(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
01 June 2023 (01.06.2023)

WIPO | PCT

(10) International Publication Number
WO 2023/096571 A2

(51) International Patent Classification:
Not classified

(21) International Application Number:
PCT/SG2022/050739

(22) International Filing Date:
18 October 2022 (18.10.2022)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
17/535,398 24 November 2021 (24.11.2021) US

(71) Applicant: **LEMON INC.** [GB/SG]; P.O. Box 31119 Grand Pavilion, Hibiscus Way, 802 West Bay Road, Grand Cayman, KY1-1205 (KY).

(72) Inventors: **YANG, Xin**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **YAO, Yuanshun**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **LIU, Tianyi**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **SUN, Jiankai**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **WANG, Chong**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **WU, Ruihan**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US).

(74) Agent: **POH, Chee Kian, Daniel**; Marks & Clerk Singapore LLP, Tanjong Pagar Post Office, P.O. Box 636, Singapore 910816 (SG).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE,

(54) Title: DATA PROCESSING FOR RELEASE WHILE PROTECTING INDIVIDUAL PRIVACY

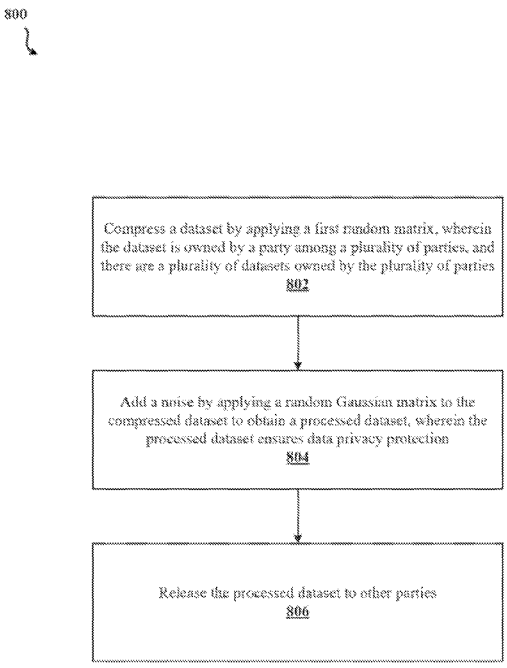

FIG. 8

(57) Abstract: The present disclosure describes techniques of releasing data while protecting individual privacy. A dataset may be compressed by applying a first random matrix. The dataset may be owned by a party among a plurality of parties and there may be a plurality of datasets owned by the plurality of parties. A noise may be added by applying a random Gaussian matrix to the compressed dataset to obtain a processed dataset. The processed dataset ensures data privacy protection. The processed dataset may be released to other parties.

WO 2023/096571 A2

[Continued on next page]

DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

DATA PROCESSING FOR RELEASE WHILE PROTECTING INDIVIDUAL PRIVACY

[0001] The present application claims priority to U.S. Application NO. 17/535,398 entitled "DATA PROCESSING FOR RELEASE WHILE PROTECTING INDIVIDUAL PRIVACY" filed with the USPTO on November 24, 2021, the disclosures of which are incorporated herein in their entireties by reference.

BACKGROUND

[0002] A single company may possess the personal information of a large quantity of individuals or customers. The company may need to keep such data private in order to protect customers' identities and to protect the company's reputation. However, maintaining the privacy of such data often leads to difficulties in disseminating the data. Improvements in techniques of data privacy and protection are desirable.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] The following detailed description may be better understood when read in conjunction with the appended drawings. For the purposes of illustration, there are shown in the drawings example embodiments of various aspects of the disclosure; however, the invention is not limited to the specific methods and instrumentalities disclosed.

[0004] FIG. 1 shows an example prior art system for data release.

[0005] FIG. 2 shows an example system for differentially private data release.

[0006] FIG. 3 shows example data associated with a plurality of parties.

[0007] FIG. 4 shows a plurality of exemplary graphs.

[0008] FIG. 5 shows an example scatter plot.

[0009] FIG. 6 shows an example table illustrating an evaluation of the techniques described in the present disclosure.

[0010] FIG. 7 shows an example process for differentially private data release.

[0011] FIG. 8 shows another example process for differentially private data release.

[0012] FIG. 9 shows another example process for differentially private data release.

[0013] FIG. 10 shows an example computing device which may be used to perform any of the techniques disclosed herein.

DETAILED DESCRIPTION OF ILLUSTRATIVE EMBODIMENTS

[0014] Open and public datasets are beneficial to many people and communities. For example, the machine learning community has greatly benefitted from open and public datasets. Unfortunately, privacy concerns surrounding the dissemination of data significantly limit the feasibility of sharing many rich and useful datasets to the public. This is especially true in privacy-sensitive domains like health care, finance, government, etc. Such restriction on the dissemination of data considerably slows down research in those areas. In particular, such restriction on the dissemination of data slows down research in the machine learning realm, as many of today's machine learning algorithms require a large amount of data to be sufficiently trained. Additionally, legal and moral concerns surrounding the protection of individual privacy are becoming increasingly important. For example, many countries have imposed strict regulations on the usage and release of sensitive data (e.g., CCPA, HIPPA, and GDPR).

[0015] As such, a tension exists between privacy protection and research promotion. Differentially Private (DP) data release can be used to mitigate such tension. DP is a standard privacy guarantee for algorithms. To hide a single data point from the whole dataset, it requires the impact of any single data point is limited for the overall output. By limiting the impact of any single data point, a change in one data point does not lead to significant change in the distribution of randomized output. Formally, DP can be stated as the following: (ε, δ)-differential privacy. A randomized mechanism $\mathcal{M}: \mathcal{D} \rightarrow \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ε, δ)-differential privacy if for any two adjacent datasets $\mathcal{D}, \mathcal{D}' \in \mathcal{D}$, which differ at exactly one data point, and for any subset of outputs $S \subseteq \mathcal{R}$, it holds that $\mathbb{P}[\mathcal{M}(\mathcal{D}) \in S] \leq e^{\varepsilon} \cdot \mathbb{P}[\mathcal{M}(\mathcal{D}') \in S] + \delta$.

[0016] DP data release is a technique to disseminate data without compromising the privacy of data subjects. DP data release provides a formal definition of privacy to regulate the trade-off between two conflicting goals: protecting sensitive information and maintaining data utility. In a DP data release mechanism, the shared dataset is a function of the aggregate of all private data samples. The DP data release guarantees regulate how difficult it is for anyone to infer the attributes or identity of any individual data sample. With a high probability, the public data would be barely affected if any single sample were replaced.

[0017] However, many existing DP data release techniques are focused on data release in scenarios where a single party owns all the data and would release that data to the public. In many real-world scenarios, though, there exist multiple parties who each own one or more datasets that are relevant to each other. These multiple parties may want to share their data amongst each other and/or with the public as a whole. In other words, the multi-party setting assumes that multiple parties exist that own disjoint sets of attributes (features or labels) belonging to the same group of data subjects (e.g., patients). For example, in the health care domain, a patient may visit multiple

clinics for specialized treatments. Each clinic may only have access to its own attributes (e.g., blood test and CT images) collected from the patient. For the same patient, attributes combined from all clinics that the patient has visited can be more useful to train machine learning models.

[0018] One existing approach to release data in a multi-party setting can be performed using the system 100 of FIG. 1. The system 100 includes multiple parties (e.g., a plurality of data providers 102a-n). Each of the data providers 102a-n may maintain a database 104a-n that is configured to store data 106a-n. For example, each of the data providers 102a-n may be a particular health clinic or healthcare provider. Each of the health clinics or healthcare providers may maintain a database that is configured to store patient data. More than one of the databases maintained by the health clinics or healthcare providers may store data associated with the same patient. It may be useful to train a machine learning model using the attributes combined from all clinics that the same patient has visited. In other embodiments, one or more (or all) of the data providers 102a-n may be unrelated to the healthcare profession. Instead, one or more of the data providers 102a-n may be any individual, entity, or business that wants to share their data with the other data providers 102a-n and/or with the public as a whole (e.g., computing devices 116a-n).

[0019] The system 100 includes a data aggregator 110. To release the data 106a-n to the other data providers 102a-n and/or with the public as a whole, the data 106a-n may be sent to a centralized place. For example, the data 106a-n may be sent to the data aggregator 110. While depicted in FIG. 1 as being separate from the data providers 102a-n, it should be appreciated that in other embodiments, the data aggregator 110 may be associated with or maintained by one of the data providers 102a-n. In other embodiments, the data aggregator is unassociated with any of the data providers 102a-n. The data aggregator 110 can combine the data from the data providers 102a-n and then release the combined data using a single-party data release approach.

[0020] The data providers 102a-n and the computing devices 116a-n may communicate with each other via one or more networks 120. The data aggregator 110 may communicate with the data providers 102a-n and/or the computing devices 116a-n via the one or more networks 120. The network 120 comprise a variety of network devices, such as routers, switches, multiplexers, hubs, modems, bridges, repeaters, firewalls, proxy devices, and/or the like. The network 120 may comprise physical links, such as coaxial cable links, twisted pair cable links, fiber optic links, a combination thereof, and/or the like. The network 120 may comprise wireless links, such as cellular links, satellite links, Wi-Fi links and/or the like.

[0021] The computing devices 116a-n may comprise any type of computing device, such as a server, a mobile device, a tablet device, laptop, a desktop computer, a smart television or other smart device (e.g., smart watch, smart speaker, smart glasses, smart helmet), a gaming device, a

set top box, digital streaming device, robot, and/or the like. The computing devices 116a-n may be associated with one or more users. A single user may use one or more of the computing devices 116a-n to access the network 120.

[0022] In some instances, one or more of the data providers 102a-n may be prohibited from sending data to another party. This may be especially true for a privacy-sensitive organization like a health clinic or healthcare provider. In such instances, the data 106a-n may not be able to be sent to the data aggregator 110. An alternative existing approach is to let each data provider 102a-n individually release its own data 106a-n to the public (e.g., computing devices 116a-n) after adding data-wise Gaussian noise to the data. The public, including machine learning practitioners, can combine the data together to train machine learning models. However. the resulting machine learning models trained on the data combined in this way show a significantly lower utility compared to the machine learning models trained on non-private data.

[0023] The systems and methods described herein bridge this utility gap related to DP data release in the multi-party setting. Different stakeholders may own disjoint sets of attributes belonging to the same group of data subjects. Two DP algorithms within the context of linear regression are described herein. The two DP algorithms described herein allow the parties in the multi-party setting to share attributes of the same data subjects publicly through a DP mechanism. The two DP algorithms described herein protect the privacy of all data subjects and can be accessed by the public, including any party in the multi-party setting. For example, the parties in the multi-party setting or the public may use the data shared through the DP mechanisms described herein to train machine learning models on the complete dataset (e.g., data from all parties in the multi-party setting) without the ability to infer private attributes or the identities of individuals.

[0024] The DP algorithms as described in the present disclosure may be utilized for a variety of different reasons in a variety of different systems. FIG. 2 illustrates an example system 200 capable of employing the two, improved DP algorithms described herein. The system 200 includes multiple parties (e.g., the plurality of data providers 102a-n). As described above, each of the data providers 102a-n may maintain a database 104a-n that is configured to store data 106a-n. For example, each of the data providers 102a-n may be a particular health clinic or healthcare provider. Each of the health clinics or healthcare providers may maintain a database that is configured to store patient data. More than one of the databases maintained by the health clinics or healthcare providers may store data associated with the same patient. It may be useful to train a machine learning model using the attributes combined from all clinics that the same patient has visited. One or more of the data providers 102a-n may be any individual, entity, or business that wants to share their data with

the other data providers 102a-n and/or with the public as a whole (e.g., computing devices 116a-n).

[0025] FIG. 3 shows an example data 300 associated with three of the data providers 102a-n. The first data set 302a is associated with a first data provider 102a, the second data set 302b is associated with a first data provider 102b, and the third data set 302c is associated with a third data provider 102c. The data sets 302a-c are all associated with the same three patients (Alice, Bob, and Charlie). The three patients have visited all three of the data providers 102a-c (e.g., health clinics or healthcare providers). For example, the three patients may all have visited the first data provider 102a, where the three patients had blood work done. The first data provider 102a may maintain data associated with the results of those blood tests for all three of the patients. Similarly, the three patients may all have visited the second data provider 102b, where the three patients had CT scans. The second data provider 102b may maintain data associated with the CT scans for all three of the patients. Lastly, the three patients may all have visited the third data provider 102c, where the three patients had their liver cancer assessed. The third data provider 102c may maintain data associated with the results of those liver cancer assessments for all three of the patients. It may be useful to train a machine learning model using the attributes combined from all clinics that these patients have visited, as each of these clinics have different information related to the patients. When the data from all of the clinics is viewed as a whole, it provides a more comprehensive understanding of the three patients' health.

[0026] Referring back now to FIG. 2, each of the data providers 102a-n include a differential privacy model 202a-n. Each differential privacy model 202a-n is configured to perform one or more of the two, improved DP algorithms described herein. Both of the two, improved DP algorithms are based on a Gaussian DP Mechanism within the context of linear regression. Gaussian mechanism is a post-hoc mechanism to convert a deterministic real valued function f: $\mathrm{D} \rightarrow \mathbb{R}^m$ to a randomized algorithm with differential privacy guarantee. It relies on sensitivity of f, which is defined as the maximum difference of output $\| f(\mathrm{D}) - f(\mathrm{D}')\|$ where D and D′ are two adjacent datasets. The Gaussian mechanism for differential privacy can be stated as the following lemma.

[0027] The first lemma states that, for any deterministic real-valued function f: $\mathrm{D} \rightarrow \mathbb{R}^m$ with sensitivity S_f, a randomized function can be defined by adding Gaussian noise to f: $f^{dp}(\mathrm{D}) := f(\mathrm{D}) + \mathrm{R}$, where R is sampled from a multivariate normal distribution $\mathcal{N}(0, S_f^2\sigma^2 \cdot I)$. When $\sigma \geq$

$\frac{\sqrt{2\log(\frac{1}{1.25\delta})}}{\varepsilon}$, f^{dp} is (ε, δ)-differential privacy. To simplify notations, $\sigma_{\varepsilon,\delta}$ can be defined as $\frac{\sqrt{2\log(\frac{1}{1.25\delta})}}{\varepsilon}$.

