-03872) from the Natural Sciences and Engineering Research Council of Canada, and Canada Research Chairs Program (950-230623).

References

[1] "Data Partners," Seventh Point, 2014, Last accessed: June 11, 2015. [Online]. Available: http://www.seventhpoint.com/whitepaper/data-partners/

[2] O. Abdel Wahab, J. Bentahar, H. Otrok, and A. Mourad, "A Survey on Trust and Reputation Models for Web services: Single, Composite, and Communities," *Decision Support Systems*, vol. 74, pp. 121–134, 2015.

[3] R. Agrawal, A. Evfimievski, and R. Srikant, "Information Sharing Across Private Databases," in *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*. ACM, 2003, pp. 86–97.

[4] D. Alhadidi, N. Mohammed, B. C. M. Fung, and M. Debbabi, "Secure Distributed Framework for Achieving ϵ-Differential Privacy," in *Proceedings of the 12th International Symposium on Privacy Enhancing Technologies*. Springer, 2012, pp. 120–139.

[5] R. R. Andridge and R. J. A. Little, "A Review of Hot Deck Imputation for Survey Non-response," *International Statistical Review*, vol. 78, no. 1, pp. 40–64, 2010.

[6] M. Anisetti, C. A. Ardagna, and E. Damiani, "A Certification-Based Trust Model for Autonomic Cloud Computing Systems," in *Proceedings of the 2014 International Conference on Cloud and Autonomic Computing*, 2014, pp. 212–219.

[7] Y. Aumann and Y. Lindell, "Security Against Covert Adversaries: Efficient Protocols for Realistic Adversaries," *Journal of Cryptology*, vol. 23, no. 2, pp. 281–343, 2010.

[8] O. Benjelloun, A. Das Sarma, A. Halevy, M. Theobald, and J. Widom, "Databases with Uncertainty and Lineage," *The VLDB Journal*, vol. 17, no. 2, pp. 243–264, 2008.

[9] L. Beretta and A. Santaniello, "Nearest Neighbor Imputation Algorithms: A Critical Evaluation," *BMC Medical Informatics and Decision Making*, vol. 16, no. 3, p. 74, 2016.

[10] B. V. D. Berg and E. Keymolen, “Regulating Security on the Internet: Control versus Trust,” *International Review of Law, Computers & Technology*, vol. 31, no. 2, pp. 188–205, 2017.

[11] E. Bertino and H.-S. Lim, “Assuring Data Trustworthiness: Concepts and Research Challenges,” in *Proceedings of the 7th VLDB Conference on Secure Data Management*. Springer, 2010, pp. 1–12.

[12] E. Bertino, L. Martino, F. Paci, and A. Squicciarini, *Security for Web Services and Service-Oriented Architectures*, 1st ed. Springer, 2009.

[13] E. Bertino and R. Sandhu, “Database Security - Concepts, Approaches, and Challenges,” *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 1, pp. 2–19, 2005.

[14] D. Boneh and V. Shoup, *A Graduate Course in Applied Cryptography*, 2017.

[15] V. Chang, Y.-H. Kuo, and M. Ramachandran, “Cloud Computing Adoption Framework: A Security Framework for Business Clouds,” *Future Generation Computer Systems*, vol. 57, pp. 24–41, 2016.

[16] C. Dai, D. Lin, E. Bertino, and M. Kantarcioglu, “An Approach to Evaluate Data Trustworthiness Based on Data Provenance,” in *Secure Data Management*. Springer, 2008, pp. 82–98.

[17] E. De Cristofaro, P. Gasti, and G. Tsudik, “Fast and Private Computation of Cardinality of Set Intersection and Union,” in *Proceedings of the 11th International Conference on Cryptology and Network Security*. Springer, 2012, pp. 218–231.

[18] T. Dierks and E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.2,” RFC 5246, 2008.

[19] C. Dong, L. Chen, J. Camenisch, and G. Russello, “Fair Private Set Intersection with a Semi-trusted Arbiter,” in *Proceedings of the 27th International Conference on Data and Applications Security and Privacy*. Springer, 2013, pp. 128–144.

[20] C. Dong, L. Chen, and Z. Wen, “When Private Set Intersection Meets Big data: An Efficient and Scalable Protocol,” in *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2013, pp. 789–800.

[21] W. Du and Z. Zhan, “Building Decision Tree Classifier on Private Data,” in *Proceedings of the 2002 IEEE International Conference on Privacy, Security and Data Mining*, vol. 14. Australian Computer Society, Inc., 2002, pp. 1–8.

[22] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating Noise to Sensitivity in Private Data Analysis,” in *Proceedings of the 3rd Conference on Theory of Cryptography*. Springer, 2006, pp. 265–284.

[23] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[24] Experian Information Solutions, Inc, “The 2018 State of Data Management: A Public Sector Benchmark Report,” 2018.

