METHOD, APPARATUS, AND DEVICE FOR TRANSACTION RISK CONTROL BASED ON GRAPHICAL STRUCTURE MODEL

TECHNICAL FIELD

[1] The present specification relates to the field of computer software technologies, and in particular, to a method, apparatus, and device for transaction risk control based on a graphical structure model.

BACKGROUND

[2] With the rapid development of computer and Internet technologies, many services can be carried out through the Internet, typically e-commerce and payment services. Based on e-commerce and payment platforms, users can conduct transactions conveniently. These platforms bring convenience as well as risks to the users. Some users with abnormal behaviors such as cheating might threaten transaction security and cause losses to funds or other interests of other users.

[3] In existing technologies, risk control on user transactions often relies on user complaints, manual reviews on platforms, etc.

[4] Based on the existing technologies, more effective solutions for transaction risk control are needed.

SUMMARY

[5] Implementations of the present specification provide a method, apparatus, and device for transaction risk control based on a graphical structure model, to alleviate the following technical problem: lack of a more effective solution for transaction risk control.

[6] To alleviate the technical problem, the implementations of the present specification are implemented as follows.

[7] An implementation of the present specification provides a method for transaction risk control based on a graphical structure model, including: obtaining a graphical structure model trained by using labeled samples, where the graphical structure model

is defined based on a transaction data network, and the transaction data network includes one or more nodes of a buyer user, a seller user, an e-commerce platform account, a third-party payment platform account, a bank card, and a product, and a relationship between the nodes; and the graphical structure model is configured to calculate embedding vectors of the nodes obtained after a plurality of iterations in a latent feature space based on original features of the nodes and/or features of associated edges between the nodes; calculating embedding vectors corresponding to to-be-tested samples by using the trained graphical structure model; and performing transaction risk control on the to-be-tested samples based on the corresponding embedding vectors; where the labeled samples are labeled whether a node corresponding to the labeled samples is a transaction risk node.

[8] An implementation of the present specification provides an apparatus for transaction risk control based on a graphical structure model, including: an acquisition module, configured to obtain a graphical structure model trained by using labeled samples, where the graphical structure model is defined based on a transaction data network, and the transaction data network includes one or more nodes of a buyer user, a seller user, an e-commerce platform account, a third-party payment platform account, a bank card, and a product, and a relationship between the nodes; and the graphical structure model is configured to calculate embedding vectors of the nodes obtained after a plurality of iterations in a latent feature space based on original features of the nodes and/or features of associated edges between the nodes; a calculation module, configured to calculate embedding vectors corresponding to to-be-tested samples by using the trained graphical structure model; and a risk control module, configured to perform transaction risk control on the to-be-tested samples based on the embedding vectors corresponding to the to-be-tested samples; where the labeled samples are labeled whether a node corresponding to the labeled samples is a transaction risk node.

[9] An implementation of the present specification provides a device for transaction risk control based on a graphical structure model, including: at least one processor; and at least one memory connects to and communicates with the at least one processor, where the memory stores an instruction executable by the at least one processor, and the instruction is executed by the at least one processor to enable the at least one processor to: obtain a graphical structure model trained by using labeled samples, where the graphical structure model is defined based on a transaction data network,

and the transaction data network includes one or more nodes of a buyer user, a seller user, an e-commerce platform account, a third-party payment platform account, a bank card, and a product, and a relationship between the nodes; and the graphical structure model is configured to calculate embedding vectors of the nodes obtained after a plurality of iterations in a latent feature space based on original features of the nodes and/or features of associated edges between the nodes; calculate embedding vectors corresponding to to-be-tested samples by using the trained graphical structure model; and perform transaction risk control on the to-be-tested samples based on the embedding vectors corresponding to the to-be-tested samples; where the labeled samples are labeled whether a node corresponding to the labeled samples is a transaction risk node.

[10] At least one of the described technical solutions adopted in the implementations of the present specification can achieve the following beneficial effects: structure features, node features, and edge features of a transaction data network can be effectively integrated, embedding vectors and/or a prediction probability corresponding to a user can be calculated, and transaction risk control can be effectively performed on the user based on the embedding vectors and/or the prediction probability.

BRIEF DESCRIPTION OF DRAWINGS

[11] To describe the technical solutions in the implementations of the present specification or in the existing technology more clearly, the following briefly describes the accompanying drawings needed for describing the implementations or the existing technology. Apparently, the accompanying drawings in the following description are merely some implementations of the present specification, and a person of ordinary skill in the art can derive other drawings from these accompanying drawings without creative efforts.

[12] FIG. 1 is a schematic diagram illustrating an overall architecture related to a solution of the present specification in an actual application scenario;

[13] FIG. 2 is a schematic flowchart illustrating a graphical structure model training method, according to an implementation of the present specification;

[14] FIG. 3 is a schematic flowchart illustrating a method for transaction risk control based on a graphical structure model, according to an implementation of the present

specification;

[15] FIG. 4 is a schematic structural diagram illustrating a graphical structure model training apparatus corresponding to FIG. 2, according to an implementation of the present specification; and

[16] FIG. 5 is a schematic structural diagram illustrating an apparatus for transaction risk control based on a graphical structure model corresponding to FIG. 3, according to an implementation of the present specification.

DESCRIPTION OF EMBODIMENTS

[17] Implementations of the present specification provide a graphical structure model training method, apparatus and device, and a method, apparatus, and device for transaction risk control based on a graphical structure model.

