

# MULTI-PARTY JOINT TRAINING METHOD AND APPARATUS FOR DATA PRIVACY PROTECTION

## TECHNICAL FIELD

[0001] One or more embodiments of this specification relate to the field of machine learning technologies, and in particular, to a multi-party joint training method and apparatus for data privacy protection, a computer-readable storage medium, and a computing device.

## BACKGROUND

[0002] With the development of artificial intelligence (Artificial Intelligence, AI for short) technologies, deep neural networks (Deep Neural Networks, DNNs for short) have been gradually applied to a plurality fields such as risk assessment, voice recognition, facial recognition, and natural language processing. DNN network structures in different application scenarios are relatively fixed. To implement better model performance, more training data are needed.

[0003] In many fields such as the medical and financial fields, different enterprises or institutions have different data samples. Once the data are used for joint training, model accuracy is greatly improved, thereby effectively improving service quality and improving user satisfaction. However, the original training data includes a large quantity of user privacy and a large quantity of business secrets. Once information is leaked, irreparable negative impact is caused. As such, when a data island problem is resolved through multi-party joint training, enhancement of data privacy protection and fully mining of data values have become a focus of research in recent years.

[0004] It is assumed that training members participating in joint training are in different industries and have a same user group. For example, for a payment platform and a social platform, the payment platform mostly holds payment data and the social platform mostly holds entertainment social data, and data collected by the payment platform and that collected by the social platform correspond to different feature spaces and a same sample space. This may be referred to as a data vertical division scenario.

[0005] However, currently, a joint training manner applied to the data vertical division scenario

is limited, and a higher requirement in actual application cannot be satisfied. Therefore, a joint training solution is needed to better satisfy an actual application requirement. For example, a non-multi-party common data set is introduced for training, to fully improve joint training effect.

## SUMMARY

5   **[0006]**     Embodiments of this specification describe a multi-party joint training method and apparatus for data privacy protection, to resolve a problem in which a non-multi-party common data set (or referred to as a private data set) of each party cannot be fully used in vertical split learning. Training effects of a model can be greatly improved by introducing more abundant training data.

10   **[0007]**     According to a first aspect, a multi-party joint training method for data privacy protection is provided. Multiple parties hold different feature parts of a sample in a common sample set. A label party in the multiple parties further holds a sample label. The method is applied to any target party of a non-label party in the multiple parties and includes:

                  determining first training losses of two locally deployed embedding models through  
15   contrastive learning based on an original sample total set and a corresponding augmented sample set, where the original sample total set includes a private sample set and the common sample set; using each sample in the original sample total set as a target sample, calculating stability indicators of the two embedding models for the target sample, using, as a label, an embedding vector that is output by an embedding model with higher stability for the target sample, and determining a  
20   second training loss of the other embedding model for the target sample; and providing, for the label party, a first embedding vector that is output by a predetermined first embedding model in the two embedding models for each common sample, so that the label party determines a third training loss based on a first embedding vector provided by each of the multiple parties, a locally deployed fusion model, and a held sample label, where the first training loss, the second training  
25   loss, and the third training loss are used for training a model locally deployed on each of the multiple parties.

**[0008]**     In an embodiment, the method further includes: performing data augmentation processing on the original sample total set to obtain the augmented sample set.

**[0009]**     In an embodiment, the stability indicator includes a representation information quantity,  
30   and calculation of the representation information quantity includes: for each embedding model in the two embedding models, determining two representation information quantities corresponding

to two embedding vectors respectively obtained after the target sample and a corresponding augmented sample are processed by using the embedding model, and determining a representation information quantity of the embedding model as a greater value in the two representation information quantities.

5    **[0010]**    In a specific embodiment, the determining two representation information quantities includes: for each embedding vector in the two embedding vectors, using a maximum value element in the embedding vector as a representation information quantity of the embedding vector.

10   **[0011]**    In an embodiment, the stability indicator includes a dispersion, and calculation of the dispersion includes: for each embedding model in the two embedding models, determining, as a dispersion of the embedding model, a difference between two embedding vectors respectively obtained after the target sample and a corresponding augmented sample are processed by using the embedding model.

15   **[0012]**    In an embodiment, determining the difference includes: calculating, as the difference, a quadratic sum of vector elements in a difference vector between the two embedding vectors, or a level 1 norm of the difference vector, or a level 2 norm of the difference vector.

20   **[0013]**    In an embodiment, the using, as a label, an embedding vector that is output by an embedding model with higher stability for the target sample, and determining a second training loss of the other embedding model for the target sample includes: when it is determined that at least one of the two embedding models reaches a stability standard, based on a calculation result of the stability indicator, using, as the label, the embedding vector that is output by the embedding model with higher stability for the target sample, and determining the second training loss of the other embedding model for the target sample.

25   **[0014]**    In a specific embodiment, the method further includes: discarding the target sample when it is determined that neither of the two embedding models satisfies the stability standard.

30   **[0015]**    Further, in an example, the stability indicator includes a representation information quantity and a dispersion, and the stability standard includes that the representation information quantity is greater than a first threshold and the dispersion is less than a second threshold.

30   **[0016]**    In an embodiment, after the providing, for the label party, a first embedding vector that is output by a predetermined first embedding model in the two embedding models for each common sample, the method further includes: receiving a target gradient from the label party, where the target gradient is a gradient that is determined based on the third training loss and that

is backhauled to the first embedding model deployed by the target party; and training the first embedding model by using the target gradient and a first training loss and a second training loss of the first embedding model.

**[0017]** In an embodiment, the two embedding models include the first embedding model and a second embedding model, and the method further includes: training the second embedding model by using a first training loss and a second training loss of the second embedding model.

**[0018]** According to a second aspect, a multi-party joint training method for data privacy protection is provided. Multiple parties hold different feature parts of a sample in a common sample set. A label party in the multiple parties further holds a sample label. The method is applied to the label party and includes:

determining first training losses of two locally deployed embedding models through contrastive learning based on an original sample total set and a corresponding augmented sample set, where the original sample total set includes a private sample set and the common sample set; using each sample in the original sample total set as a target sample, calculating stability indicators of the two embedding models for the target sample, using, as a label, an embedding vector that is output by an embedding model with higher stability for the target sample, and determining a second training loss of the other embedding model for the target sample; and determining a third training loss based on a first embedding vector that is output by a predetermined first embedding model in the two embedding models for each common sample, a first embedding vector provided by each of other parties in the multiple parties, a locally deployed fusion model, and a held sample label, where the first training loss, the second training loss, and the third training loss are used for training a model locally deployed on each of the multiple parties.

**[0019]** In an embodiment, based on a first embedding vector that is output by a predetermined first embedding model in the two embedding models for each common sample, a first embedding vector provided by each of other parties in the multiple parties, a locally deployed fusion model, and a held sample label includes: jointly inputting, into the fusion model, the locally output first embedding vector and the first embedding vector provided by each of other parties in the multiple parties, to obtain a predicted result of each common sample; and determining the third training loss based on the predicted result and a corresponding sample label.

**[0020]** In an embodiment, after the determining the third training loss, the method further includes: training the fusion model by using the third training loss; and calculating, based on the third training loss, a gradient that is backhauled to a first embedding model deployed on each of

the multiple parties, so that each party trains a locally deployed first embedding model based on a corresponding gradient, and a first training loss and a second training loss of the locally deployed first embedding model.