[25] C. Feijo, J. L. Gmez-Barroso, and P. Voigt, “Exploring the Economic Value of Personal Information from Firms’ Financial Statements,” *International Journal of Information Management*, vol. 34, no. 2, pp. 248–256, 2014.

[26] M. J. Freedman, K. Nissim, and B. Pinkas, “Efficient Private Matching and Set Intersection,” in *Proceedings of the 2004 International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology - EUROCRYPT*. Springer, 2004, pp. 1–19.

[27] J. Freudiger, S. Rane, A. E. Brito, and E. Uzun, “Privacy Preserving Data Quality Assessment for High-Fidelity Data Sharing,” in *Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security*. ACM, 2014, pp. 21–29.

[28] A. Friedman and A. Schuster, “Data Mining with Differential Privacy,” in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2010, pp. 493–502.

[29] B. C. M. Fung, T. Trojer, P. C. Hung, L. Xiong, K. Al-Hussaeni, and R. Dssouli, “Service-Oriented Architecture for High-Dimensional Private Data Mashup,” *IEEE Transactions on Services Computing*, vol. 5, no. 3, pp. 373–386, 2012.

[30] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, “Privacy-Preserving Data Publishing: A Survey of Recent Developments,” *ACM Computing Survey*, vol. 42, no. 4, pp. 14:1–14:53, 2010.

[31] B. C. M. Fung, K. Wang, and P. S. Yu, “Anonymizing Classification Data for Privacy Preservation,” *IEEE Transactions on Knowledge Data Engineering*, vol. 19, no. 5, pp. 711–725, 2007.

[32] S. R. Ganta, S. P. Kasiviswanathan, and A. Smith, “Composition Attacks and Auxiliary Information in Data Privacy,” in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2008, pp. 265–273.

[33] Y. Huang, D. Evans, and J. Katz, “Private Set Intersection: Are Garbled Circuits Better than Custom Protocols?” in *Proceedings of the 19th Network and Distributed System Security Symposium*. The Internet Society, 2012.

[34] L. A. Hunt, *Missing Data Imputation and Its Effect on the Accuracy of Classification*. Springer, 2017.

[35] Y. Ishai, J. Kilian, K. Nissim, and E. Petrank, “Extending Oblivious Transfers Efficiently,” in *Proceedings of the CRYPTO 2003 on Advances in Cryptology*. Springer, 2003, pp. 145–161.

[36] W. Jiang and C. Clifton, “A Secure Distributed Framework for Achieving K-anonymity,” *The VLDB Journal*, vol. 15, no. 4, pp. 316–333, 2006.

[37] P. Jurczyk and L. Xiong, “Distributed Anonymization: Achieving Privacy for Both Data Subjects and Data Providers,” in *Proceedings of the 23rd Annual IFIP WG 11.3 Working Conference on Data and Applications Security*. Springer, 2009, pp. 191–207.

[38] A. Jsang, R. Ismail, and C. Boyd, “A Survey of Trust and Reputation Systems for Online Service Provision,” *Decision Support Systems*, vol. 43, no. 2, pp. 618 – 644, 2007, emerging Issues in Collaborative Commerce.

[39] S. Kamara, P. Mohassel, M. Raykova, and S. Sadeghian, “Scaling Private Set Intersection to Billion-Element Sets,” in *Proceedings of the 2014 Financial Cryptography and Data Security*. Springer, 2014, pp. 195–215.

[40] S. Kamara, P. Mohassel, and B. Riva, “Salus: A System for Server-aided Secure Function Evaluation,” in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*. ACM, 2012, pp. 797–808.

[41] S. Karabati and Z. B. Yalcin, “An Auction Mechanism for Pricing and Capacity Allocation with Multiple Products,” *Production and Operations Management*, vol. 23, no. 1, pp. 81–94, 2014.

[42] V. Kher and Y. Kim, “Securing Distributed Storage: Challenges, Techniques, and Systems,” in *Proceedings of the 2005 ACM Workshop on Storage Security and Survivability*. ACM, 2005, pp. 9–25.

[43] R. H. Khokhar, B. C. M. Fung, F. Iqbal, D. Alhadidi, and J. Bentahar, “Privacy-Preserving Data Mashup Model for Trading Person-Specific Information,” *Electronic Commerce Research and Applications*, vol. 17, pp. 19–37, 2016.

[44] D. Kifer, “Attacks on Privacy and deFinetti’s Theorem,” in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*. ACM, 2009, pp. 127–138.

[45] D. J. Kim, D. L. Ferrin, and H. R. Rao, “A Trust-Based Consumer Decision-Making Model in Electronic Commerce: The Role of Trust, Perceived Risk, and their Antecedents,” *Decision Support Systems*, vol. 44, no. 2, pp. 544 – 564, 2008.