[18] To enable a person skilled in the art to better understand the technical solutions in the present specification, the following clearly and comprehensively describes the technical solutions in the implementations of the present specification with reference to the accompanying drawings in the implementations of the present specification. Apparently, the described implementations are merely some rather than all of the implementations of the present application. Based on the implementations of the present specification, other embodiments obtained by a person of ordinary skill in the art without creative efforts shall fall within the protection scope of the present application.

[19] In e-commerce and payment services, platforms accumulate a great deal of transaction data and fund payment data. These services often include a lot of abnormal behaviors, such as cash-out, fraud transaction, cheating, etc. These abnormal behaviors can cause risks to user transaction security. Therefore, how to identify risky buyers and sellers becomes an industry problem.

[20] In the e-commerce and payment services, the platforms also accumulate a great deal of network relationship data, such as a transaction relationship between a buyer and a seller, a purchase and collection relationship between a user and a product, an upstream and downstream relationship between merchants of a supply chain, a family and social attribute relationship between users, a capital inflow and outflow relationship between a user and a bank card, a relationship formed by browsing behaviors or movement tracks of a user during a period of time, etc. There is a

complete interest chain behind each abnormal behavior. These interest chains form a network of crimes. The present disclosure considers learning abnormal transaction patterns from these complex networks for risk control, to alleviate a user transaction security problem.

[21] In conventional machine learning and data analysis, different samples often lack information interaction, and nodes can be considered as individuals. However, different samples often have very close connections in practice. For example, a cash-out gang organizes cash-out and fraud for a group of buyers, which implies a whole set of organized and planned transactions and fund network relationships. Combining interaction behaviors of different nodes helps better identify a target group and regular patterns behind. Conventional deep learning technologies cannot be used for representation and feature learning of composite structure data such as graphics, networks, and dynamic networks. Deriving network features depending on a manual method is time-consuming and results are often uncontrollable. Even experienced data analysts can enumerate only some network features within their cognitive scopes. In addition, network structures in actual applications change continuously over time. Relying on service experience for network attacks and defense is inadequate.

[22] The present disclosure can effectively integrate node features and edge features, support a plurality of types of heterogeneous networks, and adapt to network changes. A training process can optimize the graphical structure model directly for a service objective, and then risk control can be performed effectively to ensure user transaction security.

[23] FIG. 1 is a schematic diagram illustrating an overall architecture related to a solution of the present specification in an actual application. The overall architecture mainly involves three parts: each buyer end, each seller end, and a platform server. The platform server can include one or more devices. The platform server builds a transaction data network based on user transaction-related data between a buyer end and a seller end, defines a corresponding graphical structure model and trains the model to perform risk control on a user transaction between a buyer end and a seller end.

[24] The following describes the solution of the present specification in detail based on the architecture in FIG. 1.

[25] FIG. 2 is a schematic flowchart illustrating a graphical structure model training method, according to an implementation of the present specification. A process in FIG.

2 includes the following steps.

[26] S202: Build a transaction data network based on user transaction-related data, where the transaction data network includes one or more nodes of a buyer user, a seller user, an e-commerce platform account, a third-party payment platform account, a bank card, and a product, and a relationship between the nodes.

[27] In an implementation of the present specification, the user transaction-related data can include transaction data recorded in orders, such as a transaction amount, a transaction time, and attribute information of a buyer and a seller. The data can further include environmental information of a buyer and a seller at the time of transaction, such as information about devices used by the buyer and the seller, a location based service (LBS) information, a network IP address, and Wi-Fi information.

[28] In an implementation of the present specification, the buyer user and the seller user both can refer to a natural person, and can be specifically represented by using an identity of the natural person, such as an identity card number and biological feature information. The natural person and an account of any platform are generally in a one-to-one relationship, but may alternatively be in a one-to-many relationship. For example, a buyer user might have a plurality of third-party payment platform accounts.

[29] In addition to the buyer user node and the seller user node, other nodes listed previously can also be represented by corresponding identification information. For example, a bank card can be represented by a corresponding bank card number, and a product can be represented by a corresponding product number or a corresponding product name and model.

[30] In an implementation of the present specification, the transaction data network can include one type of network or a combination of multiple types of heterogeneous networks and can be built to adapt to an actual demand.

[31] For example, the transaction data network can include a transaction network formed by a buyer user and a seller user, where accordingly a network node can represent a buyer user or a seller user, and an edge can represent an existing transaction relationship between the buyer user and the seller user; a relationship network of a buyer user and a debit card corresponding to a buyer user account, where accordingly a network node can represent a user account or a debit card, and an edge can represent a specific user account of the debit card; a relationship network of a buyer user and a credit card corresponding to a buyer user account, where accordingly

a network node can represent a user account or a credit card, and an edge can represent a specific user account of the credit card; a same-person relationship network between user accounts, where accordingly a network node can represent a user account, and an edge can represent which user accounts belong to the same user; a family relationship network between user accounts, where accordingly a network node can represent a user account, and an edge can represent specific user accounts, and users corresponding to the user accounts belong to the same family; and a network formed by a buyer user and a product, where accordingly a network node can represent a buyer user or a product, and an edge can represent a buyer user that buys a product.

[32] In an implementation of the present specification, the transaction data network is graph data, which is intuitive and can be conveniently processed by using a graph algorithm. Therefore, efficiency is relatively high.