**[0021]** In an embodiment, the two embedding models include the first embedding model and a second embedding model, and the method further includes: training the second embedding model by using a first training loss and a second training loss of the second embedding model.

**[0022]** According to a third aspect, a multi-party joint training method for data privacy protection is provided. Multiple parties hold different feature parts of a sample in a common sample set. A label party in the multiple parties further holds a sample label. The method includes:

determining, by each of multiple parties, first training losses of two local embedding models through contrastive learning based on a local original sample total set and a corresponding augmented sample set, where the original sample total set includes a private sample set and the common sample set; using, by each of the multiple parties, each sample in the original sample total set as a target sample, calculating stability indicators of the two embedding models for the target sample, using, as a label, an embedding vector that is output by an embedding model with higher stability for the target sample, and determining a second training loss of the other embedding model for the target sample; and determining, by a label party, a third training loss based on a first embedding vector that is output by a predetermined first embedding model in the two local embedding models for each common sample, a first embedding vector provided by each of other parties in the multiple parties, a locally deployed fusion model, and a held sample label, where the first training loss, the second training loss, and the third training loss are used for training a model locally deployed on each of the multiple parties.

**[0023]** According to a fourth aspect, a multi-party joint training apparatus for data privacy protection is provided. Multiple parties hold different feature parts of a sample in a common sample set. A label party in the multiple parties further holds a sample label. The apparatus is integrated into any target party of a non-label party in the multiple parties and includes:

a contrastive learning module, configured to determine first training losses of two locally deployed embedding models through contrastive learning based on an original sample total set and a corresponding augmented sample set, where the original sample total set includes a private sample set and the common sample set; a distillation learning module, configured to use each sample in the original sample total set as a target sample, calculate stability indicators of the two embedding models for the target sample, use, as a label, an embedding vector that is output by an embedding model with higher stability for the target sample, and determine a second training

loss of the other embedding model for the target sample; and an embedding vector sending module, configured to provide, for the label party, a first embedding vector that is output by a predetermined first embedding model in the two embedding models for each common sample, so that the label party determines a third training loss based on a first embedding vector provided by each of the multiple parties, a locally deployed fusion model, and a held sample label, where the first training loss, the second training loss, and the third training loss are used for training a model locally deployed on each of the multiple parties.

**[0024]** According to a fifth aspect, a multi-party joint training apparatus for data privacy protection is provided. Multiple parties hold different feature parts of a sample in a common sample set. A label party in the multiple parties further holds a sample label. The apparatus is integrated into the label party and includes:

a contrastive learning module, configured to determine first training losses of two locally deployed embedding models through contrastive learning based on an original sample total set and a corresponding augmented sample set, where the original sample total set includes a private sample set and the common sample set; a distillation learning module, configured to use each sample in the original sample total set as a target sample, calculate stability indicators of the two embedding models for the target sample, use, as a label, an embedding vector that is output by an embedding model with higher stability for the target sample, and determine a second training loss of the other embedding model for the target sample; and a supervised learning module, configured to determine a third training loss based on a first embedding vector that is output by a predetermined first embedding model in the two embedding models for each common sample, a first embedding vector provided by each of other parties in the multiple parties, a locally deployed fusion model, and a held sample label, where the first training loss, the second training loss, and the third training loss are used for training a model locally deployed on each of the multiple parties.

**[0025]** According to a sixth aspect, a computer-readable storage medium is provided. The computer-readable storage medium stores a computer program. When the computer program is executed on a computer, the computer is enabled to perform the method in the first aspect, the second aspect, or the third aspect.

**[0026]** According to a seventh aspect, a computing device is provided, including a memory and a processor. The memory stores executable code. When executing the executable code, the processor implements the method in the first aspect, the second aspect, or the third aspect.

**[0027]** In the above-mentioned method and apparatus provided in embodiments of this specification, contrastive learning and knowledge distillation are used. In addition to sample data

common to a plurality of participants, private sample data of each party is also introduced for model training, to improve and optimize prediction performance of a model obtained through joint training. In addition, different from a teacher model and a student model pre-specified in a conventional manner, distillation learning is used to determine a learning direction sample by sample based on two embedding models in a data party for stability of a same sample, so that an unstable model learns an embedding representation of a sample from a stable model, thereby improving learning effect of the model.

## **BRIEF DESCRIPTION OF DRAWINGS**

**[0028]** To describe the technical solutions in embodiments of the present invention more clearly, the following briefly describes the accompanying drawings needed for describing embodiments. Clearly, the accompanying drawings in the following descriptions show merely some embodiments of the present invention, and a person of ordinary skill in the art can still derive other accompanying drawings from these accompanying drawings without creative efforts.

**[0029]** FIG. 1 is schematic distribution diagram of a data vertical division scenario according to an embodiment of this specification;

**[0030]** FIG. 2 is a schematic diagram of an implementation architecture of a multi-party joint training solution for data privacy protection according to an embodiment of this specification;

**[0031]** FIG. 3 is a schematic flowchart of steps when a multi-party joint training method for data privacy protection is applied to any non-label party according to an embodiment of this specification;

**[0032]** FIG. 4 is a schematic flowchart of steps when a multi-party joint training method is applied to a label party according to an embodiment of this specification;

**[0033]** FIG. 5 is a schematic structural diagram of an apparatus when a multi-party joint training apparatus for data privacy protection is integrated into any non-label party according to an embodiment of this specification; and

**[0034]** FIG. 6 is a schematic structural diagram of an apparatus when a multi-party joint training apparatus for data privacy protection is integrated into a label party according to an embodiment of this specification.

## DESCRIPTION OF EMBODIMENTS

[0035] The following describes solutions provided in this specification with reference to the accompanying drawings.

[0036] Embodiments of this specification disclose a multi-party joint training solution for data privacy protection. In this solution, contrastive learning (Contrastive Learning) and knowledge distillation (Knowledge Distillation) are used. In addition to sample data common to a plurality of participants, private sample data of each party is also introduced for model training, to improve and optimize prediction performance of a model obtained through joint training. It should be understood that a participant is referred to as a data party, a computing party, or a party for short in this specification.

[0037] The following first describes data distribution in vertical split learning. In most vertical division scenarios, samples of participants are often not strictly aligned. To be specific, after multiple parties perform data alignment, a part of data is still unique or private to each training member. For example, data alignment may be implemented through executing a private set intersection (Private Set Intersection, PSI) protocol.

[0038] In this specification, a same sample obtained by the multiple parties through data alignment is referred to as a common sample, and a sample other than the common sample held by each of the multiple parties is referred to as a private sample (or a unique sample). It may be understood that the multiple parties hold different feature parts of the common sample, and a plurality of common samples form a common sample set. In addition, in the multiple parties, a party further holding a sample label is referred to as a label party, and another party is referred to as a non-label party.

[0039] FIG. 1 is schematic distribution diagram of a data vertical division scenario according to an embodiment of this specification. A data party A and a data party B in the multiple parties are shown. The data party A is a label party, and the data party B is a non-label party. Specifically, the data party A holds a feature X1 and a label Y1 of a common sample of the two parties, and the data party B holds a feature Z1 of the common sample. In addition, the data party A further holds a feature X2 of a private sample of the data party A, and the data party further holds a feature Z2 of a private sample of the data party. It should be understood that the data party A further holds a label Y2 of the private sample of the data party A. The label Y2 is not shown in FIG. 1 because the

label Y2 is not used in the proposed joint training solution.

