

# ENTITY TRANSACTION ATTRIBUTE DETERMINATION METHOD AND APPARATUS

## TECHNICAL FIELD

One or more embodiments of this specification relate to the field of computer technologies, and in particular, to an entity transaction attribute determination method and apparatus through computers.

## BACKGROUND

Along with the development of computer technologies, assistance from computers is becoming increasingly indispensable for daily transaction processing. For example, on a shopping platform, computers record the products browsed, clicked, and purchased by users to recommend products that may be more appealing to the users. In news APPs on terminals, the terminals record the pages searched for and browsed by users, thereby recommending more suitable news articles for users. Various similar examples exist but are not elaborated further herein. Particularly, the development of artificial intelligence technology has enabled computers to handle daily transactions more intelligently. For example, a shopping platform predicts a certain transaction attribute (such as gender) of a user to pitch to the user products that are suitable for the corresponding transaction attribute.

In exemplary applications, some transaction attributes of an entity being processed may be fixed, such as a gender transaction attribute of the user, while other transaction attributes may change under unexpected events. For example, an income transaction attribute of a user may change in accordance with events such as a job change, losing jobs, health status, etc. A finance-related overdue risk transaction attribute of a user will be subjected to a greater change upon events such as a change in occupation, a change of the family members of the user, and a switch of a loan platform of the user from a bank to a small loan company. Various similar examples exist but are not elaborated further herein. Therefore, a solution is needed to integrate the influence of such unexpected events into an artificial intelligence-based prediction of entity transaction attributes, thereby improving the prediction accuracy.

## SUMMARY

One or more embodiments of this specification describe an entity transaction attribute determination method and apparatus, thereby solving one or more technical problems mentioned in the Background section.

According to a first aspect, an entity transaction attribute determination method is

provided for determining an attribute state or an attribute category of a to-be-predicted entity on a preset transaction attribute. The method comprises: obtaining a plurality of historical relational networks sequentially arranged in accordance with a temporal order, wherein the historical relational networks respectively correspond to historical time points having a preset time interval, each of the historical relational networks comprises a plurality of nodes, each of the nodes corresponds to a plurality of entities comprising the to-be-predicted entity, each of the historical relational networks is configured to describe association relationships between a plurality of entities which correspond to a corresponding historical time point; the entities having an association relationship are connected by a connecting edge therebetween, and each of the nodes corresponds to a node vector composed of transaction characteristics extracted based on description information of the corresponding entities; determining, for each of the historical relational networks and through vector fusion of neighbor nodes, a plurality of expression vectors of the to-be-predicted entity; processing, through a pre-trained time-series neural network, the expression vectors to obtain an output result; and determining, according to the output result, the attribute state or the attribute category of the to-be-predicted entity on the preset transaction attribute.

According to an embodiment, the time-series neural network is trained in the following manner: obtaining a current training sample corresponding to a sample entity and an attribute tag of the sample entity with respect to the preset transaction attribute; executing, for each historical relational network of T historical relational networks sequentially arranged in accordance with the temporal order, a node vector fusion operation of the neighbor nodes, to determine T expression vectors respectively corresponding to the sample entity under the T relational networks, the T historical relational networks being established in accordance with T time points having the preset time interval; sequentially inputting, in accordance with the temporal order of the corresponding historical relational networks, the T expression vectors into a selected time-series neural network, and obtaining a sample output result of the time-series neural network for the sample entity; and adjusting, with an objective of matching the sample output result with the attribute tag, model parameters to train the time-series neural network.

According to an embodiment, the transaction characteristics in the node vector corresponding to a single node are extracted through description information at a corresponding historical time point of a corresponding relational network, or extracted through description information within a time interval between the corresponding historical time point and a previous historical time point of the corresponding relational network.

According to an embodiment, in the case that the to-be-predicted entity is a user, the entities in each of the historical relational networks comprise a plurality of users, and further comprise at least one of the following: an IP identifier or a WIFI identifier for a user device to access a network, and an application installed on the user device.