[46] V. Kolesnikov, R. Kumaresan, M. Rosulek, and N. Trieu, “Efficient Batched Oblivious PRF with Applications to Private Set Intersection,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 818–829.

[47] N. Li, T. Li, and S. Venkatasubramanian, “t-Closeness: Privacy Beyond k-Anonymity and ℓ-Diversity,” in *Proceedings of the 23rd IEEE International Conference on Data Engineering*, 2007, pp. 106–115.

[48] X. Li, R. Ding, X. Liu, X. Liu, E. Zhu, and Y. Zhong, “A Dynamic Pricing Reverse Auction-Based Resource Allocation Mechanism in Cloud Workflow Systems,” *Scientific Programming*, vol. 2016, pp. 1–13, 2016.

[49] H.-S. Lim, G. Ghinita, E. Bertino, and M. Kantarcioglu, “A Game-Theoretic Approach for High-Assurance of Data Trustworthiness in Sensor Networks,” in *Proceedings of the 28th IEEE International Conference on Data Engineering*, 2012, pp. 1192–1203.

[50] H.-S. Lim, Y.-S. Moon, and E. Bertino, “Provenance-Based Trustworthiness Assessment in Sensor Networks,” in *Proceedings of the 7th International Workshop on Data Management for Sensor Networks*. ACM, 2010, pp. 2–7.

[51] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, “ℓ-Diversity: Privacy Beyond k-Anonymity,” *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, 2007.

[52] F. McSherry and K. Talwar, “Mechanism Design via Differential Privacy,” in *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society, 2007, pp. 94–103.

[53] N. Mohammed, D. Alhadidi, B. C. M. Fung, and M. Debbabi, “Secure Two-Party Differentially Private Data Release for Vertically Partitioned Data,” *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 1, pp. 59–71, 2014.

[54] N. Mohammed, B. C. M. Fung, and M. Debbabi, “Anonymity Meets Game Theory: Secure Data Integration with Malicious Participants,” *The VLDB Journal*, vol. 20, no. 4, pp. 567–588, 2011.

[55] N. Mohammed, B. C. M. Fung, P. C. K. Hung, and C.-K. Lee, “Centralized and Distributed Anonymization for High-Dimensional Healthcare Data,” *ACM Transactions on Knowledge Discovery from Data*, vol. 4, no. 4, pp. 18:1–18:33, 2010.

[56] R. Nget, Y. Cao, and M. Yoshikawa, "How to Balance Privacy and Money through Pricing Mechanism in Personal Data Market," in *Proceedings of the SIGIR 2017 Workshop on eCommerce*.
[57] T. H. Noor, Q. Z. Sheng, L. Yao, S. Dustdar, and A. H. Ngu, "CloudArmor: Supporting Reputation-Based Trust Management for Cloud Services," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 367–380, 2016.
[58] OECD, "Exploring the Economics of Personal Data: A Survey of Methodologies for Measuring Monetary Value," *OECD Digital Economy Papers*, no. 220, 2013.
[59] B. Pinkas, T. Schneider, and M. Zohner, "Scalable Private Set Intersection Based on OT Extension," *ACM Transactions on Privacy Security*, vol. 21, no. 2, pp. 7:1–7:35, 2018.
[60] B. Pinkas, T. Schneider, G. Segev, and M. Zohner, "Phasing: Private Set Intersection Using Permutation-Based Hashing," in *Proceedings of the 24th USENIX Conference on Security Symposium*. USENIX Association, 2015, pp. 515–530.
[61] B. Pinkas, T. Schneider, and M. Zohner, "Faster Private Set Intersection Based on OT Extension," in *Proceedings of the 23rd USENIX Conference on Security Symposium*. USENIX Association, 2014, pp. 797–812.
[62] Y. Pu and J. Grossklags, "Valuating Friends' Privacy: Does Anonymity of Sharing Personal Data Matter?" in *Proceedings of the 13th Symposium on Usable Privacy and Security (SOUPS)*. USENIX Association, 2017, pp. 339–355.
[63] J. R. Quinlan, "Induction of Decision Trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.
[64] D. Quintero, W. M. Genovese, K. Kim, M. J. M. Li, F. Martins, A. Nainwal, D. Smolej, M. Tabinowski, and A. Tiwary, *IBM Software Defined Environment*, 1st ed., 2015.
[65] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 8446, 2018.
[66] P. Samarati, "Protecting Respondents' Identities in Microdata Release," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 1010–1027, 2001.
[67] R. Shaikh and M. Sasikumar, "Trust Model for Measuring Security Strength of Cloud Computing Service," *Procedia Computer Science*, vol. 45, pp. 380–389, 2015.
[68] C. E. Shannon, "The Mathematical Theory of Communication," 1949.
[69] L.-A. Tang, X. Yu, S. Kim, J. Han, C.-C. Hung, and W.-C. Peng, "Tru-Alarm: Trustworthiness Analysis of Sensor Networks in Cyber-Physical Systems," in *Proceedings of the 2010 IEEE International Conference on Data Mining*, 2010, pp. 1079–1084.
[70] M. Tang, Y. Xu, J. Liu, Z. Zheng, and X. Liu, "Combining Global and Local Trust for Service Recommendation," in *Proceedings of the 2014 IEEE International Conference on Web Services*, 2014, pp. 305–312.
[71] R. C.-W. Wong, A. W.-C. Fu, K. Wang, and J. Pei, "Minimality Attack in Privacy Preserving Data Publishing," in *Proceedings of the 33rd International Conference on Very Large Data Bases*. VLDB Endowment, 2007, pp. 543–554.
[72] Q. Wu, M. Zhou, Q. Zhu, and Y. Xia, "VCG Auction-Based Dynamic Pricing for Multigranularity Service Composition," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 2, pp. 796–805, 2018.
[73] X. Xiao, G. Bender, M. Hay, and J. Gehrke, "iReduct: Differential Privacy with Reduced Relative Errors," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*. ACM, 2011, pp. 229–240.
[74] Y. Xiao, L. Xiong, L. Fan, S. Goryczka, and H. Li, "DPCube: Differentially Private Histogram Release Through Multidimensional Partitioning," *Transactions on Data Privacy*, vol. 7, no. 3, pp. 195–222, 2014.
[75] X. Yang, X. Luo, X. A. Wang, and S. Zhang, "Improved Outsourced Private Set Intersection Protocol Based on Polynomial Interpolation," *Concurrency and Computation: Practice and Experience*, vol. 30, no. 1, p. e4329, 2017.
[76] L. Zhang and P. N. Suganthan, "Random Forests with Ensemble of Feature Spaces," *Pattern Recognition*, vol. 47, no. 10, pp. 3429–3437, 2014.
[77] Z. Zhang, "Missing Data Imputation: Focusing on Single Imputation," *Annals of Translational Medicine*, vol. 4, no. 1, 2016.