[33] Nodes in the transaction data network can contain original features of the nodes. The original features include at least one of the following types of transaction data: a transaction amount, a quantity of transactions, a transaction time, a transaction location, a transaction target, a transaction payment method, etc. Specifically, for example, the original features include a transaction amount that exceeds certain scales within a period of time, a quantity of transactions, and a quantity of transaction days, a quantity of cities in which transactions occur, a quantity of devices used, an amount and proportion of debit card transactions, an amount and proportion of credit card transactions, an amount and proportion of credit transactions other than credit card transactions, etc.

[34] Edges in the transaction data network can include associated features of nodes that are connected by the edges, for example, transaction data and/or interpersonal relationship data jointly involved by the nodes connected to the edges. Assume that the two nodes connected to the edge are respectively a certain buyer user and a certain seller user. In such a case, features of the edge can include, for example, between the buyer user and the seller user within a period of time, a transaction amount, a quantity of transactions, a quantity of cities in which transactions occur, and information about whether the buyer user and the seller user are friends.

[35] S204: Define a graphical structure model based on the transaction data network, where the graphical structure model is configured to calculate an embedding vector of the node obtained after a plurality of iterations in a latent feature space based on an

original feature of the node and/or a feature of an associated edge between the nodes.

[36] In an implementation of the present specification, the original features and the features of the edges of the nodes in the transaction data network can be integrated to define a feature space. Furthermore, the graphical structure model is defined based on one or more feature spaces, a pending weight matrix of the feature space, and a specified nonlinear transformation function.

[37] The weight matrix is used to perform a linear transformation on the feature space. The nonlinear transformation function is used to perform a nonlinear transformation on the feature space. Therefore, the original feature and edge feature of the node can finally be mapped to a more complex latent feature space by using the graphical structure model. The embedding vector can reflect a feature embedded by the corresponding node into the final latent feature space.

[38] When there are a plurality of iterations, the feature space can further be defined based on a previous iteration result. Compared with a feature space defined based on the original feature or the feature of the edge of the node, the feature space can also be considered as a latent feature space.

[39] Based on the previous analysis, an embedding vector of the node obtained after each iteration in the latent feature space is calculated based one or a combination of the following factors: the original feature of the node, a feature of an associated edge between the node and another node, and an embedding vector of the node obtained after one or more previous iterations in the latent feature space.

[40] S206: Train the graphical structure model by using labeled samples, where the labeled samples are labeled whether a node corresponding to the labeled samples is a transaction risk node.

[41] In an implementation of the present specification, some existing samples can be labeled in advance as risk samples or non-risk samples to train a graphical structure model. Any sample can correspond to a node in the graphical structure model. A node corresponding to an abnormal sample is an abnormal node. During transaction risk control, a sample is generally a labeled buyer user or seller user, or a corresponding user account. Certainly, a sample can also refer to another element involved in a transaction, such as a debit card or a credit card bound to the user account. These elements can also serve as nodes in the graphical structure model.

[42] The following implementations are mainly described by using examples that samples are users. Generally, based on an actual complaint penalty or a manual

review result, a buyer user and a seller user who are identified as having abnormal behaviors such as cheating can be regarded as risk users and labeled. In an actual service, they can be labeled once a day, or labeled for several days together, etc.

[43] A standard used for labeling can be defined based on an actual demand. For example, if a user is identified as having abnormal behaviors in historical transactions, and a quantity of the historical transactions is not less than a specified quantity, the user is labeled as a transaction risk user (namely, a transaction risk node in the graphical structure model). For another example, if a user is identified as having repayment overdue behaviors on a credit card bound to a transaction account for a quantity of times, and the quantity of times is not less than a specified quantity, the user is labeled as a transaction risk user.

[44] In an implementation of the present specification, training the graphical structure model aims to determine an appropriate weight matrix so that an embedding vector calculated based on the trained graphical structure model can express a nature of a node corresponding to the embedding vector as truthfully as possible, for example, whether the node is a transaction risk node.

[45] The method in FIG. 2 can effectively integrate structure features, node features, and edge features of a heterogeneous or homogeneous transaction data network, define and train a graphical structure model, and calculate embedding vectors of nodes in a latent feature space based on the graphical structure model. The embedding vectors are used for risk control on a user transaction.

[46] Based on the method in FIG. 2, an implementation of the present specification further provides some specific implementing solutions of the method as well as extensions to continue the description.

[47] In an implementation of the present specification, the graphical structure model is further configured to calculate a prediction probability of the node based on the embedding vector. During transaction risk control, the prediction probability can represent a probability that a node is a transaction risk node.

[48] In an implementation of the present specification, a description and a training process of the graphical structure model are intuitively illustrated by using an example. Assume that T represents a quantity of iterations; there are n nodes in total; y_i represents a labeling result of the ith node (for example, a risk node is labeled as 1 and a non-risk node is labeled as 0); F_i represents original features of the ith node

(a feature dimension is f); E_{ij} represents features of associated edges between the ith node and the jth node (a feature dimension is e); N_i represents a set of adjacent nodes of the ith node; a dimension of a final latent feature space that needs feature embedding is h; and H_i^t represents embedding vectors of the ith node obtained after t iterations in the latent feature space.