5           According to an embodiment, the historical relational networks comprise a first relational network; the to-be-predicted entity corresponds to a first node in the first relational network, and the determining, for each of the historical relational networks and through vector fusion of the neighbor nodes, a plurality of expression vectors of the to-be-predicted entity comprises: processing the relational network through a multilayer graph neural  
10   network, the multilayer graph neural network updating, after processing of a current layer is done, the node vector of the first node based on the following result: a weighted result from a product of the previous layer-updated node vector of each neighbor node and an auxiliary weight matrix; and using the node vector of the first node that has been updated by the last layer of the graph neural network as a first expression vector of the first node corresponding  
15   to the first relational network.

          According to a further embodiment, the neighbor nodes of the first node comprise a second node, and at the current layer, a weight of the second node relative to the first node is: an exponential form-based normalization result of a similarity level between the previous layer-updated node vector of the first node and the previous layer-updated node vector of the  
20   second node relative to a sum of similarity levels between the previous layer-updated node vector of the first node and the previous layer-updated node vector of each of the neighbor nodes.

          According to a yet further embodiment, the similarity level between the previous layer-updated node vector of the first node and the previous layer-updated node vector of the  
25   second node is determined in the following manner: concatenating the previous layer-updated node vector of the first node and the previous layer-updated node vector of the second node to form a concatenation vector; performing, through a spatial auxiliary weight matrix, a dimensionality reduction process on the concatenation vector to obtain an intermediate vector having preset dimensions; and processing, through an auxiliary vector having the preset  
30   dimensions, the intermediate vector to obtain the similarity level between the previous layer-updated node vector of the first node and the previous layer-updated node vector of the second node.

          According to an embodiment, in the case that the output result is an output vector, the determining, according to the output result, the attribute state or the attribute category of the

to-be-predicted entity on the preset transaction attribute comprises: processing the output vector through a fully connected neural network; and determining, through an obtained processing result, the attribute state or the attribute category of the to-be-predicted entity on the preset transaction attribute.

5           According to an embodiment, in the case that the output result is an output vector, the attribute states or attribute categories of the to-be-predicted entity on the preset transaction attribute further respectively correspond to description vectors, and the determining, according to the output result, the attribute state or the attribute category of the to-be-predicted entity on the preset transaction attribute comprises: determining respective  
10   corresponding similarity levels between the output vector and each of the description vectors; and determining an attribute state or attribute category corresponding to a description vector having the highest similarity level as the attribute state or attribute category of the to-be-predicted entity on the preset transaction attribute during a preset time period.

          According to a second aspect, an entity transaction attribute determination apparatus  
15   is provided for determining, through a plurality of historical relational networks based on a temporal order, an attribute state or an attribute category of a to-be-predicted entity on a preset transaction attribute. The apparatus comprises: a determining unit, configured to obtain a plurality of historical relational networks sequentially arranged in accordance with a temporal order, wherein the historical relational networks respectively correspond to  
20   historical time points having a preset time interval; each of the relational networks comprises a plurality of nodes, each of the nodes corresponds to a plurality of entities comprising the to-be-predicted entity, each of the historical relational networks is configured to describe association relationships between a plurality of entities which correspond to a corresponding historical time point, the entities having an association relationship are connected by a  
25   connecting edge therebetween, and each of the nodes corresponds to a node vector composed of transaction characteristics extracted based on description information of the corresponding entities; a fusion unit, configured to determine, for each of the relational networks and through vector fusion of the neighbor nodes, a plurality of expression vectors of the to-be-predicted entity; a time-series data processing unit, configured to process, through a pre-  
30   trained time-series neural network, the expression vectors to obtain an output result; and a prediction unit, configured to determine, according to the output result, the attribute state or the attribute category of the to-be-predicted entity on the preset transaction attribute.

          According to a third aspect, a computer-readable storage medium is provided and has a computer program stored thereon; and when the computer program is executed in a

computer, the computer is caused to execute the method of the first aspect.