Rashid Hussain Khokhar received his Master's degree in Information Systems Security in 2013 from Concordia Institute for Information Systems Engineering, Concordia University, Canada and is currently pursuing the Ph.D. degree in Information and Systems Engineering at the same university. The main topics of his current research activities are services computing, privacy-preserving data publishing, data mining and machine learning, data trustworthiness, information security, and risk-benefit analysis. He received an International Tuition Fee Remission Award for his Ph.D. studies. He served as a reviewer for prestigious conferences, including ACM SIGKDD, IEEE ICDM, IJCAI, PAKDD, etc.

Farkhund Iqbal holds the position of Associate Professor and Director of Advanced Cyber Forensics Research Laboratory in the College of Technological Innovation, Zayed University, United Arab Emirates. He holds a Master (2005) and a Ph.D. degree (2011) from Concordia University, Canada. He is using machine learning and big data techniques for problem-solving in health care, cybersecurity and cybercrime investigation in smart and safe city domain. He has published more than 100 papers in high ranked journals and conferences. He is an Affiliate Professor in the School of Information Studies, McGill University, Canada and an Adjunct Professor in the Faculty of Business and IT, Ontario Tech University, Canada. He is the recipient of several prestigious awards and research grants. He has served as a chair and TPC member of several IEEE/ACM conferences and is the reviewer of high-rank journals.

Benjamin C. M. Fung (M'09–SM'13) received the Ph.D. degree in computing science from Simon Fraser University in Canada in 2007. He is a Canada Research Chair in Data Mining for Cybersecurity, an Associate Professor with the School of Information Studies, an Associate Member with the School of Computer Science at McGill University in Canada, and a Co-Curator of cybersecurity in the World Economic Forum. He has over 120 refereed publications that span the research forums of data mining, privacy protection, cybersecurity, services computing, and building engineering. His data mining works in crime investigation and authorship analysis have been reported by media worldwide. Dr. Fung is a licensed Professional Engineer of software engineering in Ontario, Canada.

Jamal Bentahar received the bachelors degree in software engineering from the National Institute of Statistics and Applied Economics, Morocco, in 1998, the M.Sc degree in the same discipline from Mohamed V University, Morocco, in 2001, and the Ph.D. degree in computer science and software engineering from Laval University, Canada, in 2005. He is a Professor with Concordia Institute for Information Systems Engineering, Faculty of Engineering and Computer Science, Concordia University, Canada. From 2005 to 2006, he was a Postdoctoral Fellow with Laval University, and then NSERC Postdoctoral Fellow at Simon Fraser University, Canada. He is an NSERC Co-Chair for Discovery Grant for Computer Science (2016-2018). His research interests include the areas of computational logics, model checking, multi-agent systems, service and cloud computing, game theory, and software engineering.