**[0040]** The following describes, based on the two data parties shown in FIG. 1, different feature parts held by the parties in different service scenarios and a sample label held by the label party.

5 **[0041]** (1) Recommended scenario

**[0042]** In the recommended scenario, the data party A holds object feature data and a user behavior label of a recommended sample, and the data party B holds user feature data of the recommended sample. It is worthwhile to note that user-related data such as the user feature data mentioned in this specification is collected and used after acknowledgment and authorization of  
10 the user.

**[0043]** In an embodiment, the data party A may be an e-commerce platform, a payment platform, a content platform, or the like. A service object of the object feature data owned by the data party A may be a commodity, a service, an official account, a subscription account, an application, content information, an advertisement, or the like.

15 **[0044]** In an example, the object feature data is a commodity feature, for example, a category, a place of origin, costs, a price, a sales volume, and a target audience. In another example, the object feature data is an official account feature, for example, a category, a number of followers, a content publishing frequency, a number of likes, or a number of reposts. In still another example, the object feature data is an advertisement feature, for example, an advertisement carrier (including  
20 a picture, a text, a video, or audio), advertisement content, and spokesperson information.

**[0045]** In an embodiment, the user feature data includes a static attribute feature of the user, for example, a career, a hobby, a residence, and asset information. For example, these features may be extracted from user registration information. In another embodiment, the user data further includes a dynamic behavior feature of a user. In an example, the data party B is a social platform.

25 In this case, the dynamic behavior feature may include a social duration, a social frequency, a number of friends, a number of updates, a number of likes, and the like. In another example, the data party B is a payment platform. In this case, the dynamic behavior feature may include a payment frequency, a payment amount, a payment time period, and the like.

**[0046]** The above-mentioned user behavior label indicates whether the user performs a  
30 predetermined behavior on the service object. It may be understood that the predetermined behavior corresponds to the service object. In an embodiment, the service object is a commodity

or a service. In this case, the predetermined behavior may be purchase or use. In another embodiment, the service object is an official account or a subscription account. In this case, the predetermined behavior may be follow or subscribe. In still another embodiment, the service object is an application. In this case, the predetermined behavior may be download or register. In yet another embodiment, the service object is an advertisement or content information. In this case, the predetermined behavior may be click.

**[0047]** Multi-party data distribution in the recommended scenario is described above.

**[0048]** (2) Risk control scenario

**[0049]** In the risk control scenario, multiple parties may respectively hold different pieces of behavior data of a same user. In an embodiment, the data party A is a bank, and the data party B is a payment platform. The bank holds an account statement of a user, such as deposit and withdrawal records. The payment platform holds payment data of the user, such as a payment amount and a quantity of payment times. In another embodiment, the data party A is a credit service platform, and the data party B is a credit bureau. The credit service platform holds service data generated when the user uses a credit service, such as a record of renting books or bicycles with no deposit required. The credit bureau holds credit data of the user, such as a quantity of credit cards held by the user and a payment default record.

**[0050]** The label party in the multiple parties holds a risk label that may be specifically a risk category label, such as a low risk, a medium risk, and a high risk; or may be a risk score label, such as 30 points and 90 points in a hundred-mark system.

**[0051]** Vertical data distribution of multiple parties in different service scenarios is described above with reference the recommended scenario and the risk control scenario by using two parties in the multiple parties as an example. It may be understood that the multiple parties are not limited to two parties, or may be any other quantity of participants that is greater than 2. A service scenario is not limited to the recommended scenario and the risk control scenario, or may be alternatively an image classification scenario, a text classification scenario, an audio classification scenario, or the like. Examples are not exhaustively listed herein.

**[0052]** Next, an implementation architecture of the multi-party joint training solution disclosed in embodiments of this specification is described still based on the two data parties shown in FIG.

1. FIG. 2 is a schematic diagram of an implementation architecture of a multi-party joint training solution for data privacy protection according to an embodiment of this specification. As shown

in the figure, two embedding models: EA1 and EA2 and a fusion model are locally deployed on a data party A, and two embedding models: EB1 and EB2 are locally deployed on a data party B.

**[0053]** As shown in FIG. 2, the joint training solution is related to three types of learning tasks, including contrastive learning, distillation learning (or referred to as knowledge distillation), and supervised learning. Specifically, models for contrastive learning and distillation learning include four embedding models, namely, the embedding models: EA1, EA2, EB1, and EB2. Full sample data of each party (including a common sample and a private sample) is used for contrastive learning and distillation learning, thereby improving stability of an embedding model. In addition, different from a teacher model and a student model pre-specified in a conventional manner, distillation learning is used to determine a learning direction in a sample dimension based on two embedding models in a single data party for stability of a same sample. Specifically, an unstable model learns from a stable model, thereby improving learning effect of the model.

**[0054]** Models for supervised learning include three models, namely, the embedding models: EA1 and EB1 and the fusion model. The fusion model is used to perform fusion and prediction processing on outputs of the embedding models: EA1 and EB1 for a common sample to obtain a prediction result, thereby implementing supervised learning based on the prediction result and the sample label.

**[0055]** Therefore, multiple parties each can obtain two trained embedding models, and the label party further obtains a trained fusion model. When the trained model is put into actual joint use, each party may select one from the two embedding models for use, and usually, select an embedding model additionally for supervised learning.

**[0056]** With reference to FIG. 3 to FIG. 5 and more embodiments, the following describes specific implementation steps of the above-mentioned joint training solution.

**[0057]** FIG. 3 shows steps in a method performed by any non-label party in multiple parties that participate in a joint training process. It should be understood that, for brief description, any non-label party is referred to as a target party in this specification. For example, the target party may be the data party B shown in FIG. 1 or FIG. 2. In addition, each of the multiple parties may be implemented as any apparatus, platform, server, device cluster, or the like having computing and processing capabilities. The steps shown in FIG. 3 include the following steps:

**[0058]** Step S310: Determine first training losses of two locally deployed embedding models through contrastive learning based on an original sample total set and a corresponding augmented

sample set, where the original sample total set includes a private sample set and the common sample set. Step S320: Use each sample in the original sample total set as a target sample, calculate stability indicators of the two embedding models for the target sample, use, as a label, an embedding vector that is output by an embedding model with higher stability for the target sample, and determine a second training loss of the other embedding model for the target sample. Step S330: Provide, for the label party, a first embedding vector that is output by a predetermined first embedding model in the two embedding models for each common sample, so that the label party determines a third training loss based on a first embedding vector provided by each of the multiple parties, a locally deployed fusion model, and a held sample label, where the first training loss, the second training loss, and the third training loss are used for training a model locally deployed on each of the multiple parties.

**[0059]** The above-mentioned steps are elaborated as follows:

**[0060]** In step S310, the first training losses of the two locally deployed embedding models are determined through contrastive learning based on the original sample total set and the corresponding augmented sample set. First, it is worthwhile to note that similar terms such as "first" in the first training loss and "first", "second", or "third" elsewhere in this specification are used to distinguish between similar things, and do not have another limitation function such as sorting.

**[0061]** The original sample total set includes the private sample set and the common sample set. The common sample set is a set formed by a plurality of samples that are shared by the multiple parties. The private sample set is a set formed by samples that are held by the target party in addition to the common sample set. For descriptions of a common sample and a private sample, reference may be made to related descriptions in the above-mentioned embodiments.