According to a fourth aspect, a computing device is provided and comprises a memory and a processor, wherein executable code is stored in the memory; and when executing the executable code, the processor implements the method of the first aspect.

5 Through the method and apparatus provided in the embodiments of this specification, on the one hand, expression vectors of a to-be-predicted entity at different historical time points are processed through a time-series neural network to take into account time-series characteristics of the to-be-predicted entity, and on the other hand, node vector aggregation of the neighbor nodes is performed for the expression vector of a single historical time point  
10 based on a relational network determined in accordance with entity states of a corresponding historical time point, fully exploring the influence of a neighbor entity on the current entity state, thereby generating an expression vector having a better expression capability for users. Further, the accuracy of predicting entity transaction attributes is improved through processing the historical relational network having time-series characteristics.

## 15 **BRIEF DESCRIPTION OF THE DRAWINGS**

To describe the technical solutions in the embodiments of this specification more clearly, the following briefly describes the accompanying drawings required for describing the embodiments. Apparently, the accompanying drawings in the following description are some embodiments of this specification, and a person of ordinary skill in the art may further  
20 derive other accompanying drawings from these accompanying drawings without inventive efforts.

FIG. 1 shows a schematic diagram of an implementation architecture of this specification.

FIG. 2 shows a schematic diagram of a relational network according to some  
25 embodiments.

FIG. 3 shows a flowchart of a method for training a time-series neural network according to some embodiments.

FIG. 4 shows a flowchart of an entity transaction attribute determination method according to some embodiments.

30 FIG. 5 shows a schematic block diagram of an entity transaction attribute determination apparatus according to some embodiments.

## DETAILED DESCRIPTION

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

First, a description will be provided in conjunction with the implementation architecture shown in FIG. 1. As shown in FIG. 1, under the implementation architecture of this specification, a relational network may be pre-established with multiple time points as reference points. Nodes in the relational network may be various entities associated with an entity having a to-be-predicted transaction attribute. For example, if the entity having a to-be-predicted transaction attribute is a user, the entities in the relational network may be multiple users and entities related to user devices. It is to be understood that a user herein may generally refer to a virtual user (different from a biological person), for example, a terminal user, or a registered user of a network platform (website or APP). Generally, one user ID may correspond to one user. Such a user typically interacts with a server through a user device, which means that the user often relies on the user device to interact with the server. The entity related to the user device may reflect the characteristics of the user. That is, the entity is also associated with the user.

Similarly, in the case that the entity having a to-be-predicted transaction attribute is another type of entity (e.g., a product, a customer service question and answer text, etc.), the characteristics of the entity having a to-be-predicted transaction attribute may also be described by other related entities (such as a customer, a merchant, a customer raising a customer service question, etc.) according to the exemplary application scenario. Therefore, under the technical concept of this specification, when a relational network is to be established, nodes in one-to-one correspondence with all entities comprising the entity having a to-be-predicted transaction attribute are provided, and nodes in one-to-one correspondence with other associated entities may be further provided. The association relationships between the entities corresponding to the nodes may be represented by connecting lines between the nodes. In the case where the entity having a to-be-predicted transaction attribute is a user, other associated entities may include, but are not limited to, for example, an application installed on a user device, an IP identifier, WiFi identifier, or communication base station identifier for a user device to access the Internet, an activity-related organization (for example, a financial activity-related organization is a bank or a small high-interest loan company, etc.), and so on.

For each relational network, transaction characteristics of each entity may be extracted according to relevant description information of the entity to express the

corresponding entity. In some embodiments, corresponding transaction characteristics of users may be extracted according to their gender, income, employer, consumer preferences (or capital flow), and other description information to form a node vector. Corresponding characteristics for the IP identifier, WiFi identifier, or communication base station identifier for a user device to access the Internet may be extracted according to the geographic location information, the number of access users, and other description information of the identifier to form a node vector. Corresponding transaction characteristics of an application installed on a user device may be extracted according to the download volume, release time, and application category (such as shopping, gaming, etc.), and other description information to form a node vector. Corresponding transaction characteristics of an activity-related organization may be extracted according to the establishment time, user volume, asset category, scaled total asset number, organization category, and other description information to form a node vector. Various similar examples exist but are not elaborated further herein.