[0028] The second lemma (e.g., Johnson-Lindenstrauss (JL) Lemma) is a technique to compress a set of vectors S= $\{v_1, \ldots, v_l\}$ with dimension d to a lower dimension space $k < d$. With a proper selection k, S can be compressed so that the distance between any two vectors can be maintained with a high probability. The Bernoulli version of the JL Lemma states, if S is an arbitrary set of l points in $\mathbb{R}^d$, and A is a $k \times d$ random matrix, where A_{ij} are independent random variables from the following distribution: $Aij = \begin{cases} +1 & with\ probability\ 1/2 \\ -1 & with\ probability\ 1/2 \end{cases}$, and $M := \frac{1}{\sqrt{k}}A$, then with probability at least $(1-l_2)\exp(-k(\frac{\beta^2}{4} - \frac{\beta^3}{6}))$, for any $\mathbf{u}, \mathbf{v} \in S$, $(1-\beta)\|\mathbf{u} - \mathbf{v}\| \leq \|M\mathbf{u} - M\mathbf{v}\| \leq (1+\beta)\|\mathbf{u} - \mathbf{v}\|$.

[0029] From this distance preserving lemma, the following inner-product preserving corollary can be conducted. The corollary (e.g., JL Lemma (Bernoulli) corollary) for inner-product preserving corollary) follows the same definitions for the point set S and a random matrix M in the second lemma described above, and s may further be denoted as any upper bound for the maximum norm for vectors in S, *i.e.* $s \geq max_{u \in S} \|\mathbf{u}\|$. With probability of at least $1-(l+1)^2\exp(-k(\frac{\beta^2}{4} - \frac{\beta^3}{6}))$, $\frac{\mathbf{u}^T\mathbf{v}}{s^2} - 4\beta \leq \frac{(M\mathbf{u})^T}{s^2} \leq \frac{\mathbf{u}^T\mathbf{v}}{s^2} + 4\beta$.

[0030] The first improved DP algorithm is a De-biased Gaussian Mechanism for Ordinary Least Squares (DGM-OLS). To perform the first DP algorithm, the differential privacy models 202a-n may add Gaussian noise directly to data 106a-n belonging to the respective data provider 102a-n. For example, the differential privacy model 202a may add Gaussian noise directly to data 106a belonging to the data provider 102a, etc. After adding Gaussian noise to the data 106a-n, the differential privacy models 202a-n may continue to perform the first DP algorithm by removing certain biases from the resulting Hessian matrix.

[0031] The second improved DP algorithm is a Random Mixing prior to Gaussian Mechanism for Ordinary Least Squares (RMGM-OLS). To perform the second DP algorithm, all of the differential privacy models 202a-n may utilize a shared Bernoulli projection matrix. All of the differential privacy models 202a-n may utilize the shared Bernoulli projection matrix to compress the data 106a-n belonging to the respective data provider 102a-n along a sample-wise dimension. After compressing the data 106a-n, the differential privacy models 202a-n may continue to perform the second DP algorithm adding Gaussian noise to the compressed data.

[0032] The following notations will be used to describe the first and second improved DP algorithms in more detail below. D^j, where j = 1, . . . , m, as data matrices for m parties, where $D^j \in \mathbb{R}^{n \times d_j}$ and $m \geq 2$. They are aligned by the same set of subjects but have different attributes. D is defined as $[D^1, \ldots, D^m] \in \mathbb{R}^{n \times (d+1)}$ as the collection of all datasets, where $d = d_1 + \ldots + d_m - 1$ (d may be reduced by -1 because it may be assumed that one of the columns in the label, which will be treated separately). D_i may be denoted as the ith data (row) of D. It may be assumed that D_i, $i = 1, \ldots, n$, are *i.i.d* sampled from an underlying distribution P over $\mathbb{R}^{d+1}$.

[0033] The first and second improved DP algorithms described in more detail below are subject to a privacy constraint. The privacy constraint denotes A_1 as the release algorithm. A_1 takes D^j, $j = 1, \ldots, m$, from all parties as input and outputs the private dataset $D^{pub} \in \mathbb{R}^{n' \times (d+1)}$ ($n' = n$ is not restricted). It is not enough to only ask D^{pub} to be private, because each data provider 102a-n does not want to leak the privacy to the other data providers 102a-n either. If I is the set information shared between parties in the algorithm A_1, it may be required that for any $j \in \{1, \ldots, m\}$, (D^{pub}, I) is (ε, δ)-differentially private with respect to dataset D^j.

[0034] The first and second improved DP algorithms described in more detail below are associated with a utility target. One of the attributes that the linear regression task is associated with is the label and all remaining attributes are features. The label is assumed to be the last dimension. The population loss for any linear model $\mathbf{w} \in \mathbb{R}^d$ is defined by $L(\mathbf{w}; P) = E_{(x,y)} \sim P [(\mathbf{w}^T\mathbf{x} - y)^2]$. The minimizer for the population loss may be defined as $\mathbf{w}^*$. A_2 and $\hat{w}_n$ may be denoted as the learning algorithm from private data D^{pub} and the learned linear model. The learning algorithm A_2 may depend on A_1. $\hat{w}_n$ may converge to $\mathbf{w}^*$ in probability as the size of dataset n increases. Formally, it is defined as $\forall \beta > 0, \lim_{n \to \infty} \mathbb{P}[|| \hat{w}_n - \mathbf{w}^*|| > \beta] = 0$. A sequence $\{X_n\}$ of random variables in $\mathbb{R}^d$ converges in probability towards the random variable X if for all $\beta > 0$, $\lim_{n \to \infty} \mathbb{P}[||Xn-X|| > \beta]=0$. This convergence may be denoted as $\operatorname{plim}_{n \to \infty} X_n = X$.

[0035] Two more assumptions may also be made for both the first and second improved DP algorithms. First, it may be assumed that the absolute values of all attributes are bounded by 1. For example, it may be assumed that all data samples have been scaled or normalized and that $E_{(x,y)} \sim P [\mathbf{x}\mathbf{x}^T]$ exists . Second, the no perfect multicollinearity assumption may apply. The no perfect multicollinearity assumption states that there is no exact linear relationship among independent variables. For example, it may be assumed that $E_{(x,y)} \sim P [\mathbf{x}\mathbf{x}^T]$ is positive definite. For this assumption, $\mathbf{w}^*$ has an explicit form: $(E_{(x,y)} \sim P [\mathbf{x}\mathbf{x}^T])^{-1} E_{(x,y)} \sim P [\mathbf{x} \cdot y]$.

[0036] As discussed above, the first improved DP algorithm is a De-biased Gaussian Mechanism for Ordinary Least Squares (DGM-OLS). The first improved DP algorithm includes two stages:

dataset release and learning. The first improved DP algorithm may be represented by A^{dgm}, the dataset release stage may be represented by A_1^{dgm}, and the learning stage may be represented by A_2^{dgm} so that $A^{dgm} = (A_1^{dgm}, A_2^{dgm})$.

[0037] In embodiments, for A_1^{dgm}, to meet the privacy requirement for A_1, a Gaussian mechanism may be directly applied to each data matrix D^j ($j = 1, \ldots, m$). Two neighboring data matrices D^j and $(D^j)'$ may differ at exactly one row and this row index may be denoted as i. As discussed above, it is assumed that the absolute values of all attributes are bounded by 1. Accordingly, $\| D^j$ and $(D^j)'\| = \|D_i^j$ and $(D_i^j)'\| \leq 2\sqrt{d_j} \leq 2\sqrt{d}$, where $d_j \leq d$ is due to $m \geq 2$. Each data provider 102a-n may independently add a Gaussian noise, such as a Gaussian noise R^j to D^j. Elements in D^1may be independent and identically distributed (e.g., i.i.d) sampled from a Gaussian distribution with a mean zero and a variance $4d\sigma_{\varepsilon,\delta}^2$. A_1^{dgm} ($D^1, \ldots, D^m$) may be equal to [$D^1 + R^1, \ldots, D^m + R^m$]. Because there is no shared information I between different data providers 102a-n in the first improved DP algorithm and A_1^{dgm} ($D^1, \ldots, D^m$) is (ε, δ)-differentially private with respect any dataset D^j, it meets the privacy constraint described above. D^{dgm} may be denoted by $A_1^{dgm}(D^1, \ldots, D^m)$.

[0038] In embodiments, for A_2^{dgm}, the feature matrix and the label vector of the private joint dataset may be written as $[X,Y] = D$. The feature matrix and the label vector from the released dataset may be similarly denoted as $[X^{dgm}, Y^{dgm}] = D^{dgm}$. R may be defined as being equal to [$R^1, \ldots, R^m$] $\in \mathbb{R}^{n\times(d+1)}$ and R may be split into split R_X and R_Y, representing the additive noise to X and Y respectively. It may be observed that $\operatorname{plim}_{n\to\infty} \frac{1}{n}(X^{dgm})^T X^{dgm} = \operatorname{plim}_{n\to\infty} \frac{1}{n}(X^T X + X^T R_x + R_x^T X + R_x^T X) = \mathrm{E}_{(x,y)\sim P}[\mathbf{x}\mathbf{x}^T] + 4d\sigma_{\varepsilon,\delta}^2 \cdot I$, where the last equation holds by the concentration of bounded random variables and multivariate normal distribution.

[0039] Similarly, $plim_{n\to\infty} \frac{1}{n}(X^{dgm})^T Y^{dgm} = \mathrm{E}_{(x,y)\sim P}[\mathbf{x}\cdot y]$. The learning algorithm A_2^{dgm} may be defined as $A_2^{dgm}(D^{dgm}) = \frac{1}{n}(\widehat{H}_n^{dgm})^{-1}(X^{dgm})^T Y^{dgm}$, wherein $\widehat{H}_n^{dgm} := \frac{1}{n}(X^{dgm})^T X^{dgm} - 4d\sigma_{\varepsilon,\delta}^2 \cdot I$. The utility is guaranteed by the following theorem: when $\beta \leq c$ for some variable c that is dependent on $\sigma_{\varepsilon,\delta}^2$, d, and P, but is independent of n, $\mathbb{P}[\| \widehat{w}_n^{dgm} - w^* \| \leqslant \beta] \geq 1\text{-}\exp\left(-\tilde{O}\left(\beta^2 \frac{n}{\sigma_{\epsilon,\delta}^4 d^4}\right)\right)$. $\widehat{H}_n^{dgm}$ is a random matrix. Although the expectation (on average) of $\widehat{H}_n^{dgm}$ is positive definite, each instance may not be positive definite. The existence of the inverse of $\widehat{H}_n^{dgm}$ is not guaranteed. It may be possible that it has a small eigenvalue. As a result, its inverse may get

exploded. The second improved DP algorithm, discussed below in more detail, overcomes this limitation.

[0040] In summary, the first improved DP algorithm can be stated as follows: For the release algorithm $A_1^{dgm}(D = [D^1, \ldots, D^m], \epsilon, \delta)$: (1), for $j = 1, \ldots, m$, do (2) $(D^{dgm})^j := D^j + R^j$, where $R^j \in \mathbb{R}^{n \times d_j}$ and is a random Gaussian matrix and elements in R^j are i.i.d. sampled from $\mathcal{N}(0, 4d \cdot \sigma_{\epsilon,\delta}^4)$. (3) end for (4) $D^{dgm} := [(D^{dgm})^1, \ldots, (D^{dgm})^m]$ (5) return D^{dgm}. For the learning algorithm $A_2^{dgm}(D^{\text{dgm}}; \epsilon, \delta)$: (1) $[X^{\text{dgm}}, Y^{\text{dgm}}] = D^{\text{dgm}}$ (2) $\widehat{H}_n^{dgm} := \frac{1}{n}(X^{dgm})^{\mathrm{T}} X^{dgm} - 4d\sigma_{\varepsilon,\delta}^2 \cdot I$ (3) $\widehat{w}_n^{dgm} := (\widehat{H}_n^{dgm})^{-1} \frac{1}{n}(X^{dgm})^{\mathrm{T}} Y^{dgm}$. (4) return $\widehat{w}_n^{dgm}$.

[0041] As discussed above, the second improved DP algorithm is a Random Mixing prior to Gaussian Mechanism for Ordinary Least Squares (RMGM-OLS) algorithm. For the release algorithm A_1^{dgm} of the first improved DP algorithm, the norm of additive noise R_x has the same order as the norm feature matrix X in n. Both of them are in linear order of n. The noise may not diminish as $n \rightarrow \infty$ and as a result, $4d\sigma_{\varepsilon,\delta}^2 \cdot I$ has to be subtracted from $(X^{dgm})^{\mathrm{T}} X^{dgm}$ to overcome this additive noise in A_1^{dgm}. To make the summation of n data points differentially private, a constant additive noise may be enough. This constant additive noise is relatively small compared with the summation of n data points.

[0042] The second improved DP algorithm takes this insight into account. The second improved DP algorithm includes two stages: dataset release and learning. The first improved DP algorithm may be represented by A^{rmgm}, the dataset release stage may be represented by A_1^{rmgm}, and the learning stage may be represented by A_2^{rmgm} so that $A^{dgm} = (A_1^{rmgm}, A_2^{rmgm})$.

[0043] In embodiments, for A_1^{rmgm}, $\mathbf{a}$ may be denoted as any n dimensional vector in $\{-1,1\}^{\mathrm{n}}$. Suppose D^j and $(D^j)'$ are two neighboring datasets and they are different at row index i. The sensitivity of $\mathbf{a}^{\mathrm{T}} D^j$ is $\|\mathbf{a}^{\mathrm{T}} D^j - \mathbf{a}^{\mathrm{T}}(D^j)'\| = \| D_i^j - (D_i^j)'\| \leq 2\sqrt{d_j} \leq 2\sqrt{d}$. Generally, if $A \in \{-1, 1\}^{\mathrm{k\,x\,n}}$, $\frac{1}{\sqrt{k}} \cdot AD^j$ has sensitivity $2\sqrt{d}$ as well. A_1^{rmgm} can then be defined as $A_1^{rmgm}(D^1, \ldots, D^m) = \left[\frac{1}{\sqrt{k}} \cdot AD^1 + R^1, \ldots, \frac{1}{\sqrt{k}} \cdot AD^m + R^m\right]$, where $A \in \{-1, 1\}^{\mathrm{k\,x\,n}}$ is a random matrix that each element is i.i.d. sampled from the distribution A_{ij} defined above and each $R^j \in \mathbb{R}^{k \times d_j}$ is a random matrix that each element is i.i.d. sampled from a Gaussian distribution with a mean zero and a variance $4d\sigma_{\epsilon,\delta}^2$.

[0044] To see how the insight gleaned from the first improved DP algorithm can be applied to the second improved DP algorithm, rows of AD^j are k pseudo data points, where each data point is a ± 1 mixing of n points in D_j. The additive noise R^j is relatively small compared to $\frac{1}{\sqrt{k}} AD^j$.