**[0062]** Through performing data augmentation processing on an original local feature of each sample in the original sample total set, a corresponding augmented data set (or referred to as the augmented sample set) may be obtained. In an embodiment, data augmentation processing may include adding noise disturbance, masking (mask) some features, and the like. In another embodiment, for each sample in the original sample total set, data augmentation processing may be performed for several times on the original local feature of the sample, to obtain several corresponding augmented local features to be classified into the augmented data set.

**[0063]** According to an example, the original sample total set is an original user total set, and

the target party holds an original face image of each user in the original user total set. In this case, for each original face image, a means may be selected from random clipping, random color distortion, and random Gaussian blur as a data augmentation means, to obtain a corresponding augmented face image to form an augmented face image set.

5   **[0064]**     According to another example, the target party holds an original feature vector of each user in an original user total set. In this case, for each original feature vector, a manner of randomly masking (mask) some features may be used. For example, a predetermined quantity of vector elements is randomly set to zero, a null value, or default, to obtain a corresponding augmented feature vector to form an augmented feature vector set.

10   **[0065]**     Based on the original sample total set and the augmented sample set, contrastive learning is separately performed for the two locally deployed embedding models, to determine respective first training losses. The first training loss may be referred to as a contrastive loss. In this way, through introducing a private sample of each party other than a common sample into joint training, stability of the two embedding models can be effectively improved, thereby  
15   improving overall training effect of all the models in joint training.

**[0066]**     It is worthwhile to note that, usually, the two embedding models are configured to have a same input dimension and a same output dimension. Other structural parameters such as a quantity of neural network layers and a quantity of neurons, or neural network algorithms on which the two embedding models are based may be configured to be the same or different. In addition,  
20   even if structures of the two embedding models are completely consistent, parameters are initialized to be consistent. However, because training losses on which update of the parameters of the two models are based are different, updated parameters are different.

**[0067]**     For ease of understanding, any one of the two embedding models is referred to as a target embedding model below, and a process of determining a first training loss of the target  
25   embedding model through contrastive learning performed on the target embedding model is described below.

**[0068]**     (1) Embedding processing The target embedding model is used to separately perform embedding processing on  $2N$  samples to correspondingly obtain  $2N$  sample embedding vectors. The  $2N$  samples include a batch of  $N$  original samples sampled from the original sample total set,  
30   and  $N$  augmented samples corresponding to the  $N$  original samples in the augmented sample set.

**[0069]**     (2) Calculation of the first training loss  $N$  positive sample pairs may be constructed by

using the N original samples and the N augmented samples. Each sample in each positive sample pair and each sample in the other (N-1) positive sample pairs form a negative sample pair. The first training loss is determined through reducing a positive sample pair distance and increasing a negative sample pair distance. For example, the following formula is used:

$$L_{con}(x_j, x_k) = -\log \frac{\exp(\text{sim}(\mathbf{z}_{x_j}, \mathbf{z}_{x_k})/\tau)}{\sum_{l=1}^{2N} \mathbf{1}_{[l \neq j]} \exp(\text{sim}(\mathbf{z}_{x_k}, \mathbf{z}_{x_l})/\tau)} \quad (1)$$

**[0070]** In Formula (1), samples  $x_j$  and  $x_k$  form a positive sample pair,  $\mathbf{z}_{x_j}$  and  $\mathbf{z}_{x_k}$  respectively represent embedding vectors output by the target embedding model for the samples  $x_j$  and  $x_k$ ,  $\text{sim}(\mathbf{z}_{x_j}, \mathbf{z}_{x_k})$  represents a similarity (such as a cosine similarity) between the embedding vectors  $\mathbf{z}_{x_j}$  and  $\mathbf{z}_{x_k}$ , and  $\tau$  represents a temperature coefficient. For example,  $\tau = 0.5$  is manually set.

**[0071]** An example of contrastive learning for an embedding model is described above. Actually, construction of positive and negative sample pairs and design of a contrastive loss calculation formula are not limited to the above-mentioned description.

**[0072]** The first training loss of each of the two locally deployed embedding models may be obtained by using the original sample total set and the corresponding augmented data set.

**[0073]** In step S320, each sample in the original sample total set is used as the target sample, the stability indicators of the two embedding models for the target sample are calculated, the embedding vector that is output by the embedding model with higher stability for the target sample is used as the label, and the second training loss (or referred to as a distillation loss) of the other embedding model for the target sample is determined.

**[0074]** It is worthwhile to note that in this step, distillation learning is refined to a sample granularity, and a learning direction between the two embedding models is dynamically determined. The learning direction may change with different samples. Specifically, based on stability of the two embedding models on a single sample, an unstable model learns from an embedding vector of a stable model. In this way, training effect of the two embedding models can be effectively improved, thereby improving overall training effect of all models in joint training.

**[0075]** It should be understood that stability of an embedding model on a sample may be measured by a predefined stability indicator. In an embodiment, the stability indicator includes a representation information quantity. In other words, stability of a model is measured from a

perspective of an information quantity included in an embedding representation obtained through processing a sample by using an embedding model. Apparently, a larger representation information quantity indicates better representation effect and higher model stability.

[0076] A calculation manner of the representation information quantity may be obtained through reasonable definition, and is not limited to a specific calculation manner. In this step, any embedding model in the two embedding models is also referred to as a target embedding model. In a specific embodiment, a maximum element value in a target embedding vector obtained after a target sample is processed by using the target embedding model may be determined as a representation information quantity of the target embedding model for the target sample. In another specific embodiment, a sum of absolute values of all elements in a target embedding vector may be determined as a representation information quantity of the target embedding model for a target sample. In another aspect, in a specific embodiment, the two representation information quantities corresponding to the two embedding vectors respectively obtained after the target sample and the corresponding augmented sample are processed by using the target embedding model may also be first determined, and then a representation information quantity of the target embedding model is determined as a greater value in the two representation information quantities.

[0077] According to an example, a representation information quantity  $M_x$  of the target embedding model for a target sample  $x$  may be calculated by using the following formula:

$$M_x = \max(\|z_x\|_\infty, \|z_{\tilde{x}}\|_\infty) \quad (2)$$

[0078] In the above-mentioned formula,  $x$  and  $\tilde{x}$  respectively represent the target sample and a corresponding augmented sample,  $z_x$  and  $z_{\tilde{x}}$  represent embedding vectors correspondingly obtained through separately processing the samples  $x$  and  $\tilde{x}$  by using the target embedding model, and an operator  $\|\cdot\|_\infty$  represents an infinite norm of a vector. An operation result is a maximum element value in the vector.

[0079] Therefore, the representation information quantity of the embedding model for the target sample can be calculated.

[0080] In another embodiment, the stability indicator includes a dispersion. Specifically, a difference between the two embedding vectors respectively obtained after the target sample and the corresponding augmented sample are processed by using the target embedding model is determined as the dispersion of the target embedding model. Apparently, higher dispersion indicates poorer model stability.

[0081] A calculation manner of the dispersion may be obtained through reasonable definition, and is not limited to a specific calculation manner. In a specific embodiment, a difference vector between the two embedding vectors is first calculated, and then a quadratic sum of vector elements in the difference vector is used as the dispersion of the target embedding model. For example, the dispersion  $\epsilon_x$  may be calculated by using the following formula:

$$\epsilon_x = \|\mathbf{z}_x - \mathbf{z}_{\tilde{x}}\|^2 \quad (3)$$

[0082] In the above-mentioned formula,  $x$  and  $\tilde{x}$  respectively represent the target sample and the corresponding augmented sample,  $\mathbf{z}_x$  and  $\mathbf{z}_{\tilde{x}}$  represent the embedding vectors correspondingly obtained through separately processing the samples  $x$  and  $\tilde{x}$  by using the target embedding model, an operator  $\|\cdot\|$  represents a 2 norm of a vector, and a superscript 2 represents square.