In some embodiments, uniform dimensions may be configured for the vectors of various nodes. For example, nodes corresponding to entities in different categories respectively correspond to characteristics having the corresponding dimensions. The number of dimensions of each node vector is a sum of the dimensions of each of the entity categories, and the node vector takes values in the corresponding dimensions. In one example, the number of entity categories is, for example, 3, including the user, the APP, and the activity-related organization. Assuming that the transaction characteristics of the user is 5-dimensional, the transaction feature of the APP is 4-dimensional, and the transaction characteristics of the activity-related organizations is 3-dimensional, then the total number of dimensions for the transaction characteristics of the entity categories is 12. In this case, each node may be represented by a 12-dimensional node vector. A single node has meaningful characteristic values only in the corresponding dimensions. For example, a user has characteristic values in the first 5 dimensions of the 12-dimensional node vector, and the last 7 dimensions all take a preset value (such as 0), and so on.

Under the concept of this specification, each expression vector of an entity requiring transaction attribute prediction may be determined by fusing characteristics of neighbor nodes in each relational network. Then, as shown in FIG. 1, expression vectors arranged in a temporal order are sequentially inputted into the time-series neural network to obtain an output result. The output result may correspond to a preset transaction attribute of a user.

The technical concept of this specification is described in detail below.

It can be understood that under the technical concept of this specification, the

involved technical problems may include the construction of a relational network, the vector fusion of a neighbor node in a single relational network, and the processing of each expression vector by the time-series neural network. The following describes these technical problems and each of the implementation solutions to these problems.

5 First, the relational network involved in this specification may be constructed by collecting information of corresponding entities under time points having a preset time interval. The aforementioned time points may be regarded as the time basis for data truncation. The preset time interval may be, for example, one month or one week, which is not limited in this specification. The exemplary collected entity information may be various  
10 pieces of information at the current time point corresponding to the relational network, or may be entity information during the time interval between the current time point and the previous time point. In some embodiments, for a physical APP related to a user device, a user download characteristic may be the total user downloads up to the corresponding time point, or may be the user downloads during the time interval between the corresponding time point  
15 and the previous time point. For a user, an occupation characteristic may be the occupation of the user at the corresponding time point, or various occupations which the user has taken up during a preset time interval before the corresponding time point. Various similar examples exist but are not elaborated further herein.

In this specification, a corresponding relational network may be constructed through  
20 various entities related to the entity having a to-be-predicted transaction attribute. The nodes in the relational network may correspond to the various entities in a one-to-one manner. A user may have an association relationship with entities such as other users, the APPs installed on the user device into which the current user is logged, the WIFI address or IP address for accessing the Internet, and the like. The association relationship between entities may be  
25 represented by a connecting edge or may be represented by a preset identifier in a three-tuple.

In some embodiments, the association relationship between a user entity and another user entity may be the one determined by activities such as being a contact in the address book, performing a wire transfer, sending a red envelope, etc. When such an association relationship exists between two users, the two corresponding entities in the relational network  
30 may be connected by a connecting edge.

As shown in FIG. 2, an entity with a to-be-predicted transaction attribute being a user is taken as an example, and a schematic diagram of a relational network corresponding to a time point is shown. In FIG. 2, the involved entities include, but are not limited to: users, APPs associated with a user device, WIFI networks for a user device to access the Internet,

and so on. As shown in FIG. 2, user A has association relationships with user C, APP1, and WIFI1; user B has association relationships with APP1, APP2, and APP3; user C has association relationships with APP2, user E, user A, and WIFI2; user E has association relationships with user C, APP3, and WIFI2; and user D has association relationships with APP1, APP2, WIFI1, and WIFI2. The relational network in FIG. 2 may describe the association relationships between the users and other entities. As an example, other associated entities in FIG. 2 include APPs and WIFI networks. In practice, other associated entities may be various entities related to the to-be-predicted entity.