Indeed, $||R^j||$'s order is $\tilde{O}(k)$. With JL Lemma, $||\frac{1}{\sqrt{k}}AD^j|| \approx ||D^j||$ is in order $\tilde{O}(n)$. If $k = o(n)$, the additive noise will diminish as zero → ∞.

[0045] Regarding the privacy guarantee, D^{rmgm} may be denoted as $A_1^{rmgm}(D^1, \ldots, D^m)$. In the above definition, A is shared among all parties. It then requires that (D^{rmgm}, A) is (ε, δ)-differentially private. Because it is assumed that the absolute values of all attributes are bounded by 1, D^{rmgm}, A) is (ε, δ)-differentially private.

[0046] In embodiments, for A_2^{rmgm}, the feature matrix and the label vector from the released dataset may be denoted as $[X^{\text{rmgm}}, Y^{\text{rmgm}}] = D^{\text{rmgm}}$. R may be defined as being equal to $[R^1, \ldots, R^m] \in \mathbb{R}^{k\times d}$ and R may be split into split $R_X \in \mathbb{R}^{k\times d}$ and $R_Y \in \mathbb{R}^k$, representing the additive noise to $\frac{1}{\sqrt{k}}AX$ and $\frac{1}{\sqrt{k}}AY$ respectively. First, regarding $\frac{1}{n}(X^{\text{rmgm}})^{\text{T}}X^{\text{rmgm}}$: with the fact that $\frac{1}{n}(X)^{\text{T}}X$ would converge to $\text{E}_{(x,y)}\ P\ [\mathbf{x}\mathbf{x}^{\text{T}}]$, $\frac{1}{n}(X^{\text{rmgm}})^{\text{T}}X^{\text{rmgm}}$ - $\frac{1}{n}(X)^{\text{T}}X$ converges to zero by the following decomposition: $\frac{1}{n}\left(X^T\frac{A^TA}{k}X - X^TX\right) + \frac{1}{n}\left(X^T\frac{A^T}{\sqrt{k}}R_x + R_x^T\frac{A}{\sqrt{k}}X\right) + \frac{1}{n}R_x^TR_x$. Each term in this decomposition converges to zero, as detailed below.

[0047] The first term, $\frac{1}{n}\left(X^T\frac{A^TA}{k}X - X^TX\right)$, converges to zero. Because it is assumed that the absolute values of all attributes are bounded by 1, it follows that each column in X is no larger than $\sqrt{n}$. To apply the JL Lemma (Bernoulli) corollary described above, the point set S can be set as the set of column vectors and the previous argument shows that $s=\sqrt{n}$ is one upper bound for maximum norm of vectors in S. The JL Lemma (Bernoulli) corollary helps guarantee the approximation.

[0048] The second term, $\frac{1}{n}\left(X^T\frac{A^T}{\sqrt{k}}R_x + R_x^T\frac{A}{\sqrt{k}}X\right)$, converges to zero. $X_{:j}$ may be denoted as the jth column of X. With high probability (with respect to a large n), each element in $\frac{1}{n}X_{:j}$ is bounded by some small $\alpha = o\left(\frac{1}{\sqrt{n}}\right)$ from Hoeffding inequality. Conditioned on any fixed $\frac{1}{n}X_{:j} \le \alpha \cdot 1$, $\frac{1}{\sqrt{k}}((R_x)_{:,i})^T\frac{1}{n}AX_{:j}$ is equivalent to a Gaussian variable with mean zero and variance less than α. Then the concentration of Gaussian distribution implies that with high probability, $\frac{1}{\sqrt{k}}((R_x)_{:,i})^T\frac{1}{n}AX_{:j}$ is small. The union bound leads to the convergence of this $d \times d$ matrix $||\frac{1}{n}\left(X^T\frac{A^T}{\sqrt{k}}R_x + R_x^T\frac{A}{\sqrt{k}}X\right)||$.

[0049] The third term, $\frac{1}{n}R_x^TR_x$, converges to zero. $||\frac{1}{k}R_x^TR_x||$ will converge to $4\text{d}\sigma_{\varepsilon,\delta}^2 \cdot I$ if k increases as n → ∞. If k = o(n), $\frac{k}{n} \cdot 4\text{d}\ \sigma_{\varepsilon,\delta}^2 \cdot I$ will converge to zero as n → ∞.

$\operatorname{plim}_n \frac{1}{n}(X^{rmgm})^T Y^{rmgm} - \frac{1}{n}X^T Y = 0$ is analyzed similarly. The learning stage algorithm $A_2^{rmgm}(D^{rmgm})$ may then be defined as $\frac{1}{n}\left(\widehat{H}_n^{rmgm}\right)^{-1}(X^{rmgm})^T Y^{rmgm}$, where $\widehat{H}_n^{rmgm} := \frac{1}{n}(X^{rmgm})^T X^{rmgm}$. This convergence relies on the proper selection of k. There exists a trade-off. Namely a larger k makes better rate for the approximation to recover the inner product but makes a worse rate for the diminishment of additive noise. The utility is guaranteed by the following theorem: when $\beta \le c$ for some variable c that is dependent on $\alpha_{\varepsilon,\delta}$, d, and P, but is independent of n, $\mathbb{P}[\| \widehat{w}_n^{rmgm} - w^* \| \le \beta] \ge$ 1-exp $\left(-O\left(\min\{\frac{k\beta^2}{d^2}, \frac{n\beta}{kd^2\alpha_{\varepsilon,\delta}^2}, \frac{n^{\frac{1}{2}}\beta}{d^{\frac{3}{2}}\alpha_{\varepsilon,\delta}}\}\right) + \tilde{O}(1)\right)$. If k=O$\left(\frac{n^{\frac{1}{2}}\beta}{\alpha_{\varepsilon,\delta}}\right)$, $\mathbb{P}[\|\widehat{w}_n^{rmgm} - w^*\| \le \beta] \ge$ 1-exp $\left(-O\left(\min\{\frac{n^{1/2}\beta^2}{d^2\alpha_{\varepsilon,\delta}}, \frac{n^{1/2}\beta}{d^2\alpha_{\varepsilon,\delta}}, \frac{n^{\frac{1}{2}}\beta}{d^{\frac{3}{2}}\alpha_{\varepsilon,\delta}}\}\right) + \tilde{O}(1)\right)$. In the theorem, k is selected to balance $\frac{k\beta^2}{d^2}$ and $\frac{n\beta}{kd^2\alpha_{\varepsilon,\delta}^2}$. To achieve the optimal rate for $f(\beta)$ with any fixed β, the optimal k is chosen as O$\left(\frac{n^{1/2}}{\alpha_{\varepsilon,\delta}}\right)$.

[0050] In summary, the second improved DP algorithm can be stated as follows: For the release algorithm $A_1^{rmgm}(D = [D^1, \ldots, D^m], \epsilon, \delta, k)$: (1) A first data provider 102a-n samples a k x n random matrix A, where all A_{ij} are i.i.d. sampled from the distribution $Aij = \begin{cases} +1 & with\ probability\ 1/2 \\ -1 & with\ probability\ 1/2 \end{cases}$ and sends A to all other data providers 102a-n. (2) for j=1, . . . , m do (3) $(D^{\text{rmgm})j} := \frac{1}{\sqrt{k}} \cdot AD^j + R^j$ is a k x d_j random matrix and elements in R^j are i.i.d. sampled from $\mathcal{N}(0, 4d\sigma_{\epsilon,\delta}^2)$. (4) end for (5) $D^{\text{rmgm}} := [(D^{rmgm})^1, \ldots, (D^{rmgm})^m]$ (6) return D^{rmgm}. For the learning algorithm $A_2^{rmgm}(D^{\text{rmgm}})$: (1) $[X^{\text{rmgm}}, Y^{\text{rmgm}}]$ (2) $\widehat{H}_n^{rmgm} := \frac{1}{n}(X^{rmgm})^{\mathrm{T}} X^{rmgm}$ (3) $\widehat{w}_n^{rmgm} := (\widehat{H}_n^{rmgm})^{-1}\frac{1}{n}(X^{rmgm})^{\mathrm{T}} Y^{rmgm}$. (4) return $\widehat{w}_n^{rmgm}$.

[0051] As compared with the first improved DP algorithm, the second improved DP algorithm solves the near-zero eigenvalue issue because $\widehat{H}_n^{rmgm} \succcurlyeq 0$ holds naturally by definition. Although the order is sacrificed from n to $\sqrt{n}$, the second improved DP algorithm still outperforms the first improved DP algorithm on both synthetic datasets (even when n is as large as 10^6) as well as various real-world datasets. The performances of the first and second improved DP algorithms are discussed in more detail below.

[0052] The first and second improved DP algorithms were evaluated on both synthetic and real-world datasets. The performance evaluations with regard to the synthetic datasets were designed

to verify the theoretical asymptotic results by increasing the training set size n. The performance of the two improved algorithms were also evaluated on four real world datasets.

[0053] To evaluate the second improved DP algorithm, k was set as $\frac{\sqrt{n}}{\sigma_{\varepsilon,\delta}}$ for synthetic dataset experiments and k was set as $\frac{3\sqrt{n}}{\sigma_{\varepsilon,\delta}}$ for real-world dataset experiments. For the numerical computation stability of the inverse operation, a small $\lambda \cdot I$ may be added with $\lambda=10^{-5}$ to all Hessian matrices. Additionally, the following baselines were considered to help qualify the performance of the improved DP algorithms: OLS refers to the explicit solution for linear regression given training data (X, Y) and serves as the performance's upper bound for private algorithms. Biased Gaussian mechanism (BGM-OLS) refers to the algorithm that under the same $A_1^{bgm} = A_1^{dgm}$, $D^{\text{bgm}} := A_1^{bgm}(D^1, \ldots, D^m)$ is taken as the new feature matrix and the explicit form of the solution may be directly applied on top of (X^{bgm}, Y^{bgm}). A_2^{bgm} is defined as $((X^{\text{bgm}})^{\text{T}} X^{\text{bgm}})^{-1} (X^{\text{bgm}})^{\text{T}} Y^{\text{bgm}}$. Compared to A_2^{dgm}, $4d\sigma_{\varepsilon,\delta}^2 \cdot I$ is not subtracted when estimating the Hessian matrix in A_2^{bgm}.

[0054] In the experiments on synthetic datasets, the probability of the ℓ^2 distance between the linear weights from each algorithm or baseline and the ground truth linear weight were estimated. The expectation of the ℓ^2 distance between weights for different algorithms was also evaluated. For the experiments on real world datasets, the two improved DP algorithms were evaluated by the population loss, i.e., mean squared loss on the test set.

[0055] First, with regard to evaluation on synthetic datasets, to simulate the data, feature dimension d was set as d=10. To generate the ground truth linear model $\mathbf{w}^*$, each weight value was independently sampled from uniform distribution U $[-\frac{1}{d}, \frac{1}{d}]$. Each data ($\mathbf{x}$, y) is sampled as following: each feature value in $\mathbf{x}$ is independently sampled from U [-1, 1] and $y = (\mathbf{w}^*)^{\text{T}}\mathbf{x}$. Both assumptions described above can be easily verified. For the first assumption that the absolute values of all attributes are bounded by 1, $\|\mathbf{x}_i\| \leq 1$ holds by the definition and $\|y\| \leq \|\mathbf{x}\| \cdot \|\mathbf{w}^*\| \leq \sqrt{d} \cdot \frac{1}{\sqrt{d}} = 1$. For the no perfect multicollinearity assumption $\mathrm{E}[\mathbf{x}\mathbf{x}^{\text{T}}] = \frac{1}{3} \cdot I$, which is invertible.

[0056] Different training set sizes were experimented with. For example, training set sizes of n=10^4, 3 x 10^4, 10^5, 3 x 10^5, and 10^6 and different privacy budgets ε =1, 3, 10 with fixed $\delta = 10^{-5}$. For each setting (n, ε), the experiment was repeated 10^4 times under different random seeds to estimate the $\mathbb{P}$ $[\|\mathbf{w}_{\text{n}} - \mathbf{w}^*\| \leq \beta]$ for different algorithms. FIG. 4 shows a series of graphs 400 that depict the convergence of $\mathbb{P}$ $[\|\mathbf{w}_{\text{n}} - \mathbf{w}^*\| \leq \beta]$ as the training data set size n increased for different algorithms. Rows are differential privacy budget ε and columns are convergences at different β in $\mathbb{P}$ $[\|\mathbf{w}_{\text{n}} - \mathbf{w}^*\| \leq \beta]$.

[0057] Regarding the two baselines outlined above, OLS solutions, without any private constraint, are close to the ground truth $\mathbf{w}^*$ under all β with probability 1. For all β, the probability for BGM-OLS stays very low as n increases. $\operatorname{plim}_{n\to\infty} \frac{1}{n}(Z^{dgm})^{\mathrm{T}} Z^{dgm} = \mathrm{E}_{\mathrm{x}}[\mathbf{x}\mathbf{x}^{\mathrm{T}}] + 4\mathrm{d}\sigma_{\varepsilon,\delta}^2 \cdot I$, which introduces a non-diminishing bias $4\mathrm{d}\sigma_{\varepsilon,\delta}^2 \cdot I$. Although it is in the form of ℓ^2 regularization, $4\mathrm{d}\sigma_{\varepsilon,\delta}^2$ is larger than the reasonable regularization term and has a negative impact on the solution. Usually, the regularization is around 10^{-5}, while $4\mathrm{d}\sigma_{\varepsilon,\delta}^2 \approx 10$ even if $(\varepsilon, \delta) = (10, 10^{-5})$.

[0058] The performances of the first and second improved DP algorithms may be compared. The first improved DP algorithm shows asymptotic tendencies for most (ε, β), while the second improved DP algorithm shows such tendences in few cases. The second improved DP algorithm outperforms the first improved DP algorithm at both the convergence of probability $\mathbb{P}[\|\mathbf{w}_{\mathrm{n}} - \mathbf{w}^*\| \leq \beta]$ (the first three columns in the series of graphs 400) and the expected distance $\mathrm{E}[\|\mathbf{w}_{\mathrm{n}} - \mathbf{w}^*\|]$ (the last column in the series of graphs 400). The second improved DP algorithm shows the convergences in all settings except for (ε, β) = (1, 0.2) and (1, 0.05). Although the first DP algorithm has better rate at n than the second DP algorithm, $\mathbb{P}[\|\mathbf{w}_{\mathrm{n}} - \mathbf{w}^*\| \leq \beta]$ is always close to zero when ε=1, 3, and even when n is as large as 10^6.