[0083] In another specific embodiment, a sum of absolute values of vector elements in the difference vector, or a level 1 norm of the difference vector, or a level 2 norm of the difference vector is used as the dispersion of the target model. In still another specific embodiment, a cosine distance or an Euclidean distance between the two vectors is calculated as the dispersion of the target model.

[0084] The stability indicator is described above. The representation information quantity and the dispersion are mainly used as examples. It should be understood that other indicators may be involved in practice. Examples are not exhaustively listed herein.

[0085] Based on a calculation result of the stability indicator, an embedding model with higher stability in the two embedding models may be determined. In an embodiment, the stability indicator includes a representation information quantity. In this case, an embedding model with a higher representation information quantity in the two embedding models may be determined as the embedding model with higher stability. In another embodiment, the stability indicator includes a dispersion. In this case, an embedding model with a lower dispersion in the two embedding models may be determined as the embedding model with higher stability. In still another embodiment, the stability indicator includes a representation information quantity and a dispersion. In this case, an embedding model with a larger difference obtained by subtracting a dispersion from a representation information quantity may be determined as the embedding model with higher stability.

[0086] After the embedding model with higher stability is determined, an embedding vector

output by the embedding model for the target sample may be used as a label, and a second training loss of the other embedding model for the target sample is determined, thereby implementing embedding representation for the unstable model learning from the stable model.

**[0087]** It may be understood that a purpose of constructing the second training loss is to reduce a distance between the embedding vector label and an embedding vector output by the other embedding model for the target sample. Calculation of the second training loss is performed based on the two embedding vectors of the two embedding models for the target sample. There are a plurality of calculation manners. For example, a distance, such as an Euclidean distance, a square of an Euclidean distance, or a cosine distance, between the two embedding vectors may be calculated as the second training loss.

$$L_{kd}(x) = Dis(\mathbf{z}_x - \mathbf{z}_{\tilde{x}}) \quad (4)$$

**[0088]** In Formula (4),  $L(x)$  represents the second training loss, and  $Dis(\mathbf{z}_x - \mathbf{z}_{\tilde{x}})$  represents the distance between the two embedding vectors  $\mathbf{z}_x$  and  $\mathbf{z}_{\tilde{x}}$ .

**[0089]** In another aspect, availability of the target sample is further considered. If stability of the two embedding models for the target sample does not satisfy a preset stability standard, the target sample is discarded, that is, the target sample does not participate in distillation learning. If stability of at least one of the two embedding models for the target sample satisfies a stability standard, embedding representation learning indicating that the unstable model learns from the stable model is performed by using the target sample.

**[0090]** In an embodiment, the stability indicator includes a representation information quantity. In this case, the stability standard includes that the representation information quantity is greater than a preset first threshold. In another embodiment, the stability indicator includes a dispersion. In this case, the stability standard includes that the dispersion is less than a preset second threshold. In still another embodiment, the stability indicator includes both a representation information quantity and a dispersion. In this case, the stability standard includes that the representation information quantity is greater than a first threshold and the dispersion is less than a second threshold. For example, the stability standard may be represented as follows:

$$R_x = \{M_x > \xi\}_1 \& \{\epsilon_x < \eta\}_1 \quad (5)$$

**[0091]**  $M_x$  represents the representation information quantity;  $\xi$  represents the first threshold;  $\{M_x > \xi\}_1$  represents that 1 is taken if  $M_x > \xi$  is satisfied, or otherwise 0 is taken;  $\epsilon_x$  represents the dispersion;  $\eta$  represents the second threshold;  $\{\epsilon_x < \eta\}_1$  represents that 1 is taken

if  $\epsilon_x < \eta$  is satisfied, or otherwise, 0 is taken; and  $\&$  represents a logical operator AND. It may be understood that, if  $R_x = 1$  is obtained through calculation, it indicates that the stability standard is satisfied; or if  $R_x = 0$  is obtained through calculation, it indicates that the stability standard is not satisfied.

5 **[0092]** The content of the stability standard and the manner of determining whether the standard is satisfied are described above.

**[0093]** According to an example, determining whether stability reaches a standard may be integrated into a calculation formula of the second training loss:

$$L_{kd}^{(1)} = \begin{cases} \left\{ \epsilon_x^{(1)} > \epsilon_x^{(2)} \right\}_1 Dis(\mathbf{z}_x - \mathbf{z}_{\tilde{x}}), & R_x^{(1)} = R_x^{(2)} = 1 \\ R_x^{(2)} Dis(\mathbf{z}_x - \mathbf{z}_{\tilde{x}}), & \text{Otherwise} \end{cases} \quad (6)$$

$$10 \quad L_{kd}^{(2)} = \begin{cases} \left\{ \epsilon_x^{(2)} > \epsilon_x^{(1)} \right\}_1 Dis(\mathbf{z}_x - \mathbf{z}_{\tilde{x}}), & R_x^{(1)} = R_x^{(2)} = 1 \\ R_x^{(1)} Dis(\mathbf{z}_x - \mathbf{z}_{\tilde{x}}), & \text{Otherwise} \end{cases} \quad (7)$$

**[0094]** In Formulas (6) and (7), a superscript (1) corresponds to one of the two locally deployed embedding models, and a superscript (2) corresponds to the other embedding model. For a meaning of another mathematical symbol, reference may be made to the related descriptions in the above-mentioned embodiments.

15 **[0095]** Therefore, the learning direction between the two embedding models may be dynamically determined based on the original sample total set and an idea that the unstable model learns the embedding representation from the stable model. Specifically, it is determined to calculate, by using a part of samples, a second training loss when one embedding model learns an embedding representation from the other embedding model, and calculate, by using another  
20 different part of samples, a second training loss of reverse learning of the other embedding model.

**[0096]** Then, in step S330, the first embedding vector that is output by the predetermined first embedding model in the two embedding models for each common sample is provided for the label party.

25 **[0097]** It should be noted that any embedding model may be specified from the two embedding models, and an embedding vector of the embedding model may be provided for the label party. Therefore, compared with the other embedding model, in a reverse propagation process, the specified first embedding model further additionally uses a third training loss generated in supervised learning to implement parameter adjustment. Usually, if the two embedding models

have different parameter orders of magnitude or different complexity, an embedding model with a larger parameter order of magnitude or a more complex structure is usually set as the first embedding model that provides an embedding vector for the label party. In addition, repeated specifying is not needed after the first embedding model is specified. In other words, the embedding model that provides the embedding vector for the label party is always the specified model in a plurality of rounds of iterative training.

**[0098]** After the target party provides, for the label party, the first embedding vector output by the first embedding model for each common sample, the label party determines the third training loss based on the first embedding vector provided by each of the multiple parties, the locally deployed fusion model, and the held sample label. It should be understood that, when the sample label is a category label, the third training loss may be referred to as a category loss. The above-mentioned first training loss, the second training loss, and the third training loss are used for training the locally deployed models of the multiple parties. It should be understood that outputs of some models do not participate in calculation of a training loss. In this case, a derivative or a gradient of the training loss to these model parameters is 0. In other words, for these models, a parameter adjustment result remains the same regardless of whether the training loss is used.