[49] Assume that an embedding vector of the node obtained after t iterations in the latent feature space is calculated based on the original feature of the node, a feature of an associated edge between the node and another node, and an embedding vector of the node obtained after $t-1$ iterations in the latent feature space. An example graphical structure model is defined as follows:

„ W_1 ($h*f$ dimensions), W_2 ($h*e$ dimensions), W_3 ($h*h$ dimensions), and W_4 ($h*2$ dimensions) are initialized, for example, by using standard Gaussian distribution;

for t=1 to T: //executing T iterations

for i=0 to $n-1$: //traversing n nodes

$$H_i^t = \sigma\left(W_1 F_i + W_2 \sum_{j \in N_i} E_{ij} + W_3 \sum_{j \in N_i} \alpha_{ij} H_i^{t-1} \right);$$

//(equation 1) a process of the tth iteration of the ith node.

for i=0 to $n-1$: //traversing n nodes

$$pred_i = softmax\left(W_4 * H_i^T \right);$$

//(equation 2) a prediction probability of the ith node obtained after T iterations

optimize $\arg\min_{w_1,w_2,w_3,w_4} \sum_i corss_entrep\left(pred_i, y_i \right)$; and

//(equation 3) a prediction probability of the ith node obtained after T iterations; where

σ represents a nonlinear transformation function (such as Relu, Sigmoid, Tanh, etc.), W_1, W_2, W_3, and W_4 represent weight matrices, α_{ij} represents a weight matrix (such as a diagonal matrix) for the ith node and the jth node, $pred_i$ represents a prediction probability of the i th node obtained after T iterations, and

$corss_entrep$ represents a function used to calculate cross entropy.

[50] In the previous example, training the graphical structure model by using labeled samples in step S206 can specifically include: optimizing $\arg\min_{w_1,w_2,w_3,w_4} \sum_i corss_entrep(pred_i, y_i)$ by using a back-propagation algorithm and the labeled samples to obtain optimal W_1, W_2, W_3, and W_4. The optimization objective here is set based on cross entropy of a prediction probability and a labeling result. It is worthwhile to note that the optimization objective can be set based on a parameter other than the cross entropy, where the parameter can measure consistency between a prediction probability and a labeling result. The optimization objective is to maximize the consistency between a prediction probability and a labeling result.

[51] The described equations 1, 2 and 3 are examples and not unique. For example, respective items of F_i, E_{ij}, H_i^{t-1} in equation 1 can be transformed through multiplication, exponent or logarithm, or two of the items can be combined, or one of the items can be deleted. For another example, the $softmax$ function in equation 2 can be replaced with another suitable normalization function.

[52] Furthermore, after training the graphical structure model, an embedding vector and/or a prediction probability of an existing sample and an embedding vector and/or a prediction probability of a new sample can be calculated by using the trained graphical structure model through forward propagation. Further, the calculated embedding vector and/or prediction probability can be used for a sample-related service. For example, the embedding vector is used as input of other models (for example, used as input to other supervised, unsupervised or reinforcement learning for re-integration), or the embedding vector or the prediction probability is directly used to determine a nature of a corresponding sample. Transaction risk control is used as an example. When a prediction probability of a user is higher than a certain threshold, the user can be identified as a transaction risk user.

[53] Based on the same idea, an implementation of the present specification further provides a method for transaction risk control based on a graphical structure model. FIG. 3 is a schematic flowchart of the method for transaction risk control. A process in FIG. 3 includes the following steps.

[54] S302: Obtain a graphical structure model trained by using labeled samples,

where the graphical structure model is defined based on a transaction data network, and the transaction data network includes one or more nodes of a buyer user, a seller user, an e-commerce platform account, a third-party payment platform account, a bank card, and a product, and a relationship between the nodes; and the graphical structure model is configured to calculate embedding vectors of the nodes obtained after a plurality of iterations in a latent feature space based on original features of the nodes or features of associated edges between the nodes.

[55] In an implementation of the present specification, the graphical structure model in FIG. 3 is trained by the method in FIG. 2, and the method in FIG. 3 can be understood with reference to the previous description of FIG. 2.

[56] S304: Calculate an embedding vector corresponding to a to-be-tested sample by using the trained graphical structure model.

[57] S306: Perform transaction risk control on the to-be-tested samples based on the corresponding embedding vectors, where the labeled samples are labeled whether a node corresponding to the labeled samples is a transaction risk node.

[58] In an actual application, a prediction probability can intuitively represent a risk of a sample. Therefore, a prediction probability corresponding to the to-be-tested sample can further be calculated by using the trained graphical structure model based on the embedding vector corresponding to the to-be-tested sample, and transaction risk control can be performed on the to-be-tested sample based on the prediction probability corresponding to the to-be-tested sample in step S306.

[59] It is worthwhile to note that the previous implementations are mainly described by using risk control as an example and the graphical structure model can alternatively be used in other circumstances. For example, if a sample labeling result indicates a certain type of hobby of a user, embedding vectors and/or a prediction probability calculated based on the graphical structure model can be used for a friend recommendation function in an instant messaging service, so that the user can be automatically matched with a friend having a similar hobby.

[60] Based on the same idea, implementations of the present specification further provide apparatuses corresponding to the described methods, as shown in FIG. 4 and FIG. 5.

[61] FIG. 4 is a schematic structural diagram illustrating a graphical structure model training apparatus corresponding to FIG. 2, according to an implementation of the present specification. The apparatus includes the following: a building module 401,

configured to build a transaction data network based on user transaction-related data, where the transaction data network includes one or more nodes of a buyer user, a seller user, an e-commerce platform account, a third-party payment platform account, a bank card, and a product, and a relationship between the nodes; a defining module 402, configured to define a graphical structure model based on the transaction data network, where the graphical structure model is configured to calculate embedding vectors of the nodes obtained after a plurality of iterations in a latent feature space based on original features of the nodes and/or features of associated edges between the nodes; and a training module 403, configured to train the graphical structure model by using labeled samples, where the labeled samples are labeled whether a node corresponding to the labeled samples is a transaction risk node.