[0059] Moreover, in the evaluation of expected weight distance, the first improved DP algorithm sometimes is even worse than BGM-OLS, which is almost random guess. This may be caused by the small eigenvalue issue discussed above. Figure 5 shows a scatter plot 500 of ℓ^2 distance vs. minimum absolute eigenvalue of Hessian matrix when n=10^6 and ε=3. Each point is processed by a different random seed for the first and second improved DP algorithms. Small eigenvalues may be highly correlated to large $\|\widehat{\mathbf{w}}_n - \mathbf{w}^*\|$. The x-axis in the scatter plot 500 is represents minimum eigenvalues of the Hessian matrix and the y-axis represents the distance between the solution and the optimal solution. Each point is processed by a different random seed for the first improved DP algorithm and BGM-OLS when n=10^6 and ε =3. $\|\widehat{\mathbf{w}}_n - \mathbf{w}^*\|$ and the minimum absolute eigenvalues of $\widehat{H}_n$ have a strong positive correlation. Some samples from the first improved DP algorithm have relatively small eigenvalues ($\leq 10^{-3}$) and have corresponding large distance between $\widehat{\mathbf{w}}_n$ and $\mathbf{w}^*$ ($\geq$10). Overall, the second improved DP algorithm has the best performance across various settings of ε and n on the synthetic data, as its asymptotically optimality is verified and its consistently outperforms the first improved DP algorithm and BGN-OLS.

[0060] Second, with regard to evaluation on real world datasets, the first and second improved DP algorithms were evaluated on four real-world datasets: a bike dataset to predict the count of rental bikes from features such as season, holiday, etc., a superconductor dataset to predict critical temperature from chemical features; a GPU dataset to predict running time for multiplying two

2048 x 2048 matrices using a GPU OpenCL SGEMM kernel with varying parameters, and a music song dataset to predict the release year of a song from audio features. The original dataset was split into training data and test data by the ratio 4:1. The number of training data n and the number of features d used for later evaluated linear regression tasks are listed in a table 600 depicted in FIG. 6. All features and labels are normalized into [0,1]. The table 600 depicts mean squared losses on real world datasets (lower is better). As shown in the table 600, the first improved DP algorithm achieves the lowest losses in most settings

[0061] For each dataset, OLS and three differentially private algorithms were evaluated by the mean squared loss on the test split. The table 600 shows the results for $\varepsilon = 1, 3, 10$ and $\delta = 10^{-5}$. The loss of the first improved DP algorithm is much larger than others and the second improved DP algorithm achieves the lowest losses for most cases (9 out of 12). This result is consistent with the results on the synthetic dataset. Accordingly, the second improved DP algorithm is a practical solution to privately release datasets and build linear models for regression tasks.

[0062] FIG. 7 illustrates an example process 700 performed by a differential privacy model (e.g., differential privacy model 202a-n). The differential privacy model may perform the process 700 to perform the first improved DP algorithm described above. Although depicted as a sequence of operations in FIG. 7, those of ordinary skill in the art will appreciate that various embodiments may add, remove, reorder, or modify the depicted operations.

[0063] As described above, the first improved DP algorithm is a De-biased Gaussian Mechanism for Ordinary Least Squares (DGM-OLS). To perform the first DP algorithm, the differential privacy models 202a-n may add Gaussian noise directly to data 106a-n belonging to the respective data provider 102a-n. For example, the differential privacy model 202a may add Gaussian noise directly to data 106a belonging to the data provider 102a, etc. At 702, noise may be added by applying a random Gaussian matrix to a dataset to obtain a processed dataset. The dataset is owned by a party (e.g., a data provider 102a) among a plurality of parties (e.g., data providers 102a-n). There may be a plurality of datasets owned by the plurality of parties. The processed dataset ensures data privacy protection

[0064] After adding Gaussian noise to the dataset, the differential privacy models 202a-n may continue to perform the first DP algorithm by removing certain biases from the resulting Hessian matrix. At 704, at least one bias may be removed from the processed dataset. After the bias(es) are removed, the processed dataset is ready to be released to other parties (e.g., other data providers 102a-n). For example, the processed dataset may satisfy (ε, δ)-differential privacy. At 706, the processed dataset may be released to other parties. The other parties may be from the plurality of parties, or may be different parties. Once released to other parties, the other parties may utilize the

data without being able to infer the attributes or identity of any individual data sample. For example, the other parties may utilize the data to train one or more machine learning models.

[0065] FIG. 8 illustrates an example process 800 performed by a differential privacy model (e.g., differential privacy model 202a-n). The differential privacy model may perform the process 800 to perform the second improved DP algorithm described above. Although depicted as a sequence of operations in FIG. 8, those of ordinary skill in the art will appreciate that various embodiments may add, remove, reorder, or modify the depicted operations.

[0066] As described above, the second improved DP algorithm is a Random Mixing prior to Gaussian Mechanism for Ordinary Least Squares (RMGM-OLS). To perform the second DP algorithm, all of the differential privacy models 202a-n may utilize a shared Bernoulli projection matrix. All of the differential privacy models 202a-n may utilize the shared Bernoulli projection matrix to compress the data 106a-n belonging to the respective data provider 102a-n along a sample-wise dimension. At 802, a dataset may be compressed by applying a first random matrix. The dataset is owned by a party (e.g., data provider 102a) among a plurality of parties (e.g., data providers 102a-n). There may be a plurality of datasets owned by the plurality of parties. The first random matrix may be a Bernoulli projection matrix. The elements of the Bernoulli projection matrix may include independent random variables Aij sampled from a distribution, wherein Aij is equal to positive one with a probability of ½ and wherein Aij is equal to negative one with a probability of ½. The first random matrix may be shared among the plurality of parties for compressing their respective datasets.

[0067] After compressing the dataset, Gaussian noise made be added to the compressed data. At 804, a noise may be added by applying a random Gaussian matrix to the compressed dataset to obtain a processed dataset. The random Gaussian matrix may comprise elements sampled from a Gaussian distribution with mean 0 and variance of $4d\sigma_{\varepsilon,\delta}^2$. The processed dataset ensures data privacy protection. For example, the processed dataset may satisfy (ε, δ)-differential privacy. The processed dataset may be denoted by $(D^{rmgm})^j$, and wherein $(D^{rmgm})^j$ is equal to a dot product of one divided by a square root of k and AD^j plus R^j, and wherein k represents a dimension of the compressed dataset, A represents the first random matrix, D^j, j = 1 … m, represents the dataset owned by the party among the plurality of m parties, $m \geq 2$, and R^j represents the random Gaussian matrix applied to the dataset.

[0068] At 806, the processed dataset may be released to other parties. The other parties may be from the plurality of parties or may be different parties. Once released to other parties, the other parties may utilize the data without being able to infer the attributes or identity of any individual

data sample. For example, the other parties may utilize the data to train one or more machine learning models.

[0069] FIG. 9 illustrates an example process 900 performed by a differential privacy model (e.g., differential privacy model 202a-n). The differential privacy model may perform the process 900 to perform the second improved DP algorithm described above. Although depicted as a sequence of operations in FIG. 9, those of ordinary skill in the art will appreciate that various embodiments may add, remove, reorder, or modify the depicted operations.

[0070] As described above, the first improved DP algorithm is a De-biased Gaussian Mechanism for Ordinary Least Squares (DGM-OLS). To perform the first DP algorithm, the differential privacy models 202a-n may add Gaussian noise directly to data 106a-n belonging to the respective data provider 102a-n. For example, the differential privacy model 202a may add Gaussian noise directly to data 106a belonging to the data provider 102a, etc. At 702, noise may be added by applying a random Gaussian matrix to a dataset to obtain a processed dataset. The dataset is owned by a party (e.g., a data provider 102a) among a plurality of parties (e.g., data providers 102a-n). There may be a plurality of datasets owned by the plurality of parties. The processed dataset ensures data privacy protection

[0071] After adding Gaussian noise to the dataset, the differential privacy models 202a-n may continue to perform the first DP algorithm by removing certain biases from the resulting Hessian matrix. At 704, at least one bias may be removed from the processed dataset. After the bias(es) are removed, the processed dataset is ready to be released to other parties (e.g., other data providers 102a-n). For example, the processed dataset may satisfy (ε, δ)-differential privacy. At 706, the processed dataset may be released to other parties. The other parties may be from the plurality of parties, or may be different parties. Once released to other parties, the other parties may utilize the data without being able to infer the attributes or identity of any individual data sample. For example, the other parties may utilize the data to train one or more machine learning models.

[0072] FIG. 8 illustrates an example process 800 performed by a differential privacy model (e.g., differential privacy model 202a-n). The differential privacy model may perform the process 800 to perform the second improved DP algorithm described above. Although depicted as a sequence of operations in FIG. 8, those of ordinary skill in the art will appreciate that various embodiments may add, remove, reorder, or modify the depicted operations.

[0073] As described above, the second improved DP algorithm is a Random Mixing prior to Gaussian Mechanism for Ordinary Least Squares (RMGM-OLS). To perform the second DP algorithm, all of the differential privacy models 202a-n may utilize a shared Bernoulli projection matrix. All of the differential privacy models 202a-n may utilize the shared Bernoulli projection

matrix to compress the data 106a-n belonging to the respective data provider 102a-n along a sample-wise dimension. At 902, a dataset may be compressed by applying a first random matrix. The dataset is owned by a party (e.g., data provider 102a) among a plurality of parties (e.g., data providers 102a-n). There may be a plurality of datasets owned by the plurality of parties. The first random matrix may be a Bernoulli projection matrix. The elements of the Bernoulli projection matrix may include independent random variables Aij sampled from a distribution, wherein Aij is equal to positive one with a probability of ½ and wherein Aij is equal to negative one with a probability of ½. The first random matrix may be shared among the plurality of parties for compressing their respective datasets.

[0074] After compressing the dataset, Gaussian noise made be added to the compressed data. At 904, a noise may be added by applying a random Gaussian matrix to the compressed dataset to obtain a processed dataset. The random Gaussian matrix may comprise elements sampled from a Gaussian distribution with mean 0 and variance of $4d\sigma_{\varepsilon,\delta}^2$. The processed dataset ensures data privacy protection. For example, the processed dataset may satisfy (ε, δ)-differential privacy. The processed dataset may be denoted by $(D^{rmgm})^j$, and wherein $(D^{rmgm})^j$ is equal to a dot product of one divided by a square root of k and AD^j plus R^j, and wherein k represents a dimension of the compressed dataset, A represents the first random matrix, D^j, j = 1 … m, represents the dataset owned by the party among the plurality of m parties, m ≥ 2, and R^j represents the random Gaussian matrix applied to the dataset.

[0075] At 906, the processed dataset may be released to other parties. The other parties may be from the plurality of parties or may be different parties. Once released to other parties, the other parties may utilize the data without being able to infer the attributes or identity of any individual data sample. For example, the other parties may utilize the data to train one or more machine learning models.

[0076] At 908, other processed datasets owned by the other parties among the plurality of parties may be obtained. The other processed datasets also ensure data privacy protection.

At 910, a linear model may be trained on complete data comprising the processed datasets and the other processed datasets. The linear model may be trained by applying a learning algorithm from the complete data. The learning algorithm from the complete data may be denoted as $A_2^{rmgm}(D^{rmgm})$, and wherein $A_2^{rmgm}(D^{rmgm})$ is equal to one divided by n, multiplied by one divided by $\widehat{H}_n^{rmgm}$, multiplied by a transpose of X^{rmgm}, multiplied by Y^{rmgm}, and wherein D^{rmgm} represents a collection of all processed datasets, n represents a size of a dataset $\widehat{H}_n^{rmgm}$, $\widehat{H}_n^{rmgm}$ is equal to one divided by n, multiplied by a transpose of X^{rmgm}, multiplied by X^{rmgm}, X^{rmgm}

represents a feature matrix from the processed datasets, and Y^{rmgm} represents label vectors from the processed datasets.

[0077] Many recent works study differentially private data release algorithms. However, those algorithms either only serve for data release from a single-party, or focus on the feature dimension reduction or empirical improvement, which is orthogonal to the study of asymptotical optimality with respect to dataset size. The random Gaussian projection matrices in these methods contribute to the differential privacy guarantee, hence the sharing of a projection matrix would violate the privacy guarantee between parties. Nevertheless, without sharing this projection matrix, the utility cannot be guaranteed. These methods may train a differentially private GAN. However, it is not obvious to privately share data information during their training when each party holds different attributes but same instances. These methods propose a random mixing method and also analyze the linear model. However, they mix only works for realizable linear data, and this is not able to be extended to the general linear regression and the asymptotic optimality guarantee. Some of these methods focus on feature dimension reduction, which is orthogonal to the study of asymptotical optimality with respect to dataset size.

[0078] A large amount of works study differentially private optimization for convex problems or for linear regression. However, they mainly differ from the techniques described herein in the sense that their goal is to release the final model while the goal of the techniques described herein is to release the dataset.

[0079] Linear regression is a fundamental machine learning task, and there have been some works studying linear regression over vertically partitioned datasets based on secure multi-party computation. However, cryptographic protocols such as Homomorphic Encryption and garbled circuits lead to heavy overhead on computation and communication. From this aspect, DP-based techniques are more practical.

[0080] In conclusion, the techniques described herein involve two differentially private algorithms under a multi-party setting for linear regression. Each solution is asymptotically optimal with increasing dataset size. It is empirically verified that the second DP algorithm described herein has the best performance on both synthetic datasets and real-world datasets. While the techniques described herein focus on linear regression only, it should be appreciated that the techniques described herein could instead be extended to classification, e.g., logistic regression, while achieving the same asymptotic optimality. In addition, while the techniques described herein assume that different parties (e.g., different data providers) own the same set of data subjects, it should be appreciated that in some embodiments, the set of subjects owned by different parties might be slightly different.

[0081] FIG. 10 illustrates a computing device that may be used in various aspects, such as the networks, models, and/or devices depicted in FIG. 2. With regard to the example architecture of FIG. 2, the data providers 102a-n, the differential privacy models 202a-n, the network 120, and/or the computing devices 116a-n may each be implemented by one or more instance of a computing device 1000 of FIG. 10. The computer architecture shown in FIG. 10 shows a conventional server computer, workstation, desktop computer, laptop, tablet, network appliance, PDA, e-reader, digital cellular phone, or other computing node, and may be utilized to execute any aspects of the computers described herein, such as to implement the methods described herein.