**[0099]** In addition, when a forward propagation process of joint training and the calculation manner of the training loss are determined, a reverse propagation process is determined accordingly. In consideration of this, with reference to FIG. 4 and more embodiments, the following first describes a forward propagation process of the label party other than the target party in joint training, and then briefly describes a reverse propagation process in joint training.

**[0100]** FIG. 4 is a schematic flowchart of steps when a multi-party joint training method is applied to a label party according to an embodiment of this specification. The steps shown in the figure include:

**[0101]** Step S410: Determine first training losses of two locally deployed embedding models through contrastive learning based on an original sample total set and a corresponding augmented sample set, where the original sample total set includes a private sample set and the common sample set. Step S420: Use each sample in the original sample total set as a target sample, calculate stability indicators of the two embedding models for the target sample, use, as a label, an embedding vector that is output by an embedding model with higher stability for the target sample, and determine a second training loss of the other embedding model for the target sample. Step

S430: Determine a third training loss based on a first embedding vector that is output by a predetermined first embedding model in the two embedding models for each common sample, a first embedding vector provided by each of other parties in the multiple parties, a locally deployed fusion model, and a held sample label, where the first training loss, the second training loss, and the third training loss are used for training a model locally deployed on each of the multiple parties.

[0102] The above-mentioned steps are elaborated as follows:

[0103] First, for step S410 and step S420, because the label party also holds a sample feature, and the label party is the same as a non-label party in terms of the sample feature, the label party performs steps S410 and S420, and processes a local sample by using two locally deployed embedding models, to determine a first training loss and a second training loss. For this process, reference may be made to descriptions of steps S310 and S320 performed by the non-label party in the above-mentioned embodiment. Details are not described again.

[0104] For step S430, the third training loss is determined based on the first embedding vector that is output by the predetermined first embedding model in the two embedding models for each common sample, the first embedding vector provided by each of the other parties in the multiple parties, the locally deployed fusion model, and the held sample label.

[0105] It should be noted that for selection of the first embedding model, reference may be made to the related descriptions in the above-mentioned embodiment.

[0106] In this step, the locally output first embedding vector and the first embedding vector provided by each of the other parties in the multiple parties may be first jointly input into the fusion model to obtain a prediction result of each common sample. Then, the third training loss is determined based on the prediction result and the corresponding sample label.

[0107] It may be understood that, in the fusion model, fusion processing is first performed on a plurality of first embedding vectors provided by the multiple parties for a same common sample, and then prediction processing is performed on a fusion vector obtained through fusion processing, to obtain a prediction result corresponding to the common sample. For example, fusion processing can be concatenation processing, addition processing, or contraposition multiplication processing.

[0108] In addition, there are a plurality of manners for determining the third training loss  $L_{cla}(\hat{y}, y)$  based on the prediction result  $\hat{y}$  and the sample label  $y$ . For example, a cross-entropy loss (cross-entropy loss) or a hinge loss (hinge loss) may be calculated.

[0109] The forward propagation process of the label party and the calculation manner of the

third training loss in the joint training process are described above.

[0110] It may be understood that, after the forward propagation process of joint training and the calculation of the training loss are determined, the reverse propagation process is determined accordingly. The following briefly describes reverse propagation processes of local models in the non-label party and the label party.

[0111] In general, a loss function of the fusion model in the label party may be expressed as follows:

$$L^{fuse} = L_{cla} \quad (8)$$

[0112] In the above-mentioned formula,  $L_{cla}$  represents the third training loss.

[0113] For two embedding models of each party in the multiple parties, a loss function of the first embedding model may be expressed as follows:

$$L^{(1)} = L_{cla} + \lambda_1 L_{con}^{(1)} + \lambda_2 L_{kd}^{(1)} \quad (9)$$

[0114] In the above-mentioned formula,  $L_{cla}$  represents the third training loss,  $L_{con}^{(1)}$  and  $L_{kd}^{(1)}$  respectively represents the first training loss and the second training loss of the first embedding model, and  $\lambda_1$  and  $\lambda_2$  are both super parameters with a value range of (0, 1). For example,  $\lambda_1 = 0.01$  and  $\lambda_2 = 0.02$  are set.

[0115] A loss function of a second embedding model (a non-first embedding model in the two embedding models) may be expressed as follows:

$$L^{(2)} = \beta_1 L_{con}^{(2)} + \beta_2 L_{kd}^{(2)} \quad (10)$$

[0116] In the above-mentioned formula,  $L_{con}^{(2)}$  and  $L_{kd}^{(2)}$  respectively represent a first training loss and a second training loss of the second embedding model, and  $\beta_1$  and  $\beta_2$  are both super parameters with a value range of (0, 1). For example,  $\beta_1 = \beta_2 = 0.05$  is set.

[0117] Therefore, the reverse propagation process may be performed based on the first training loss, the second training loss, and the third training loss.

[0118] (1) The label party trains the fusion model by using the third training loss  $L_{cla}$ , calculates, by using the third training loss  $L_{cla}$ , a gradient that is backhauled to the first embedding model deployed on each party in the multiple parties, and correspondingly provides the gradient for the multiple parties.

[0119] (2) The multiply parties (including the label party and the non-label party) each train a local first embedding model by using a calculated gradient that is backhauled to the local first

embedding model, and a first training loss  $L_{con}^{(1)}$  and a second training loss  $L_{kd}^{(1)}$  of the local first embedding model.

**[0120]** (3) The multiple parties (including the label party and the non-label party) each train a local second embedding model by using a first training loss  $L_{con}^{(2)}$  and a second training loss  $L_{kd}^{(2)}$  of the local second embedding model.

**[0121]** In general, in the multi-party joint training method for data privacy protection disclosed in embodiments of this specification, contrastive learning and knowledge distillation are used. In addition to sample data common to a plurality of participants, private sample data of each party is also introduced for model training, to improve and optimize prediction performance of a model obtained through joint training. In addition, different from a teacher model and a student model pre-specified in a conventional manner, distillation learning is used to determine a learning direction sample by sample based on two embedding models in a data party for stability of a same sample, so that an unstable model learns an embedding representation of a sample from a stable model, thereby improving learning effect of the model.

**[0122]** An embodiment of this specification further discloses a corresponding apparatus corresponding to the above-mentioned multi-party joint training method.

**[0123]** FIG. 5 is a schematic structural diagram of an apparatus when a multi-party joint training apparatus for data privacy protection is integrated into any non-label party according to an embodiment of this specification. Multiple parties hold different feature parts of a sample in a common sample set. A label party in the multiple parties further holds a sample label. As shown in FIG. 5, the apparatus 500 includes:

a contrastive learning module 510, configured to determine first training losses of two locally deployed embedding models through contrastive learning based on an original sample total set and a corresponding augmented sample set, where the original sample total set includes a private sample set and the common sample set; a distillation learning module 520, configured to use each sample in the original sample total set as a target sample, calculate stability indicators of the two embedding models for the target sample, use, as a label, an embedding vector that is output by an embedding model with higher stability for the target sample, and determine a second training loss of the other embedding model for the target sample; and an embedding vector sending module 530, configured to provide, for the label party, a first embedding vector that is output by a predetermined first embedding model in the two embedding models for each common sample, so that the label party determines a third training loss based on a first embedding vector provided by

each of the multiple parties, a locally deployed fusion model, and a held sample label, where the first training loss, the second training loss, and the third training loss are used for training a model locally deployed on each of the multiple parties.