[62] FIG. 5 is a schematic structural diagram illustrating an apparatus for transaction risk control based on a graphical structure model corresponding to FIG. 3, according to an implementation of the present specification. The apparatus includes the following: an acquisition module 501, configured to obtain a graphical structure model trained by using labeled samples, where the graphical structure model is defined based on a transaction data network, and the transaction data network includes one or more nodes of a buyer user, a seller user, an e-commerce platform account, a third-party payment platform account, a bank card, and a product, and a relationship between the nodes; and the graphical structure model is configured to calculate embedding vectors of the nodes obtained after a plurality of iterations in a latent feature space based on original features of the nodes and/or features of associated edges between the nodes; a calculation module 502, configured to calculate embedding vectors corresponding to to-be-tested samples by using the trained graphical structure model; and a risk control module 503, configured to perform transaction risk control on the to-be-tested samples based on the corresponding embedding vectors; where the labeled samples are labeled whether a node corresponding to the labeled samples is a transaction risk node.

[63] Optionally, the graphical structure model is further configured to calculate a prediction probability of the node based on the embedding vector, where the prediction probability represents a probability that the node is a transaction risk node.

[64] The risk control module 503 performs transaction risk control on the to-be-tested samples based on the corresponding embedding vectors, specifically including: calculating, by the risk control module 503, a prediction probability corresponding to

the to-be-tested sample by using the trained graphical structure model based on the embedding vector corresponding to the to-be-tested sample; and performing transaction risk control on the to-be-tested sample based on the prediction probability corresponding to the to-be-tested sample.

[65] Optionally, the transaction data network is built based on user transaction-related data.

[66] Optionally, the transaction data network includes one or a combination of the following networks: a transaction network formed by a buyer user and a seller user, a relationship network of a buyer user and a debit card corresponding to a buyer user account, a relationship network of a buyer user and a credit card corresponding to a buyer user account, a same-person relationship network between user accounts, a family relationship network between user accounts, and a network formed by a buyer user and a product, where the user account includes an e-commerce platform account and/or a third-party payment platform account.

[67] Optionally, the original feature includes at least one of the following types of transaction data: a transaction amount, a quantity of transactions, a transaction time, a transaction location, a transaction target, and a transaction payment method; and the feature of the edge includes transaction data and/or interpersonal relationship data jointly involved by the nodes connected to the edge.

[68] Optionally, an embedding vector of the node obtained after t iterations in the latent feature space is calculated based on the original feature of the node, a feature of an associated edge between the node and another node, and an embedding vector of the node obtained after $t-1$ iterations in the latent feature space.

[69] Optionally, the calculating an embedding vector of the node obtained after a plurality of iterations in a latent feature space based on an original feature of the node and a feature of an associated edge between the nodes specifically includes the following: calculating, by the calculation module 502, the embedding vector of the node obtained after the plurality of iterations in the latent feature space based on the following equation:

$$H_i^t = \sigma\left(W_1 F_i + W_2 \sum_{j \in N_i} E_{ij} + W_3 \sum_{j \in N_i} \alpha_{ij} H_i^{t-1} \right)$$; where

H_i^t represents an embedding vector of the ith node obtained after t iterations in the

latent feature space; σ represents a nonlinear transformation function; W_1, W_2, and W_3 represent weight matrices; F_i represents original features of the ith node; E_{ij} represents features of associated edges between the ith node and the jth node; N_i represents a set of adjacent nodes of the ith node; and α_{ij} represents a weight matrix for the ith node and the jth node.

[70] Optionally, the calculating a prediction probability of the node based on the embedding vector specifically includes the following: calculating, by the calculation module 502, the prediction probability of the node based on the following equation: $pred_i = softmax\left(W_4 * H_i^T\right)$; where

$pred_i$ represents a prediction probability of the ith node obtained after T iterations, and W_4 represents a weight matrix.

[71] Optionally, training the graphical structure model by using the labeled samples specifically includes the following: training the graphical structure model by using the labeled samples with a training objective that consistency between the prediction probability and a sample labeling result corresponding to the prediction probability is maximized.

[72] Optionally, training the graphical structure model by using the labeled samples specifically includes the following: optimizing $\arg\min_{w_1,w_2,w_3,w_4} \sum_i corss_entrep\left(pred_i, y_i\right)$ by using a back-propagation algorithm and the labeled samples to obtain optimal W_1, W_2, W_3, and W_4; where $corss_entrep$ represents a function used to calculate cross entropy.

[73] Based on the same idea, an implementation of the present specification provides a graphical structure model training device corresponding to FIG. 2, including: at least one processor; and at least one memory connects to and communicates with the at least one processor, where the memory stores an instruction executable by the at least one processor, and the instruction is executed by the at least one processor to enable the at least one processor to: build a transaction data network based on user transaction-related data, where the transaction data network includes one or more nodes of a buyer user, a seller user, an e-commerce platform account, a third-party payment platform account, a bank card, and a product, and a relationship between the

nodes; define a graphical structure model based on the transaction data network, where the graphical structure model is configured to calculate embedding vectors of the nodes obtained after a plurality of iterations in a latent feature space based on original features of the nodes and/or features of associated edges between the nodes; and train the graphical structure model by using labeled samples, where the labeled samples are labeled whether a node corresponding to the labeled samples is a transaction risk node.