[0082] The computing device 1000 may include a baseboard, or "motherboard," which is a printed circuit board to which a multitude of components or devices may be connected by way of a system bus or other electrical communication paths. One or more central processing units (CPUs) 1004 may operate in conjunction with a chipset 1006. The CPU(s) 1004 may be standard programmable processors that perform arithmetic and logical operations necessary for the operation of the computing device 1000.

[0083] The CPU(s) 1004 may perform the necessary operations by transitioning from one discrete physical state to the next through the manipulation of switching elements that differentiate between and change these states. Switching elements may generally include electronic circuits that maintain one of two binary states, such as flip-flops, and electronic circuits that provide an output state based on the logical combination of the states of one or more other switching elements, such as logic gates. These basic switching elements may be combined to create more complex logic circuits including registers, adders-subtractors, arithmetic logic units, floating-point units, and the like.

[0084] The CPU(s) 1004 may be augmented with or replaced by other processing units, such as GPU(s) 1005. The GPU(s) 1005 may comprise processing units specialized for but not necessarily limited to highly parallel computations, such as graphics and other visualization-related processing.

[0085] A chipset 1006 may provide an interface between the CPU(s) 1004 and the remainder of the components and devices on the baseboard. The chipset 1006 may provide an interface to a random-access memory (RAM) 1008 used as the main memory in the computing device 1000. The chipset 1006 may further provide an interface to a computer-readable storage medium, such as a read-only memory (ROM) 1020 or non-volatile RAM (NVRAM) (not shown), for storing basic routines that may help to start up the computing device 1000 and to transfer information between the various components and devices. ROM 1020 or NVRAM may also store other software

components necessary for the operation of the computing device 1000 in accordance with the aspects described herein.

[0086] The computing device 1000 may operate in a networked environment using logical connections to remote computing nodes and computer systems through local area network (LAN). The chipset 1006 may include functionality for providing network connectivity through a network interface controller (NIC) 1022, such as a gigabit Ethernet adapter. A NIC 1022 may be capable of connecting the computing device 1000 to other computing nodes over a network 1016. It should be appreciated that multiple NICs 1022 may be present in the computing device 1000, connecting the computing device to other types of networks and remote computer systems.

[0087] The computing device 1000 may be connected to a mass storage device 1028 that provides non-volatile storage for the computer. The mass storage device 1028 may store system programs, application programs, other program modules, and data, which have been described in greater detail herein. The mass storage device 1028 may be connected to the computing device 1000 through a storage controller 1024 connected to the chipset 1006. The mass storage device 1028 may consist of one or more physical storage units. The mass storage device 1028 may comprise a management component 1010. A storage controller 1024 may interface with the physical storage units through a serial attached SCSI (SAS) interface, a serial advanced technology attachment (SATA) interface, a fiber channel (FC) interface, or other type of interface for physically connecting and transferring data between computers and physical storage units.

[0088] The computing device 1000 may store data on the mass storage device 1028 by transforming the physical state of the physical storage units to reflect the information being stored. The specific transformation of a physical state may depend on various factors and on different implementations of this description. Examples of such factors may include, but are not limited to, the technology used to implement the physical storage units and whether the mass storage device 1028 is characterized as primary or secondary storage and the like.

[0089] For example, the computing device 1000 may store information to the mass storage device 1028 by issuing instructions through a storage controller 1024 to alter the magnetic characteristics of a particular location within a magnetic disk drive unit, the reflective or refractive characteristics of a particular location in an optical storage unit, or the electrical characteristics of a particular capacitor, transistor, or other discrete component in a solid-state storage unit. Other transformations of physical media are possible without departing from the scope and spirit of the present description, with the foregoing examples provided only to facilitate this description. The computing device 1000 may further read information from the mass storage device 1028 by

detecting the physical states or characteristics of one or more particular locations within the physical storage units.

[0090] In addition to the mass storage device 1028 described above, the computing device 1000 may have access to other computer-readable storage media to store and retrieve information, such as program modules, data structures, or other data. It should be appreciated by those skilled in the art that computer-readable storage media may be any available media that provides for the storage of non-transitory data and that may be accessed by the computing device 1000.

[0091] By way of example and not limitation, computer-readable storage media may include volatile and non-volatile, transitory computer-readable storage media and non-transitory computer-readable storage media, and removable and non-removable media implemented in any method or technology. Computer-readable storage media includes, but is not limited to, RAM, ROM, erasable programmable ROM ("EPROM"), electrically erasable programmable ROM ("EEPROM"), flash memory or other solid-state memory technology, compact disc ROM ("CD-ROM"), digital versatile disk ("DVD"), high definition DVD ("HD-DVD"), BLU-RAY, or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage, other magnetic storage devices, or any other medium that may be used to store the desired information in a non-transitory fashion.

[0092] A mass storage device, such as the mass storage device 1028 depicted in FIG. 10, may store an operating system utilized to control the operation of the computing device 1000. The operating system may comprise a version of the LINUX operating system. The operating system may comprise a version of the WINDOWS SERVER operating system from the MICROSOFT Corporation. According to further aspects, the operating system may comprise a version of the UNIX operating system. Various mobile phone operating systems, such as IOS and ANDROID, may also be utilized. It should be appreciated that other operating systems may also be utilized. The mass storage device 1028 may store other system or application programs and data utilized by the computing device 1000.

[0093] The mass storage device 1028 or other computer-readable storage media may also be encoded with computer-executable instructions, which, when loaded into the computing device 1000, transforms the computing device from a general-purpose computing system into a special-purpose computer capable of implementing the aspects described herein. These computer-executable instructions transform the computing device 1000 by specifying how the CPU(s) 1004 transition between states, as described above. The computing device 1000 may have access to computer-readable storage media storing computer-executable instructions, which, when executed by the computing device 1000, may perform the methods described herein.

[0094] A computing device, such as the computing device 1000 depicted in FIG. 10, may also include an input/output controller 1032 for receiving and processing input from a number of input devices, such as a keyboard, a mouse, a touchpad, a touch screen, an electronic stylus, or other type of input device. Similarly, an input/output controller 1032 may provide output to a display, such as a computer monitor, a flat-panel display, a digital projector, a printer, a plotter, or other type of output device. It will be appreciated that the computing device 1000 may not include all of the components shown in FIG. 10, may include other components that are not explicitly shown in FIG. 10, or may utilize an architecture completely different than that shown in FIG. 10.

[0095] As described herein, a computing device may be a physical computing device, such as the computing device 1000 of FIG. 10. A computing node may also include a virtual machine host process and one or more virtual machine instances. Computer-executable instructions may be executed by the physical hardware of a computing device indirectly through interpretation and/or execution of instructions stored and executed in the context of a virtual machine.

[0096] It is to be understood that the methods and systems are not limited to specific methods, specific components, or to particular implementations. It is also to be understood that the terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting.

[0097] As used in the specification and the appended claims, the singular forms "a," "an," and "the" include plural referents unless the context clearly dictates otherwise. Ranges may be expressed herein as from "about" one particular value, and/or to "about" another particular value. When such a range is expressed, another embodiment includes from the one particular value and/or to the other particular value. Similarly, when values are expressed as approximations, by use of the antecedent "about," it will be understood that the particular value forms another embodiment. It will be further understood that the endpoints of each of the ranges are significant both in relation to the other endpoint, and independently of the other endpoint.

[0098] "Optional" or "optionally" means that the subsequently described event or circumstance may or may not occur, and that the description includes instances where said event or circumstance occurs and instances where it does not.

[0099] Throughout the description and claims of this specification, the word "comprise" and variations of the word, such as "comprising" and "comprises," means "including but not limited to," and is not intended to exclude, for example, other components, integers or steps. "Exemplary" means "an example of" and is not intended to convey an indication of a preferred or ideal embodiment. "Such as" is not used in a restrictive sense, but for explanatory purposes.

[00100] Components are described that may be used to perform the described methods and systems. When combinations, subsets, interactions, groups, etc., of these components are described, it is understood that while specific references to each of the various individual and collective combinations and permutations of these may not be explicitly described, each is specifically contemplated and described herein, for all methods and systems. This applies to all aspects of this application including, but not limited to, operations in described methods. Thus, if there are a variety of additional operations that may be performed it is understood that each of these additional operations may be performed with any specific embodiment or combination of embodiments of the described methods.

[00101] The present methods and systems may be understood more readily by reference to the following detailed description of preferred embodiments and the examples included therein and to the Figures and their descriptions.

[00102] As will be appreciated by one skilled in the art, the methods and systems may take the form of an entirely hardware embodiment, an entirely software embodiment, or an embodiment combining software and hardware aspects. Furthermore, the methods and systems may take the form of a computer program product on a computer-readable storage medium having computer-readable program instructions (e.g., computer software) embodied in the storage medium. More particularly, the present methods and systems may take the form of web-implemented computer software. Any suitable computer-readable storage medium may be utilized including hard disks, CD-ROMs, optical storage devices, or magnetic storage devices.

[00103] Embodiments of the methods and systems are described below with reference to block diagrams and flowchart illustrations of methods, systems, apparatuses and computer program products. It will be understood that each block of the block diagrams and flowchart illustrations, and combinations of blocks in the block diagrams and flowchart illustrations, respectively, may be implemented by computer program instructions. These computer program instructions may be loaded on a general-purpose computer, special-purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions which execute on the computer or other programmable data processing apparatus create a means for implementing the functions specified in the flowchart block or blocks.

[00104] These computer program instructions may also be stored in a computer-readable memory that may direct a computer or other programmable data processing apparatus to function in a particular manner, such that the instructions stored in the computer-readable memory produce an article of manufacture including computer-readable instructions for implementing the function specified in the flowchart block or blocks. The computer program instructions may also be loaded

onto a computer or other programmable data processing apparatus to cause a series of operational steps to be performed on the computer or other programmable apparatus to produce a computer-implemented process such that the instructions that execute on the computer or other programmable apparatus provide steps for implementing the functions specified in the flowchart block or blocks.

[00105] The various features and processes described above may be used independently of one another, or may be combined in various ways. All possible combinations and sub-combinations are intended to fall within the scope of this disclosure. In addition, certain methods or process blocks may be omitted in some implementations. The methods and processes described herein are also not limited to any particular sequence, and the blocks or states relating thereto may be performed in other sequences that are appropriate. For example, described blocks or states may be performed in an order other than that specifically described, or multiple blocks or states may be combined in a single block or state. The example blocks or states may be performed in serial, in parallel, or in some other manner. Blocks or states may be added to or removed from the described example embodiments. The example systems and components described herein may be configured differently than described. For example, elements may be added to, removed from, or rearranged compared to the described example embodiments.

[00106] It will also be appreciated that various items are illustrated as being stored in memory or on storage while being used, and that these items or portions thereof may be transferred between memory and other storage devices for purposes of memory management and data integrity. Alternatively, in other embodiments, some or all of the software modules and/or systems may execute in memory on another device and communicate with the illustrated computing systems via inter-computer communication. Furthermore, in some embodiments, some or all of the systems and/or modules may be implemented or provided in other ways, such as at least partially in firmware and/or hardware, including, but not limited to, one or more application-specific integrated circuits ("ASICs"), standard integrated circuits, controllers (e.g., by executing appropriate instructions, and including microcontrollers and/or embedded controllers), field-programmable gate arrays ("FPGAs"), complex programmable logic devices ("CPLDs"), etc. Some or all of the modules, systems, and data structures may also be stored (e.g., as software instructions or structured data) on a computer-readable medium, such as a hard disk, a memory, a network, or a portable media article to be read by an appropriate device or via an appropriate connection. The systems, modules, and data structures may also be transmitted as generated data signals (e.g., as part of a carrier wave or other analog or digital propagated signal) on a variety of computer-readable transmission media, including wireless-based and wired/cable-based media,

and may take a variety of forms (e.g., as part of a single or multiplexed analog signal, or as multiple discrete digital packets or frames). Such computer program products may also take other forms in other embodiments. Accordingly, the present invention may be practiced with other computer system configurations.

[00107] While the methods and systems have been described in connection with preferred embodiments and specific examples, it is not intended that the scope be limited to the particular embodiments set forth, as the embodiments herein are intended in all respects to be illustrative rather than restrictive.

[00108] Unless otherwise expressly stated, it is in no way intended that any method set forth herein be construed as requiring that its operations be performed in a specific order. Accordingly, where a method claim does not actually recite an order to be followed by its operations or it is not otherwise specifically stated in the claims or descriptions that the operations are to be limited to a specific order, it is no way intended that an order be inferred, in any respect. This holds for any possible non-express basis for interpretation, including: matters of logic with respect to arrangement of steps or operational flow; plain meaning derived from grammatical organization or punctuation; and the number or type of embodiments described in the specification.

[00109] It will be apparent to those skilled in the art that various modifications and variations may be made without departing from the scope or spirit of the present disclosure. Other embodiments will be apparent to those skilled in the art from consideration of the specification and practices described herein. It is intended that the specification and example figures be considered as exemplary only, with a true scope and spirit being indicated by the following claims.

WHAT IS CLAIMED IS:

1. A method of releasing data while protecting individual privacy, comprising:

compressing a dataset by applying a first random matrix, wherein the dataset is owned by a party among a plurality of parties, and there are a plurality of datasets owned by the plurality of parties;

adding a noise by applying a random Gaussian matrix to the compressed dataset to obtain a processed dataset, wherein the processed dataset ensures data privacy protection; and

releasing the processed dataset to other parties.

2. The method of claim 1, wherein the first random matrix is a Bernoulli projection matrix, and wherein the Bernoulli projection matrix comprises independent random variables A_{ij} sampled from a distribution $Aij = \begin{cases} +1 & with\ probability\ 1/2 \\ -1 & with\ probability\ 1/2 \end{cases}$.

3. The method of any of claims 1-2, wherein the first random matrix is shared among the plurality of parties for compressing their respective datasets.

4. The method of any of claims 1-3, wherein the random Gaussian matrix comprises elements sampled from a Gaussian distribution with mean 0 and variance of $4d\sigma_{\varepsilon,\delta}^2$.

5. The method of any of claims 1-4, wherein the processed dataset is denoted by $(D^{rmgm})^j$, and wherein $(D^{rmgm})^j$ is equal to $\frac{1}{\sqrt{k}} \cdot AD^j + R^j$, and wherein k represents a dimension of the compressed dataset, A represents the first random matrix, D^j, j = 1 … m, represents the dataset owned by the party among the plurality of m parties, m ≥ 2, and R^j represents the random Gaussian matrix applied to the dataset.

6. The method of any of claims 1-5, further comprising:

obtaining other processed datasets owned by the other parties among the plurality of parties, wherein the other processed datasets also ensure data privacy protection; and

training a linear model on complete data comprising the processed datasets and the other processed datasets.