As mentioned above, each node may have corresponding description information. The corresponding transaction characteristics may be extracted from the description information to form a characteristic vector, which is used as an initial node vector describing the corresponding entity.

In this way, for each of the time points with a preset interval, a corresponding relational network may be constructed to describe the association relationships between various entities. Each of the relational networks may have node vectors determined according to the current description information of the corresponding entities. In this way, states of the entities at each of the time points arranged in accordance with a time interval may be characterized, thereby describing, in a temporal order, the states of the entity having a to-be-predicted transaction attribute at different times.

Secondly, the vector fusion may be performed for vectors of neighbor nodes in an individual relational network. It is to be understood that a node may have various associations with neighbor nodes, and such associations may reflect implicit properties of the entity. In some embodiments, a user having an association relationship with a game APP may demonstrate the user's current interest in the game APP. Users who are interested in the game APP may share some similar traits, such as being a student, being unemployed, having a stress-free occupation, and so on. The vector fusion of neighbor nodes is to fuse the information of the neighbor nodes for each node, thereby enhancing the expression capability of a node vector for a node.

In some embodiments, the node vectors of neighbor nodes within a preset order with respect to the current node may be fused together through methods such as a weighted average, a maximum value, etc., to obtain the neighbor fusion result, which may be further fused with the node vector of the current node. The obtained fusion vector is used as the expression vector of the current node. In practice, other methods may be used for the neighbor node fusion.

In some embodiments, the current node may be considered as its own neighbor node to perform the neighbor-node vector fusion.

In some embodiments, some nodes may be selected from neighbor nodes to perform the neighbor-node vector fusion.

5 In some embodiments, node vectors of neighbor nodes within a preset order with respect to the current node may be fused layer-by-layer in an iterative manner from the outermost layer to the innermost layer. In some embodiments, the preset order may be 3, which means that the neighbor-node vector fusion may first be performed on the third-order neighbor nodes of the current node, and the node vectors of the third-order neighbor nodes  
10 after the fusion are used to perform neighbor-node vector fusion on the second-order neighbor nodes. Then, neighbor-node vector fusion is performed on the first-order neighbor nodes. Finally, the node vectors of the first-order neighbor nodes with the neighbor-node vector fusion performed is fused with the current node.

15 In some embodiments, all nodes in the relational network may be traversed to perform node vector fusion of neighbor nodes on each of the nodes, causing the current node vector of each node to be updated. After multiple iterations, the current node vector of the current node is used as the expression vector of the current node in the single relational network.

Various similar examples exist but are not elaborated further herein.

20 In some embodiments, an individual relational network may be processed through a graph neural network (GNN), causing the node vector fusion of neighbor nodes to be performed on each node. It is to be understood that for the graph neural network, each neural network layer means traversing each node in the relational network, and updating the current node vector of each node as the fusion vector for the vectors of the neighbor nodes of the current node.

25 As an example, when the relational network is processed through a graph neural network, the  $(l+1)$ -th layer of the neural network may update the node vector of node  $u$  to be a function of a weighted result based on setting the weight of each neighbor node as an attention value:

$$h_u^{l+1} = f\left(\sum_{v \in N_u} \alpha(u, v) h_v^l w^l\right)$$

30 where  $h_u^{l+1}$  represents the node vector of the node  $u$  after being updated by the  $(l+1)$ -th layer,  $\alpha(u, v)$  represents the importance coefficient of the node  $v$  for the node  $u$  in the  $(l+1)$ -th layer of the graph neural network,  $h_v^l$  represents the node vector of the node  $u$  after being updated

by the  $l$ -th layer,  $N_u$  represents a set of all or some (e.g., no more than 5) of the neighbor nodes of node  $u$  within a preset order (e.g., 1). Generally, the neighbor nodes may include node  $u$  itself, and  $W^l$  represents the auxiliary weight matrix in the  $(l+1)$ -th layer of the graph neural network and may be used to change the dimensions of the node vector. It is to be understood that  $W^l$  is a model parameter that may be determined by training the training samples.