[74] Based on the same idea, an implementation of the present specification provides a device for transaction risk control based on a graphical structure model corresponding to FIG. 3, including: at least one processor; and at least one memory connects to and communicates with the at least one processor, where the memory stores an instruction executable by the at least one processor, and the instruction is executed by the at least one processor to enable the at least one processor to: obtain a graphical structure model trained by using labeled samples, where the graphical structure model is defined based on a transaction data network, and the transaction data network includes one or more nodes of a buyer user, a seller user, an e-commerce platform account, a third-party payment platform account, a bank card, and a product, and a relationship between the nodes; and the graphical structure model is configured to calculate embedding vectors of the nodes obtained after a plurality of iterations in a latent feature space based on original features of the nodes and/or features of associated edges between the nodes; calculate embedding vectors corresponding to to-be-tested samples by using the trained graphical structure model; and perform transaction risk control on the to-be-tested samples based on the corresponding embedding vectors; where the labeled samples are labeled whether a node corresponding to the labeled samples is a transaction risk node.

[75] Based on the same idea, an implementation of the present specification further provides a non-volatile computer storage medium corresponding to FIG. 2. The non-volatile computer storage medium stores a computer executable instruction. The computer executable instruction is set to perform the following: building a transaction data network based on user transaction-related data, where the transaction data network includes one or more nodes of a buyer user, a seller user, an e-commerce platform account, a third-party payment platform account, a bank card, and a product, and a relationship between the nodes; defining a graphical structure model based on the transaction data network, where the graphical structure model is configured to

calculate embedding vectors of the nodes obtained after a plurality of iterations in a latent feature space based on original features of the nodes and/or features of associated edges between the nodes; and training the graphical structure model by using labeled samples, where the labeled samples are labeled whether a node corresponding to the labeled samples is a transaction risk node.

[76] Based on the same idea, an implementation of the present specification further provides a non-volatile computer storage medium corresponding to FIG. 3. The non-volatile computer storage medium stores a computer executable instruction. The computer executable instruction is set to perform the following: obtaining a graphical structure model trained by using labeled samples, where the graphical structure model is defined based on a transaction data network, and the transaction data network includes one or more nodes of a buyer user, a seller user, an e-commerce platform account, a third-party payment platform account, a bank card, and a product, and a relationship between the nodes; and the graphical structure model is configured to calculate embedding vectors of the nodes obtained after a plurality of iterations in a latent feature space based on original features of the nodes and/or features of associated edges between the nodes; calculating embedding vectors corresponding to to-be-tested samples by using the trained graphical structure model; and performing transaction risk control on the to-be-tested samples based on the corresponding embedding vectors; where the labeled samples are labeled whether a node corresponding to the labeled samples is a transaction risk node.

[77] The specific implementations of the present specification are described previously. Other implementations shall fall within the scope of the appended claims. In some cases, actions or steps described in the claims can be performed in a sequence different from a sequence in an implementation and still can achieve a desired result. In addition, a process depicted in the accompanying drawings does not necessarily need a particular sequence or consecutive sequence as shown to achieve the desired result. In some implementations, multi-task processing and parallel processing are also possible or can be advantageous.

[78] The implementations in the present specification are described in a progressive way. For a same or similar part in the implementations, reference can be made to each other, and each implementation focuses on differences from another implementation. In particular, for an implementation of an apparatus, a device, or a non-volatile computer storage medium, description is relatively simple because the

implementation of the apparatus, the device, or the non-volatile computer storage medium is basically similar to a method implementation, and reference can be made to some descriptions of the method embodiment.

[79] The apparatus, the device, and the non-volatile computer storage medium provided in the implementations of the present specification correspond to the method. Therefore, the apparatus, the device, and the non-volatile computer storage medium also have a beneficial technical effect similar to a beneficial technical effect of the corresponding method. Since the beneficial technical effect of the method has been described in detail previously, the beneficial technical effect of the apparatus, the device, and the non-volatile computer storage medium that correspond to the method is omitted here for simplicity.

[80] In the 1990s, improvements to a technology could clearly be distinguished from improvements to hardware (for example, improvements to circuit structures such as diodes, transistors, switches, etc.) or software (improvements to method processes). However, with the development of technologies, improvements of many method processes can be regarded as direct improvements of hardware circuit structures now. Designers almost always program an improved method process into a hardware circuit to get a corresponding hardware circuit structure. Therefore, it is possible to improve a method process by using a hardware entity module. For example, a programmable logic device (PLD) (for example, field programmable gate array (FPGA)) is such an integrated circuit, and a logical function of the PLD is determined by a user through programming the device. Designers independently program to "integrate" a digital system on a PLD without asking a chip manufacturer to design and prepare a dedicated integrated circuit chip. In addition, at present, instead of manually manufacturing an integrated chip, such programming is mostly implemented by using "logic compiler" software, which is similar to a software compiler used to develop and write a program. Original code needs to be written in a particular programming language for compilation. The language is referred to as hardware description language (HDL). There are many HDLs, such as the Advanced Boolean Expression Language (ABEL), the Altera Hardware Description Language (AHDL), Confluence, the Cornell University Programming Language (CUPL), HDCal, the Java Hardware Description Language (JHDL), Lava, Lola, MyHDL, PALASM, and the Ruby Hardware Description Language (RHDL). The very-high-speed integrated circuit hardware description language (VHDL) and Verilog are most commonly used. A

person skilled in the art should also understand that a hardware circuit that implements a logical method procedure can be readily obtained once the method procedure is logically programmed by using the several described hardware description languages and is programmed into an integrated circuit.