7. The method of claim 6, further comprising:

training the linear model by applying a learning algorithm from the complete data, wherein the learning algorithm from the complete data is denoted as $A_2^{rmgm}(D^{rmgm})$, wherein $A_2^{rmgm}(D^{rmgm})$ is defined as $A_2^{rmgm}(D^{rmgm}) = \frac{1}{n}(\widehat{H}_n^{rmgm})^{-1}(X^{rmgm})^T Y^{rmgm}$, wherein D^{rmgm} represents a collection of all processed datasets, n represents a size of the complete data, $\widehat{H}_n^{rmgm} := \frac{1}{n}(X^{rmgm})^T X^{rmgm}$, X^{rmgm} represents a feature matrix from the processed datasets, and Y^{rmgm} represents label vectors from the processed datasets.

8. A system, comprising:

at least one processor; and

at least one memory communicatively coupled to the at least one processor and storing instructions that upon execution by the at least one processor cause the system to perform operations, the operations comprising:

compressing a dataset by applying a first random matrix, wherein the dataset is owned by a party among a plurality of parties, and there are a plurality of datasets owned by the plurality of parties;

adding a noise by applying a random Gaussian matrix to the compressed dataset to obtain a processed dataset, wherein the processed dataset ensures data privacy protection; and

releasing the processed dataset to other parties.

9. The system of claim 8, wherein the first random matrix is a Bernoulli projection matrix, and wherein the Bernoulli projection matrix comprises independent random variables A_{ij} sampled from a distribution $Aij = \begin{cases} +1 & with\ probability\ 1/2 \\ -1 & with\ probability\ 1/2 \end{cases}$.

10. The system of any of claims 8-9, wherein the first random matrix is shared among the plurality of parties for compressing their respective datasets.

11. The system of any of claims 8-10, wherein the random Gaussian matrix comprises elements sampled from a Gaussian distribution with mean 0 and variance of $4d\sigma_{\varepsilon,\delta}^2$.

12. The system of any of claims 8-11, wherein the processed dataset is denoted by $(D^{rmgm})^j$, and wherein $(D^{rmgm})^j$ is equal to $\frac{1}{\sqrt{k}} \cdot AD^j + R^j$, and wherein k represents a dimension of the compressed dataset, A represents the first random matrix, D^j, j = 1 … m, represents the dataset

owned by the party among the plurality of m parties, $m \geq 2$, and R^j represents the random Gaussian matrix applied to the dataset.

13. The system of any of claims 8-12, the operations further comprising:

obtaining other processed datasets owned by the other parties among the plurality of parties, wherein the other processed datasets also ensure data privacy protection; and

training a linear model on complete data comprising the processed datasets and the other processed datasets.

14. The system of claim 13, the operations further comprising:

training the linear model by applying a learning algorithm from the complete data, wherein the learning algorithm from the complete data is denoted as $A_2^{rmgm}(D^{rmgm})$, wherein $A_2^{rmgm}(D^{rmgm})$ is defined as $A_2^{rmgm}(D^{rmgm}) = \frac{1}{n}(\widehat{H}_n^{rmgm})^{-1}(X^{rmgm})^T Y^{rmgm}$, wherein D^{rmgm} represents a collection of all processed datasets, n represents a size of the complete data, $\widehat{H}_n^{rmgm} := \frac{1}{n}(X^{rmgm})^T X^{rmgm}$, X^{rmgm} represents a feature matrix from the processed datasets, and Y^{rmgm} represents label vectors from the processed datasets.

15. A non-transitory computer-readable storage medium, storing computer-readable instructions that upon execution by a processor cause the processor to implement operations, the operation comprising:

compressing a dataset by applying a first random matrix, wherein the dataset is owned by a party among a plurality of parties, and there are a plurality of datasets owned by the plurality of parties;

adding a noise by applying a random Gaussian matrix to the compressed dataset to obtain a processed dataset, wherein the processed dataset ensures data privacy protection; and

releasing the processed dataset to other parties.

16. The non-transitory computer-readable storage medium of claim 15, wherein the first random matrix is a Bernoulli projection matrix, and wherein the first random matrix is shared among the plurality of parties for compressing their respective datasets.

17. The non-transitory computer-readable storage medium of any of claims 15-16, wherein the random Gaussian matrix comprises elements sampled from a Gaussian distribution with mean 0 and variance of $4d\sigma_{\varepsilon,\delta}^2$.

18. The non-transitory computer-readable storage medium of any of claims 15-17, wherein the processed dataset is denoted by $(D^{rmgm})^j$, and wherein $(D^{rmgm})^j$ is equal to $\frac{1}{\sqrt{k}} \cdot AD^j + R^j$, and wherein k represents a dimension of the compressed dataset, A represents the first random matrix, D^j, j = 1 … m, represents the dataset owned by the party among the plurality of m parties, $m \geq 2$, and R^j represents the random Gaussian matrix applied to the dataset.

19. The non-transitory computer-readable storage medium of any of claims 15-18, the operations further comprising:

obtaining other processed datasets owned by the other parties among the plurality of parties, wherein the other processed datasets also ensure data privacy protection; and

training a linear model on complete data comprising the processed datasets and the other processed datasets.

20. The non-transitory computer-readable storage medium of claim 19, the operations further comprising:

training the linear model by applying a learning algorithm from the complete data, wherein the learning algorithm from the complete data is denoted as $A_2^{rmgm}(D^{rmgm})$, wherein $A_2^{rmgm}(D^{rmgm})$ is defined as $A_2^{rmgm}(D^{rmgm}) = \frac{1}{n}(\widehat{H}_n^{rmgm})^{-1}(X^{rmgm})^T Y^{rmgm}$, wherein D^{rmgm} represents a collection of all processed datasets, n represents a size of the complete data, $\widehat{H}_n^{rmgm} := \frac{1}{n}(X^{rmgm})^T X^{rmgm}$, X^{rmgm} represents a feature matrix from the processed datasets, and Y^{rmgm} represents label vectors from the processed datasets.

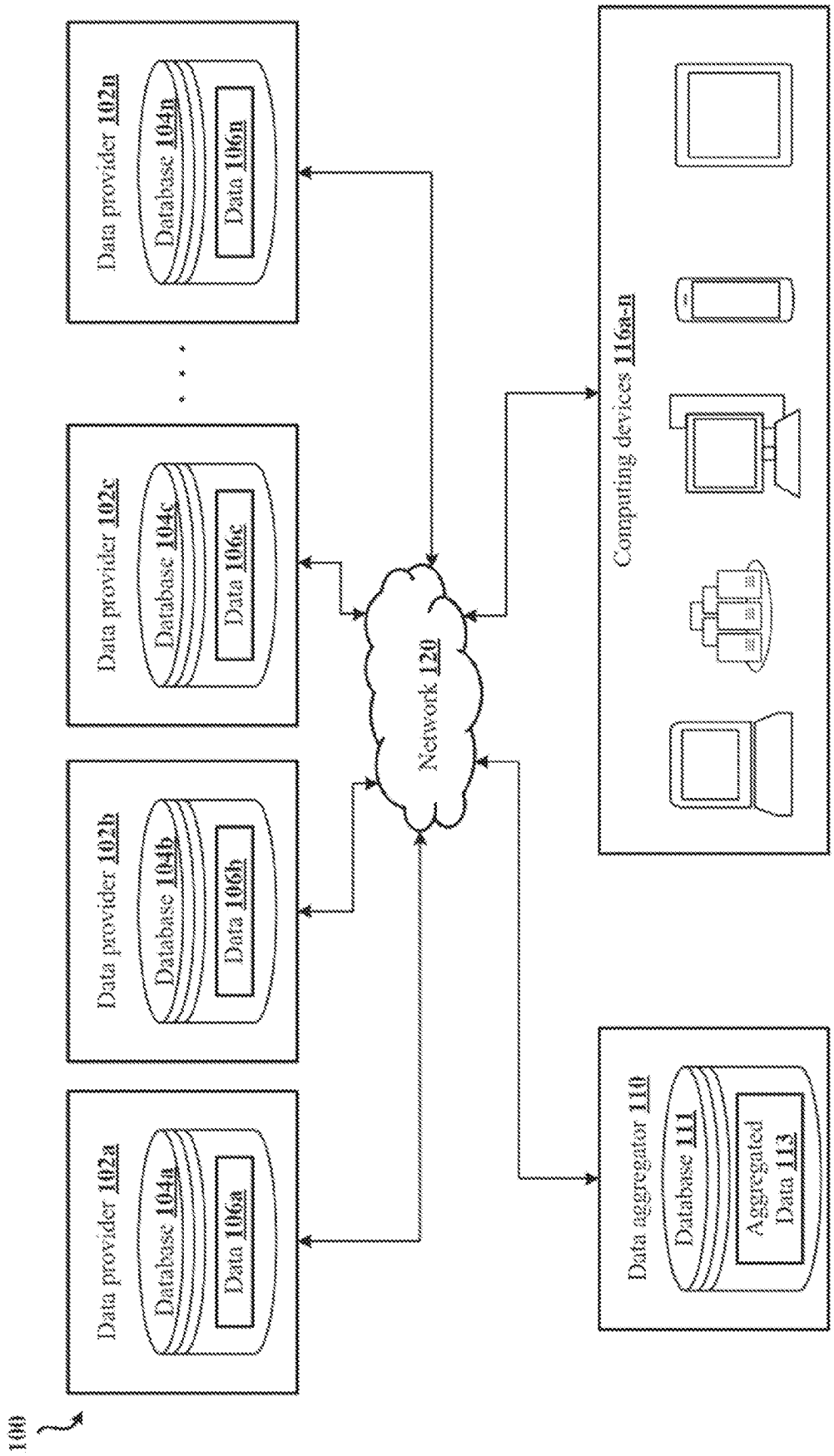

FIG. 1 PRIOR ART

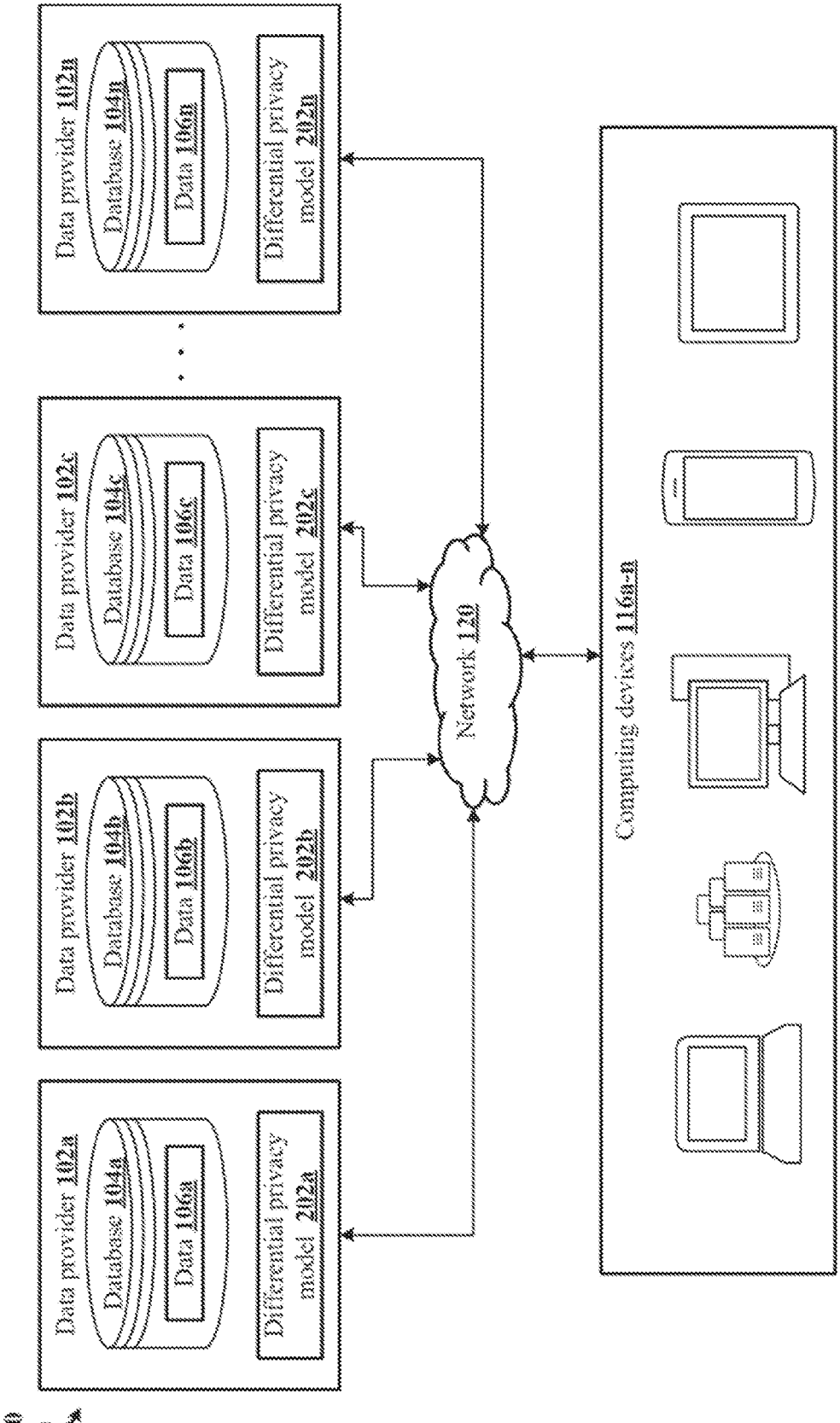
200
Data provider 102a
Database 104a
Data 106a
Differential privacy model 202a
Data provider 102b
Database 104b
Data 106b
Differential privacy model 202b
Data provider 102c
Database 104c
Data 106c
Differential privacy model 202c
Data provider 102n
Database 104n
Data 106n
Differential privacy model 202n
Network 120
Computing devices 116a-n