**[0124]** In an embodiment, the apparatus 500 further includes a data augmentation module 540, configured to perform data augmentation processing on the original sample total set to obtain the augmented sample set.

**[0125]** In an embodiment, the stability indicator includes a representation information quantity, and that the distillation learning module 520 is configured to calculate the representation information quantity specifically includes: for each embedding model in the two embedding models, determining two representation information quantities corresponding to two embedding vectors respectively obtained after the target sample and a corresponding augmented sample are processed by using the embedding model, and determining a representation information quantity of the embedding model as a greater value in the two representation information quantities.

**[0126]** In a specific embodiment, that the distillation learning module 520 is configured to determine the two representation information quantities specifically includes: for each embedding vector in the two embedding vectors, using a maximum value element in the embedding vector as a representation information quantity of the embedding vector.

**[0127]** In an embodiment, the stability indicator includes a dispersion, and that the distillation learning module 520 is configured to calculate the dispersion specifically includes: for each embedding model in the two embedding models, determining, as a dispersion of the embedding model, a difference between two embedding vectors respectively obtained after the target sample and a corresponding augmented sample are processed by using the embedding model.

**[0128]** In an embodiment, that the distillation learning module 520 is configured to determine the difference specifically includes: calculating, as the difference, a quadratic sum of vector elements in a difference vector between the two embedding vectors, or a level 1 norm of the difference vector, or a level 2 norm of the difference vector.

**[0129]** In an embodiment, that the distillation learning module 520 is configured to determine the second training loss specifically includes: when it is determined that at least one of the two embedding models reaches a stability standard, based on a calculation result of the stability indicator, using, as the label, the embedding vector that is output by the embedding model with higher stability for the target sample, and determining the second training loss of the other

embedding model for the target sample.

**[0130]** In a specific embodiment, the distillation learning module 520 is further configured to discard the target sample when it is determined that neither of the two embedding models satisfies the stability standard.

5 **[0131]** Further, in an example, the stability indicator includes a representation information quantity and a dispersion, and the stability standard includes that the representation information quantity is greater than a first threshold and the dispersion is less than a second threshold.

**[0132]** In an embodiment, the apparatus 500 further includes: a target gradient receiving module 550, configured to receive a target gradient from the label party, where the target gradient  
10 is a gradient that is determined based on the third training loss and that is backhauled to the first embedding model deployed by the target party; and a first training module 560, configured to train the first embedding model by using the target gradient and a first training loss and a second training loss of the first embedding model.

**[0133]** In an embodiment, the two embedding models include the first embedding model and  
15 a second embedding model, and the apparatus 500 further includes: a second training module 570, configured to train the second embedding model by using a first training loss and a second training loss of the second embedding model.

**[0134]** FIG. 6 is a schematic structural diagram of an apparatus when a multi-party joint training apparatus for data privacy protection is integrated into a label party according to an  
20 embodiment of this specification. Multiple parties hold different feature parts of a sample in a common sample set. A label party in the multiple parties further holds a sample label. As shown in FIG. 6, the apparatus 600 includes:

a contrastive learning module 610, configured to determine first training losses of two locally deployed embedding models through contrastive learning based on an original sample total  
25 set and a corresponding augmented sample set, where the original sample total set includes a private sample set and the common sample set; a distillation learning module 620, configured to use each sample in the original sample total set as a target sample, calculate stability indicators of the two embedding models for the target sample, use, as a label, an embedding vector that is output by an embedding model with higher stability for the target sample, and determine a second training  
30 loss of the other embedding model for the target sample; and a supervised learning module 630, configured to determine a third training loss based on a first embedding vector that is output by a predetermined first embedding model in the two embedding models for each common sample, a

first embedding vector provided by each of other parties in the multiple parties, a locally deployed fusion model, and a held sample label, where the first training loss, the second training loss, and the third training loss are used for training a model locally deployed on each of the multiple parties.

**[0135]** In an embodiment, the supervised learning module 630 is specifically configured to:

jointly input, into the fusion model, the locally output first embedding vector and the first embedding vector provided by each of other parties in the multiple parties, to obtain a predicted result of each common sample; and determine the third training loss based on the predicted result and a corresponding sample label.

**[0136]** In a specific embodiment, the apparatus 600 further includes: a first training module 640, configured to train the fusion model by using the third training loss; and a gradient backhaul module 650, configured to calculate, based on the third training loss, a gradient that is backhauled to a first embedding model deployed on each of the multiple parties, so that each party trains a locally deployed first embedding model based on a corresponding gradient, and a first training loss and a second training loss of the locally deployed first embedding model.

**[0137]** In an embodiment, the two embedding models include the first embedding model and a second embedding model, and the apparatus 600 further includes: a second training module 660, configured to train the second embedding model by using a first training loss and a second training loss of the second embedding model.

**[0138]** An embodiment of another aspect further provides a computer-readable storage medium. The computer-readable storage medium stores a computer program. When the computer program is executed by a computer, the computer is enabled to perform the method described with reference to FIG. 3 or FIG. 4.

**[0139]** An embodiment of still another aspect further provides a computing device, including a memory and a processor. The memory stores executable code. When the processor executes the executable code, the method described with reference to FIG. 3 or FIG. 4 is implemented. A person skilled in the art should be aware that, in the above-mentioned one or more examples, functions described in the present invention can be implemented by hardware, software, firmware, or any combination thereof. When being implemented by software, these functions can be stored in a computer-readable medium or transmitted as one or more instructions or code on a computer-readable medium.

**[0140]** In the above-mentioned specific implementations, the objectives, technical solutions,

and beneficial effects of the present invention are further described in detail. It should be understood that the above-mentioned descriptions are merely specific implementations of the present invention, but are not intended to limit the protection scope of the present invention. Any modification, equivalent replacement, improvement, and the like made based on the technical  
5 solutions of the present invention shall fall within the protection scope of the present invention.

## CLAIMS

1. A multi-party joint training method for data privacy protection, wherein multiple parties hold different feature parts of a sample in a common sample set, a label party in the multiple parties further holds a sample label, and the method is applied to any target party of a non-label party in the multiple parties and comprises:

determining first training losses of two locally deployed embedding models through contrastive learning based on an original sample total set and a corresponding augmented sample set, wherein the original sample total set comprises a private sample set and the common sample set;

using each sample in the original sample total set as a target sample, calculating stability indicators of the two embedding models for the target sample, using, as a label, an embedding vector that is output by an embedding model with higher stability for the target sample, and determining a second training loss of the other embedding model for the target sample; and

providing, for the label party, a first embedding vector that is output by a predetermined first embedding model in the two embedding models for each common sample, so that the label party determines a third training loss based on a first embedding vector provided by each of the multiple parties, a locally deployed fusion model, and a held sample label, wherein the first training loss, the second training loss, and the third training loss are used for training a model locally deployed on each of the multiple parties.

2. The method according to claim 1, further comprising:

performing data augmentation processing on the original sample total set to obtain the augmented sample set.

3. The method according to claim 1, wherein the stability indicator comprises a representation information quantity, and calculation of the representation information quantity comprises:

for each embedding model in the two embedding models, determining two representation information quantities corresponding to two embedding vectors respectively obtained after the target sample and a corresponding augmented sample are processed by using the embedding model, and determining a representation information quantity of the embedding model as a greater value in the two representation information quantities.

4. The method according to claim 3, wherein the determining two representation information

quantities comprises:

for each embedding vector in the two embedding vectors, using a maximum value element in the embedding vector as a representation information quantity of the embedding vector.