[81] A controller can be implemented by using any appropriate method. For example, the controller can be a microprocessor or a processor, or a computer-readable medium, a logic gate, a switch, an application-specific integrated circuit (ASIC), a programmable logic controller, or an embedded microprocessor that stores computer readable program code (such as software or firmware) that can be executed by the microprocessor or the processor. Examples of the controller include but are not limited to the following microprocessors: ARC 625D, Atmel AT91SAM, Microchip PIC18F26K20, and Silicone Labs C8051F320. The memory controller can also be implemented as a part of the control logic of the memory. A person skilled in the art also knows that, in addition to implementing the controller by using only the computer readable program code, logic programming can be performed on method steps to allow the controller to implement the same function in forms of the logic gate, the switch, the application-specific integrated circuit, the programmable logic controller, and the embedded microcontroller. Therefore, th controller can be considered as a hardware component, and an apparatus included in the controller for implementing various functions can also be considered as a structure within the hardware component. Or, the apparatus configured to implement various functions can even be considered as both a software module implementing the method and a structure in the hardware component.

[82] The system, apparatus, module, or unit illustrated in the previous implementations can be implemented by using a computer chip or an entity, or can be implemented by using a product having a certain function. A typical implementing device is a computer. Specifically, the computer can be, for example, a personal computer, a laptop computer, a cellular phone, a camera phone, a smartphone, a personal digital assistant, a media player, a navigation device, an email device, a game console, a tablet computer, or a wearable device, or a combination of any of these devices.

[83] For ease of description, the apparatus above is described by dividing functions into various units. Certainly, when the present specification is implemented, the functions of the units can be implemented in one or more pieces of software and/or

hardware.

[84] A person skilled in the art should understand that the implementations of the present specification can be provided as methods, systems, or computer program products. Therefore, the implementations of the present specification can use a form of hardware only implementations, software only implementations, or implementations with a combination of software and hardware. Furthermore, the implementations of the present specification can use a form of a computer program product that is implemented on one or more computer-usable storage media (including but not limited to a magnetic disk memory, a CD-ROM, an optical memory, etc.) that include computer-usable program code.

[85] The present specification is described with reference to the method, the device (system), and a flowchart and/or a block diagram of the computer program product according to the implementations of the present specification. It should be understood that computer program instructions can be used to implement each process in the flowchart and/or each block in the block diagram, and a combination of a process in a flowchart and/or a block in a block diagram. These computer program instructions can be provided to a processor of a general-purpose computer, a special-purpose computer, an embedded processor or another programmable data processing device to produce a machine, so that instructions executed by a processor of a computer or another programmable data processing device produce an apparatus for implementing functions specified in one or more processes of the flowchart and/or one or more blocks of the block diagram.

[86] These computer program instructions can also be stored in a computer-readable memory that can direct a computer or another programmable data processing device to work in a particular way, so that the instructions stored in the computer-readable memory produce a manufacture that includes an instruction apparatus. The instruction apparatus implements the functions specified in one or more processes of the flowchart and/or one or more blocks of the block diagram.

[87] These computer program instructions can also be loaded onto a computer or another programmable data processing device so that a series of operational steps are performed on the computer or the another programmable device to produce processing implemented by a computer. As such, instructions executed on the computer or the another programmable device provide steps for implementing the functions specified in one or more processes in the flowchart and/or one or more

blocks in the block diagram.

[88] In a typical configuration, a computing device includes one or more processors (CPUs), an input/output interface, a network interface, and a memory.

[89] The memory might include a non-persistent memory, a random access memory (RAM), and/or a non-volatile memory in a computer-readable medium, such as a read-only memory (ROM) or a flash random access memory (flash RAM). The memory is an example of the computer-readable medium.

[90] The computer-readable medium includes persistent, non-persistent, movable, and unmovable media that can implement information storage by using any method or technology. The information can be a computer-readable instruction, a data structure, a program module, or other data. Examples of storage media for a computer include, but are not limited to, a phase change random access memory (PRAM), a static random access memory (SRAM), a dynamic random access memory (DRAM), another type of random access memory (RAM), a read-only memory (ROM), an electrically erasable programmable read-only memory (EEPROM), a flash memory or another memory technology, a compact disc read-only memory (CD-ROM), a digital versatile disk (DVD) or another optical storage, a magnetic cassette tape, a magnetic tape magnetic disk storage or another magnetic storage device or any other non-transmission medium that can be configured to store information accessible to the computing device. As described in the present specification, the computer-readable medium does not include transitory media, such as modulated data signals and carriers.

[91] It should also be noted that, the terms "include", "comprise", or their any other variant are intended to cover a non-exclusive inclusion, so that a process, a method, a product, or a device that includes a list of elements not only includes those elements but also includes other elements which are not expressly listed, or further includes elements inherent to such process, method, product, or device. An element confined by "includes a ..." does not, without more constraints, preclude the presence of additional identical elements in the process, method, product, or device that includes the element.

[92] The present specification can be described in a general context of computer executable instructions executed by a computer, such as a program module. Generally, the program module includes a routine, a program, an object, a component, a data structure, etc. executing a specific task or implementing a specific abstract data type.

It is also possible to practice the present specification in distributed computing environments where tasks are performed by remote processing devices connected through a communication network. In a distributed computing environment, the program module can be located in local and remote computer storage media including storage devices.