FIG. 2

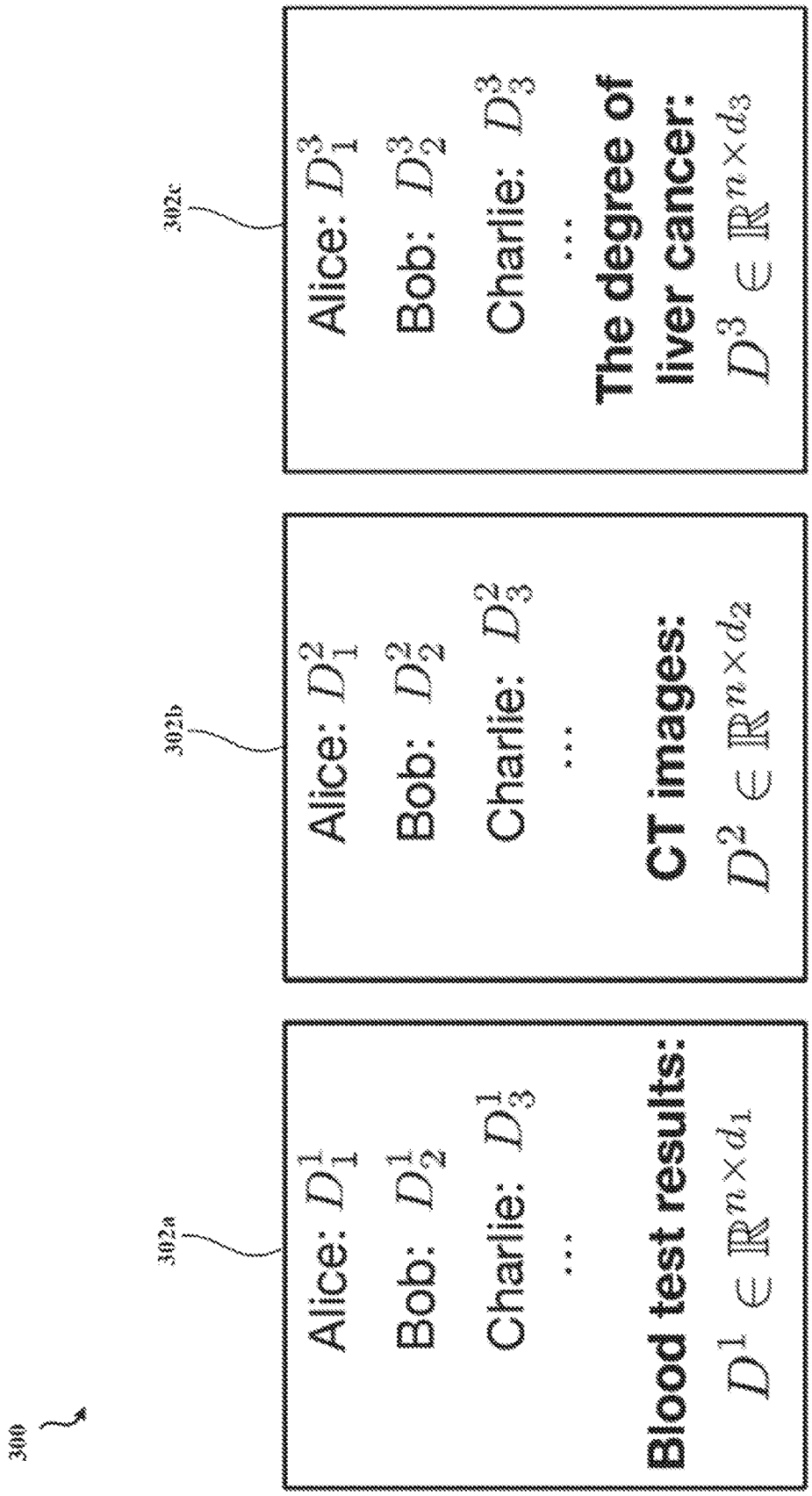
300
302a
Alice: D_1^1
Bob: D_2^1
Charlie: D_3^1
...
Blood test results:
$D^1 \in \mathbb{R}^{n \times d_1}$
302b
Alice: D_1^2
Bob: D_2^2
Charlie: D_3^2
...
CT images:
$D^2 \in \mathbb{R}^{n \times d_2}$
302c
Alice: D_1^3
Bob: D_2^3
Charlie: D_3^3
...
The degree of liver cancer:
$D^3 \in \mathbb{R}^{n \times d_3}$

FIG. 3

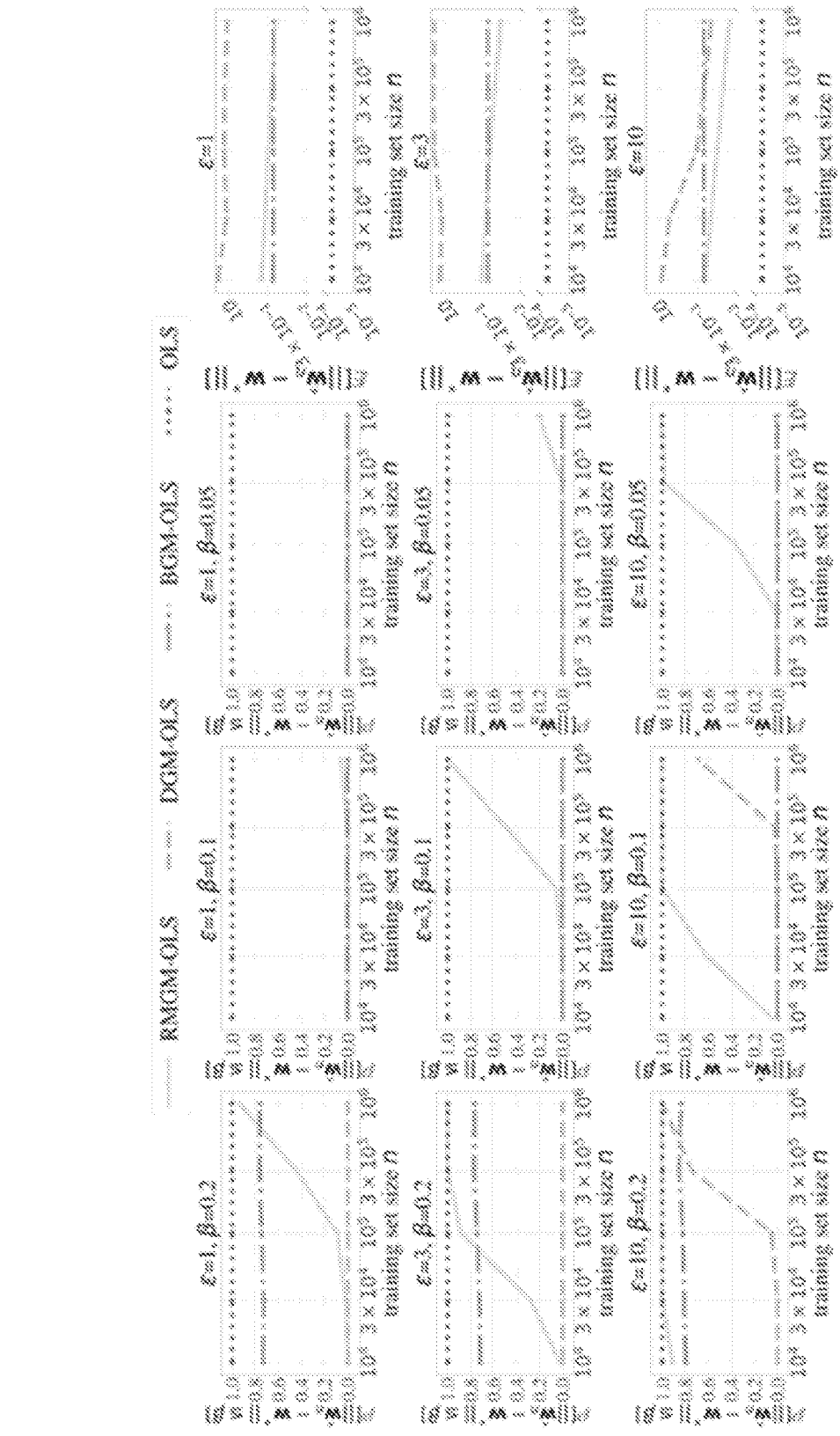
400
RMGM-OLS
DGM-OLS
BGM-OLS
OLS
ε=1, β=0.2
ε=1, β=0.1
ε=1, β=0.05
ε=1
ε=3, β=0.2
ε=3, β=0.1
ε=3, β=0.05
ε=3
ε=10, β=0.2
ε=10, β=0.1
ε=10, β=0.05
ε=10
training set size n
$\|w - w_0\|_2$

FIG. 4

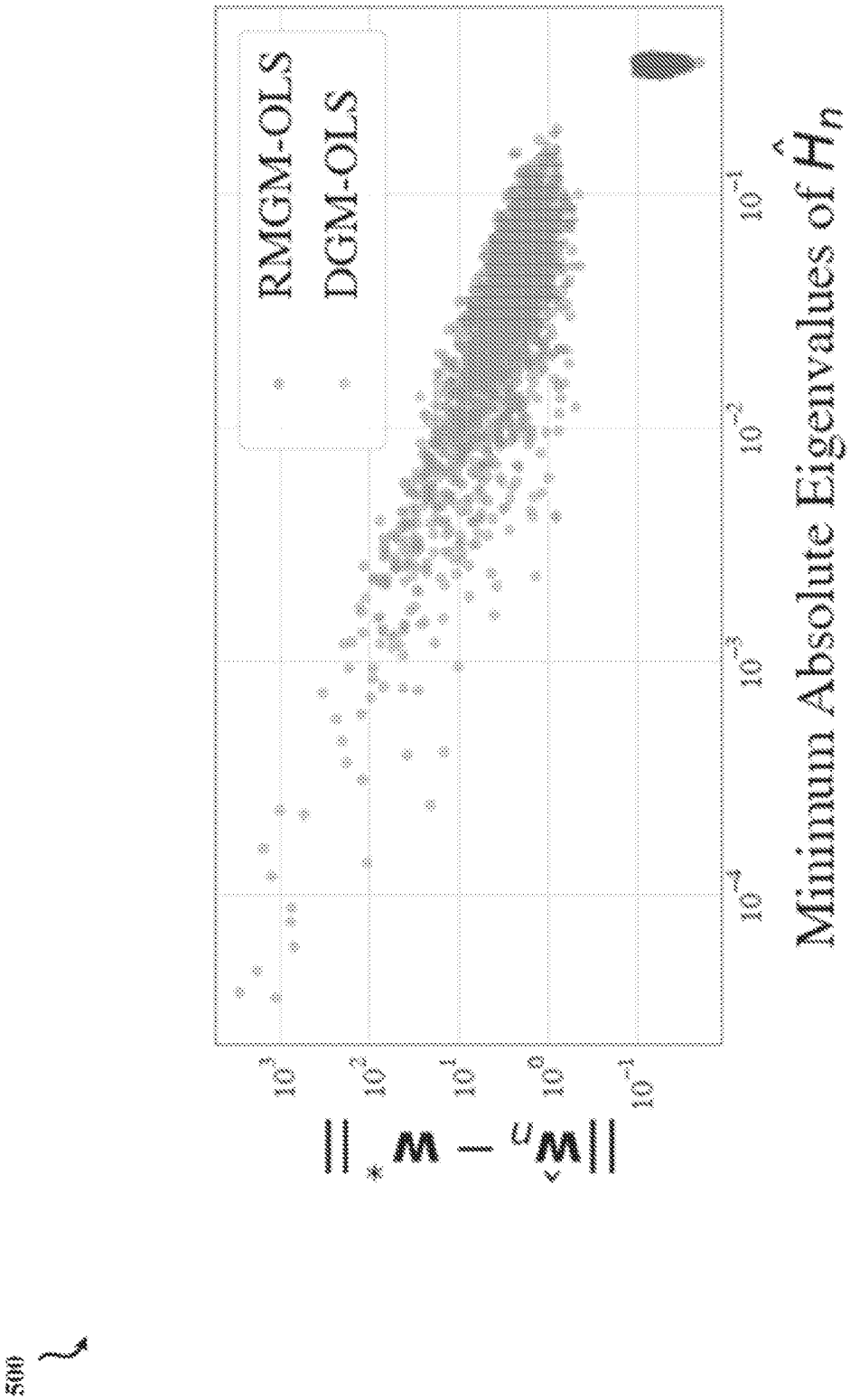
500
RMGM-OLS
DGM-OLS
$\|\hat{\mathbf{w}}_n - \mathbf{w}^*\|$
10^{3}
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
Minimum Absolute Eigenvalues of $\hat{H}_n$

FIG. 5

600

Dataset	Stat.	OLS	$\varepsilon = 1$			$\varepsilon = 3$			$\varepsilon = 10$		
			BGM	DGM	RMGM	BGM	DGM	RMGM	BGM	DGM	RMGM
Bike	$n = 13903$, $d = 13$	0.017	0.072	0.657	**0.060**	0.066	0.677	**0.061**	0.059	0.508	**0.029**
Superconductor	$n = 17010$, $d = 81$	0.009	**0.067**	1.240	0.262	**0.060**	0.708	0.069	0.061	0.728	**0.037**
GPU	$n = 193280$, $d = 14$	0.007	**0.016**	0.648	0.023	0.016	0.659	**0.012**	0.014	0.473	**0.009**
Music Song	$n = 412276$, $d = 90$	0.011	0.751	1.065	**0.148**	0.728	1.053	**0.030**	0.487	0.285	**0.016**