5        5. The method according to claim 1, wherein the stability indicator comprises a dispersion, and calculation of the dispersion comprises:

for each embedding model in the two embedding models, determining, as a dispersion of the embedding model, a difference between two embedding vectors respectively obtained after the target sample and a corresponding augmented sample are processed by using the embedding model.

10        6. The method according to claim 5, wherein determining the difference comprises:  
calculating, as the difference, a quadratic sum of vector elements in a difference vector between the two embedding vectors, or a level 1 norm of the difference vector, or a level 2 norm of the difference vector.

15        7. The method according to claim 1, wherein the using, as a label, an embedding vector that is output by an embedding model with higher stability for the target sample, and determining a second training loss of the other embedding model for the target sample comprises:

when it is determined that at least one of the two embedding models reaches a stability standard, based on a calculation result of the stability indicator, using, as the label, the embedding vector that is output by the embedding model with higher stability for the target sample, and determining the second training loss of the other embedding model for the target sample.

20        8. The method according to claim 7, further comprising:

discarding the target sample when it is determined that neither of the two embedding models satisfies the stability standard.

25        9. The method according to claim 7 or 8, wherein the stability indicator comprises a representation information quantity and a dispersion, and the stability standard comprises that the representation information quantity is greater than a first threshold and the dispersion is less than a second threshold.

10. The method according to claim 1, wherein after the providing, for the label party, a first embedding vector that is output by a predetermined first embedding model in the two embedding models for each common sample, the method further comprises:

30        receiving a target gradient from the label party, wherein the target gradient is a gradient that is determined based on the third training loss and that is backhauled to the first embedding model

deployed by the target party; and

training the first embedding model by using the target gradient and a first training loss and a second training loss of the first embedding model.

11. The method according to claim 1 or 10, wherein the two embedding models comprise the first embedding model and a second embedding model, and the method further comprises:

training the second embedding model by using a first training loss and a second training loss of the second embedding model.

12. A multi-party joint training method for data privacy protection, wherein multiple parties hold different feature parts of a sample in a common sample set, a label party in the multiple parties further holds a sample label, and the method is applied to the label party and comprises:

determining first training losses of two locally deployed embedding models through contrastive learning based on an original sample total set and a corresponding augmented sample set, wherein the original sample total set comprises a private sample set and the common sample set;

using each sample in the original sample total set as a target sample, calculating stability indicators of the two embedding models for the target sample, using, as a label, an embedding vector that is output by an embedding model with higher stability for the target sample, and determining a second training loss of the other embedding model for the target sample; and

determining a third training loss based on a first embedding vector that is output by a predetermined first embedding model in the two embedding models for each common sample, a first embedding vector provided by each of other parties in the multiple parties, a locally deployed fusion model, and a held sample label, wherein the first training loss, the second training loss, and the third training loss are used for training a model locally deployed on each of the multiple parties.

13. The method according to claim 12, wherein determining a third training loss based on a first embedding vector that is output by a predetermined first embedding model in the two embedding models for each common sample, a first embedding vector provided by each of other parties in the multiple parties, a locally deployed fusion model, and a held sample label comprises:

jointly inputting, into the fusion model, the locally output first embedding vector and the first embedding vector provided by each of other parties in the multiple parties, to obtain a predicted result of each common sample; and

determining the third training loss based on the predicted result and a corresponding sample

label.

14. The method according to claim 12, wherein after the determining the third training loss, the method further comprises:

training the fusion model by using the third training loss; and

5 calculating, based on the third training loss, a gradient that is backhauled to a first embedding model deployed on each of the multiple parties, so that each party trains a locally deployed first embedding model based on a corresponding gradient, and a first training loss and a second training loss of the locally deployed first embedding model.

15. The method according to claim 12, wherein the two embedding models comprise the first embedding model and a second embedding model, and the method further comprises:

10 training the second embedding model by using a first training loss and a second training loss of the second embedding model.

16. A multi-party joint training method for data privacy protection, wherein multiple parties hold different feature parts of a sample in a common sample set, a label party in the multiple parties further holds a sample label, and the method comprises:

15 determining, by each of multiple parties, first training losses of two local embedding models through contrastive learning based on a local original sample total set and a corresponding augmented sample set, wherein the original sample total set comprises a private sample set and the common sample set;

20 using, by each of the multiple parties, each sample in the original sample total set as a target sample, calculating stability indicators of the two embedding models for the target sample, using, as a label, an embedding vector that is output by an embedding model with higher stability for the target sample, and determining a second training loss of the other embedding model for the target sample; and

25 determining, by a label party, a third training loss based on a first embedding vector that is output by a predetermined first embedding model in the two local embedding models for each common sample, a first embedding vector provided by each of other parties in the multiple parties, a locally deployed fusion model, and a held sample label, wherein the first training loss, the second training loss, and the third training loss are used for training a model locally deployed on each of  
30 the multiple parties.

17. A multi-party joint training apparatus for data privacy protection, wherein multiple parties

hold different feature parts of a sample in a common sample set, a label party in the multiple parties further holds a sample label, and the apparatus is integrated into any target party of a non-label party in the multiple parties and comprises:

a contrastive learning module, configured to determine first training losses of two locally  
5 deployed embedding models through contrastive learning based on an original sample total set and a corresponding augmented sample set, wherein the original sample total set comprises a private sample set and the common sample set;

a distillation learning module, configured to use each sample in the original sample total set as a target sample, calculate stability indicators of the two embedding models for the target sample,  
10 use, as a label, an embedding vector that is output by an embedding model with higher stability for the target sample, and determine a second training loss of the other embedding model for the target sample; and

an embedding vector sending module, configured to provide, for the label party, a first embedding vector that is output by a predetermined first embedding model in the two embedding  
15 models for each common sample, so that the label party determines a third training loss based on a first embedding vector provided by each of the multiple parties, a locally deployed fusion model, and a held sample label, wherein the first training loss, the second training loss, and the third training loss are used for training a model locally deployed on each of the multiple parties.

18. A multi-party joint training apparatus for data privacy protection, wherein multiple parties  
20 hold different feature parts of a sample in a common sample set, a label party in the multiple parties further holds a sample label, and the apparatus is integrated into the label party and comprises:

a contrastive learning module, configured to determine first training losses of two locally  
25 deployed embedding models through contrastive learning based on an original sample total set and a corresponding augmented sample set, wherein the original sample total set comprises a private sample set and the common sample set;

a distillation learning module, configured to use each sample in the original sample total set as a target sample, calculate stability indicators of the two embedding models for the target sample,  
30 use, as a label, an embedding vector that is output by an embedding model with higher stability for the target sample, and determine a second training loss of the other embedding model for the target sample; and

a supervised learning module, configured to determine a third training loss based on a first

embedding vector that is output by a predetermined first embedding model in the two embedding models for each common sample, a first embedding vector provided by each of other parties in the multiple parties, a locally deployed fusion model, and a held sample label, wherein the first training loss, the second training loss, and the third training loss are used for training a model locally  
5 deployed on each of the multiple parties.

19. A computer-readable storage medium, wherein the computer-readable storage medium stores a computer program, and when the computer program is executed in a computer, the computer is enabled to perform the method according to any one of claims 1 to 16.

20. A computing device, comprising a memory and a processor, wherein the memory stores  
10 executable code, and when executing the executable code, the processor implements the method according to any one of claims 1 to 16.