[93] The implementations in the present specification are described in a progressive way. For a same or similar part in the implementations, reference can be made to each other, and each implementation focuses on differences from another implementation. In particular, for a system implementation, description is relatively simple because the system implementation is basically similar to a method implementation, and reference can be made to some descriptions of the method implementation.

[94] The previous descriptions are merely implementations of the present specification and are not intended to limit the present application. For a person skilled in the art, the present application is subject to a plurality of modifications and changes. Any modification, equivalent replacement, or improvement made within the spirit and principles of the present application shall be included in the claims of the present application.

CLAIMS

What is claimed is:

1. A method for transaction risk control based on a graphical structure model, comprising:

training a graphical structure model by using labeled samples, wherein the graphical structure model is defined based on a transaction data network and the transaction data network comprises one or more nodes of a buyer user, a seller user, an e-commerce platform account, a third-party payment platform account, a bank card, and a product, and a relationship between the nodes, wherein the labeled samples are labeled whether a node corresponding to the labeled samples is a transaction risk node;

wherein the graphical structure model is configured to calculate embedding vectors of the nodes obtained after a plurality of iterations in a latent feature space based on original features of the nodes and/or features of associated edges between the nodes, and to calculate, for each node of the nodes and after the plurality of iterations in a latent feature space, a prediction probability of the node based on the embedding vector of the node, wherein the prediction probability represents a probability that the node is a transaction risk node;

wherein training the graphical structure model by using the labeled samples comprises training the graphical structure model by using the labeled samples with a training objective of maximizing, for each node corresponding to a labeled sample, consistency between the prediction probability of the node and a sample labeling result corresponding to the prediction probability;

wherein the method further comprises:

calculating (S304) an embedding vector corresponding to a to-be-tested sample by using the trained graphical structure model; and

performing (S306) transaction risk control on the to-be-tested sample based on the corresponding embedding vector, comprising:

calculating a prediction probability corresponding to the to-be-tested sample by using the trained graphical structure model based on the embedding vector corresponding to the to-be-tested sample; and

performing transaction risk control on the to-be-tested sample based on the prediction probability corresponding to the to-be-tested sample.

2. The method according to claim 1, wherein the transaction data network is built based on user transaction-related data.

3. The method according to claim 1, wherein the transaction data network comprises one or a combination of the following networks: a transaction network formed by a buyer user and a seller user, a relationship network of a buyer user and a debit card corresponding to a buyer user account, a relationship network of a buyer user and a credit card corresponding to a buyer user account, a same-person relationship network between user accounts, a family relationship network between user accounts, and a network formed by a buyer user and a product, wherein the user account comprises an e-commerce platform account and/or a third-party payment platform account.

4. The method according to claim 1, wherein the original feature comprises at least one of the following types of transaction data: a transaction amount, a quantity of transactions, a transaction time, a transaction location, a transaction target, and a transaction payment method; and

the feature of the edge comprises transaction data and/or interpersonal relationship data jointly involved by the nodes connected to the edge.

5. The method according to claim 1, wherein an embedding vector of a node of the nodes of the transaction data network obtained after t iterations in the latent feature space is calculated based on the original feature of the node, a feature of an associated edge between the node and another node, and an embedding vector of the node obtained after $t-1$ iterations in the latent feature space.

6. The method according to claim 5, wherein the node is a ith node of the nodes corresponding to the labeled samples, wherein calculating an embedding vector of the node obtained after a plurality of iterations in the latent feature space comprises calculating the embedding vector of the node based on the following equation:

$$H_i^t = \sigma\left(W_1 F_i + W_2 \sum_{j \in N_i} E_{ij} + W_3 \sum_{j \in N_i} \alpha_{ij} H_i^{t-1} \right)$$; wherein

H_i^t represents the embedding vector of the ith node obtained after t iterations

of the plurality of iterations in the latent feature space; σ represents a nonlinear transformation function; W_1, W_2, and W_3 represent weight matrices; F_i represents original features of the ith node; E_{ij} represents features of associated edges between the ith node and the jth node; N_i represents a set of adjacent nodes of the ith node; and α_{ij} represents a weight matrix for the ith node and the jth node.

7. The method according to claim 6, wherein the plurality of iterations comprises *T* iterations, wherein the calculating a prediction probability of the node based on the embedding vector specifically comprises:

calculating the prediction probability of the node based on the following equation:

$pred_i = softmax\left(W_4 * H_i^T\right)$; wherein

$pred_i$ represents a prediction probability of the ith node obtained after T iterations, and W_4 represents a weight matrix.

8. The method according to claim 7, wherein training the graphical structure model by using the labeled samples specifically comprises:

optimizing $\arg\min_{w_1,w_2,w_3,w_4} \sum_i corss_entrep\left(pred_i, y_i\right)$ by using a back-propagation algorithm and the labeled samples to obtain optimal W_1, W_2, W_3, and W_4; wherein

$corss_entrep$ represents a function used to calculate cross entropy.

9. A device for transaction risk control based on a graphical structure model, comprising:

at least one processor; and

at least one memory connects to and communicates with the at least one processor, wherein

the memory stores an instruction (401, 402, 403, 501, 502, 503) executable by the at least one processor, and the instruction (401, 402, 403, 501, 502, 503) is executed by the at least one processor to enable the at least one processor to perform a method according to any one of the preceding claims.

10. One or more computer-readable media comprising instructions (401, 402, 403, 501, 502, 503) stored thereon which, when executed by a computing device, cause the computing device to perform a method according to any of the claims 1 to 8.