FIG. 6

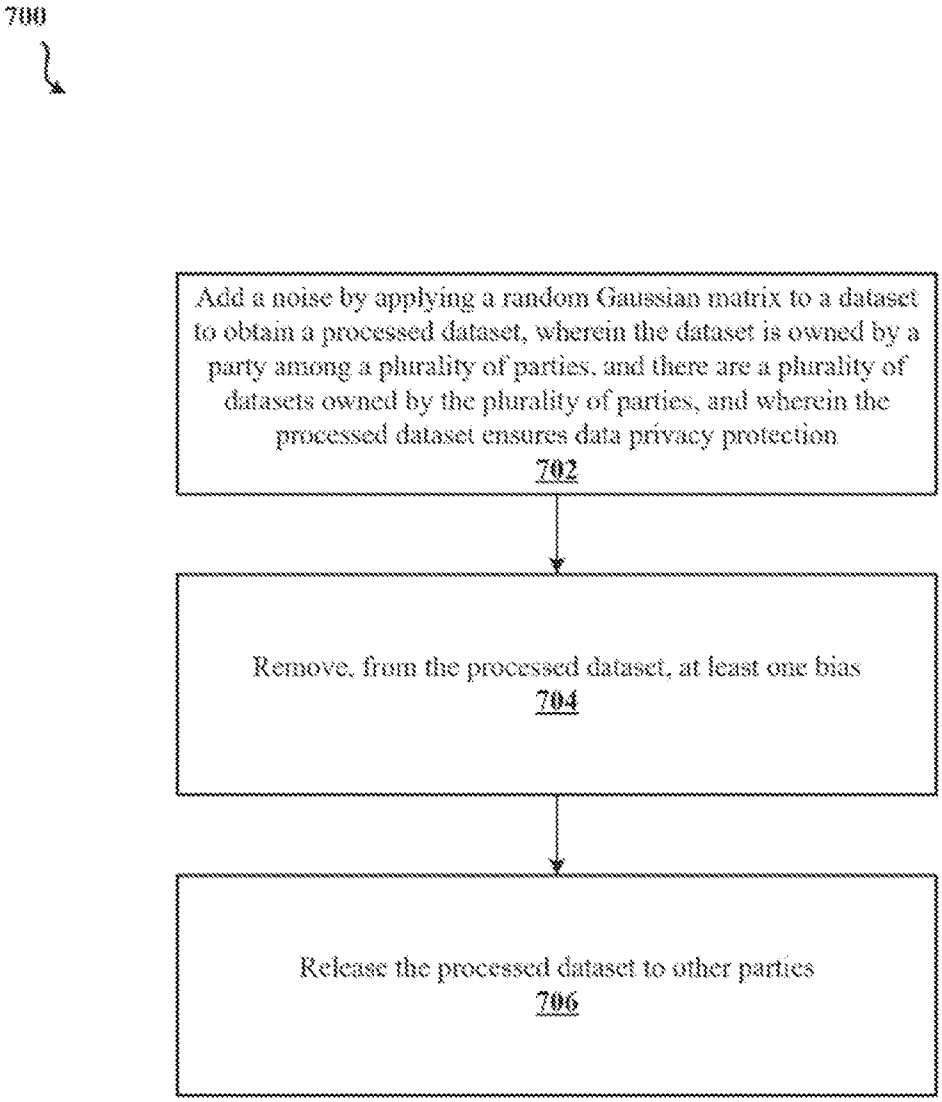
700
Add a noise by applying a random Gaussian matrix to a dataset to obtain a processed dataset, wherein the dataset is owned by a party among a plurality of parties, and there are a plurality of datasets owned by the plurality of parties, and wherein the processed dataset ensures data privacy protection
702
Remove, from the processed dataset, at least one bias
704
Release the processed dataset to other parties
706

FIG. 7

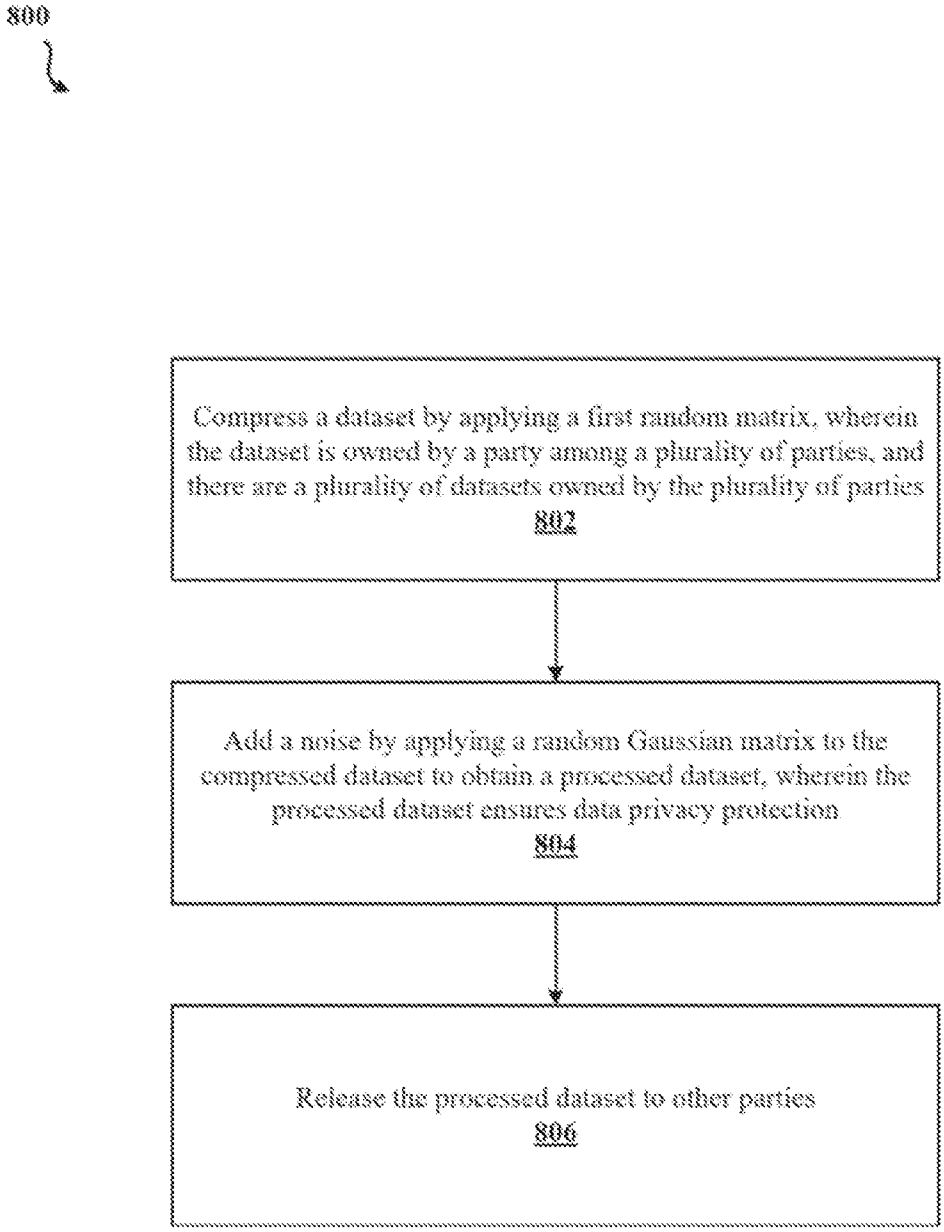
800
Compress a dataset by applying a first random matrix, wherein the dataset is owned by a party among a plurality of parties, and there are a plurality of datasets owned by the plurality of parties
802
Add a noise by applying a random Gaussian matrix to the compressed dataset to obtain a processed dataset, wherein the processed dataset ensures data privacy protection
804
Release the processed dataset to other parties
806

FIG. 8

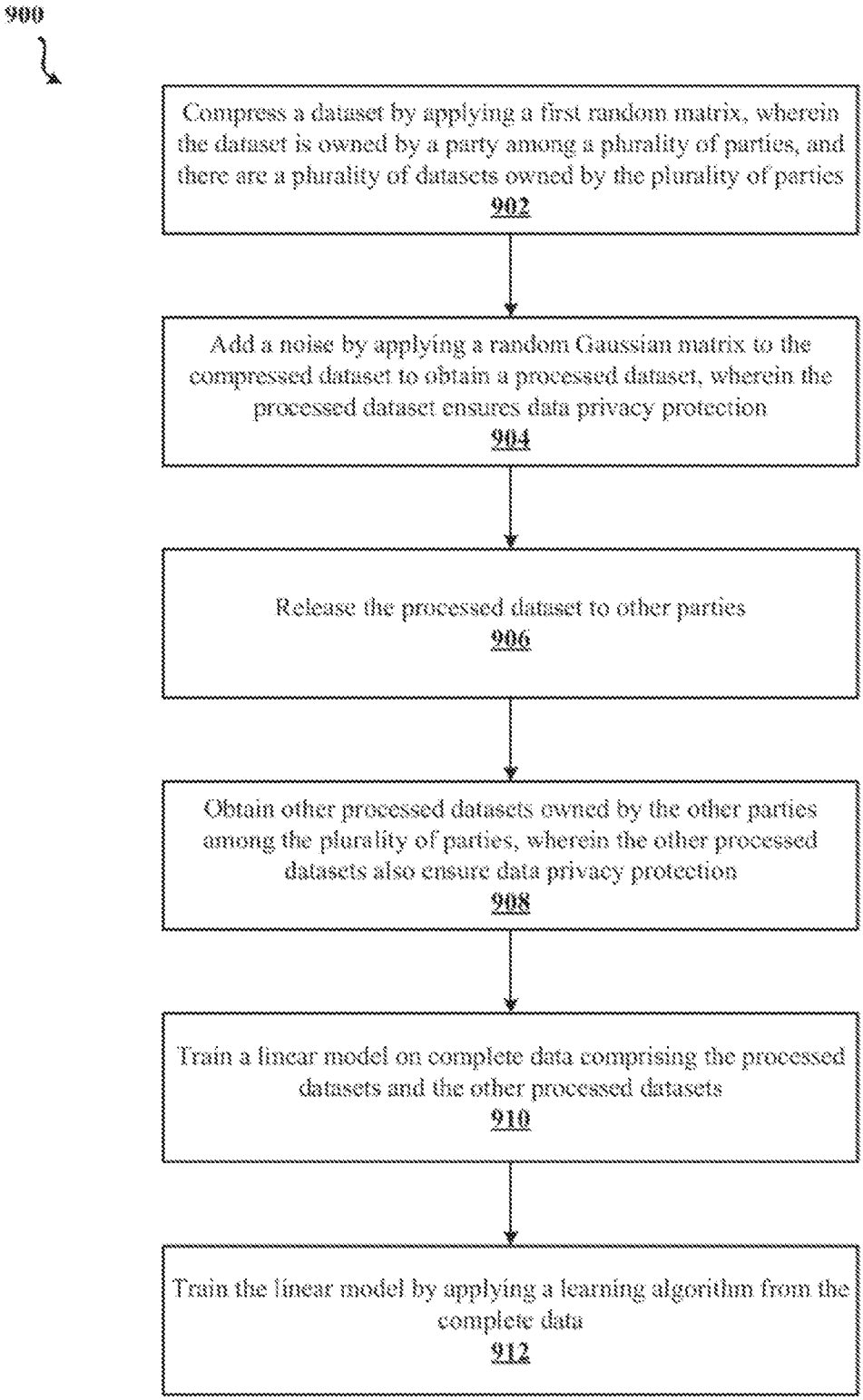
900
Compress a dataset by applying a first random matrix, wherein the dataset is owned by a party among a plurality of parties, and there are a plurality of datasets owned by the plurality of parties
902
Add a noise by applying a random Gaussian matrix to the compressed dataset to obtain a processed dataset, wherein the processed dataset ensures data privacy protection
904
Release the processed dataset to other parties
906
Obtain other processed datasets owned by the other parties among the plurality of parties, wherein the other processed datasets also ensure data privacy protection
908
Train a linear model on complete data comprising the processed datasets and the other processed datasets
910
Train the linear model by applying a learning algorithm from the complete data
912

FIG. 9

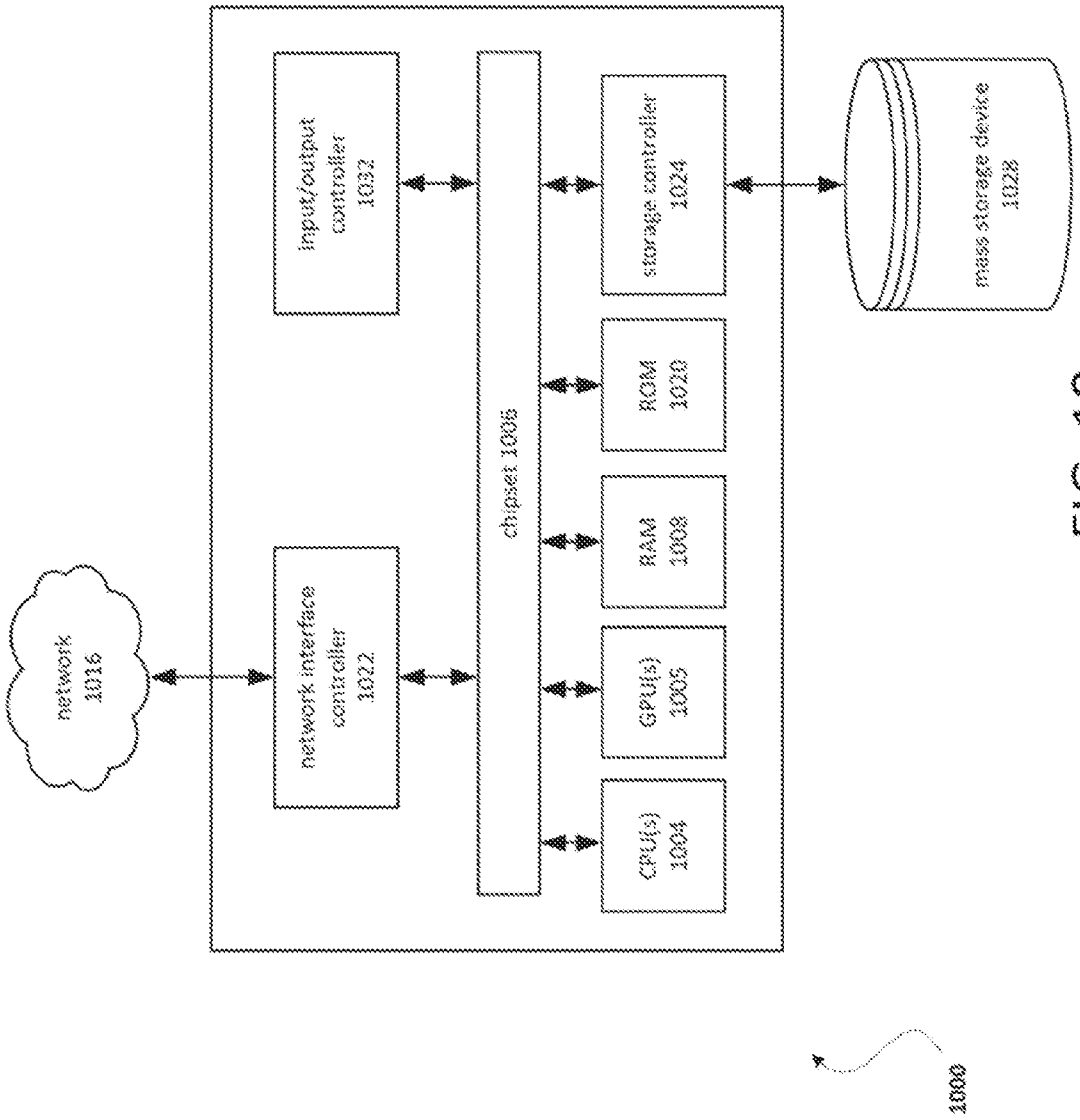
network
1016
network interface controller
1022
chipset 1006
input/output controller
1032
CPU(s)
1004
GPU(s)
1005
RAM
1008
ROM
1020
storage controller
1024
mass storage device
1028
1000

FIG. 10