In an individual relational network, considering that different neighbor nodes have different degrees of influence on the current node, this specification uses the importance coefficient to distinguish the importance of each neighbor node for node  $u$ . In some embodiments, the node  $u$  has the greatest influence on itself and may have a large importance coefficient. The importance coefficient of each neighbor node may be, for example, preset, or may also be the similarity level between the expression vector of the neighbor node and that of the current node, and so on. In an example, the importance coefficient of each neighbor node may be an attention value determined according to the similarity level between each of the neighbor node and the current node.

In some embodiments, the similarity between node  $v$  and node  $u$  is represented by a similarity level. The importance coefficient of node  $v$  relative to node  $u$  ( $\alpha(u, v)$ ) may be an exponential form-based normalization result of the similarity level between the node vector of node  $v$  and the node vector of node  $u$  relative to a sum of the similarity levels between the node vector of each neighbor node of node  $u$  and the node vector of node  $u$ , e.g.:

$$\alpha(u, v) = \frac{\exp(s(u, v))}{\sum_{j \in N_u} \exp(s(u, j))}$$

where  $s(u, v)$  represents the similarity level value between node  $u$  and node  $v$ . In one embodiment, the similarity level value may be represented by a dot product of the respective node vectors corresponding to node  $u$  and node  $v$ . In another embodiment, the similarity level value may be determined by:

$$s(u, v) = \mathbf{a}^T \cdot \mathbf{W}_{spatial} \cdot (\mathbf{h}_u || \mathbf{h}_v)$$

where “||” represents the concatenation of two vectors,  $\mathbf{W}_{spatial}$  represents an auxiliary weight matrix for the vector space and is used to reduce the dimensions of a vector or for mapping to the corresponding characteristic space having preset dimensions. A similarity level value may be obtained through processing the auxiliary vector  $\mathbf{a}^T$  by reducing the vector dimensions or mapping to the vector of the corresponding characteristic space, and

$W_{spatial}$  and  $\mathbf{a}^T$  are model parameters that may be determined by training the training samples.

In this way, the transaction characteristics described by the node vectors in the multi-order neighborhood may be fused into the current node (node  $u$ ) through the multilayer graph neural network, thereby obtaining an expression vector having a better expression capability.

Further, for the relational network corresponding to each time point, one expression vector of the entity having a to-be-predicted transaction attribute may be obtained. Since the time points have a temporal order, respective expression vectors may also have a corresponding temporal order.

Finally, each expression vector may be processed by the time-series neural network. It is to be understood that the time-series neural network may be a neural network for processing various data having a temporal order, such as a recurrent neural network (RNN).

An RNN is a time-recursive neural network that may process sequential data. In RNNs, a current output of a sequence is related to the previous output. For example, an RNN memorizes and applies the previous information in the calculation of the current output, i.e., the nodes being connected between the hidden layers. The input for the hidden layers includes not only the output of the input layer but also the output of the hidden layers at the previous time instance. In the time-series diagram of the recurrent neural network as shown in FIG. 3, the hidden layer state at the  $t$ -th iteration may be expressed as:

$$\mathbf{S}_t = f(\mathbf{U} * \mathbf{X}_t + \mathbf{W} * \mathbf{S}_{t-1})$$

$\mathbf{X}_t$  is the state of the input layer at the  $t$ -th iteration,  $\mathbf{S}_{t-1}$  is the state of the hidden layer at the  $(t-1)$ -th iteration,  $f$  is the calculation function, and  $\mathbf{W}$  and  $\mathbf{U}$  are the weights. In this way, the RNN loops the previous state back to the current input to take into account the influence of the historical input and is therefore suitable for a time-series data sequence.

In a further embodiment, as shown in FIG. 1, the time-series neural network may be, for example, a Long Short-Term Memory (LSTM). For a neuron in an LSTM, at time instances of  $t-1$ ,  $t$ , and  $t+1$ , the expression vectors, arranged in accordance with an order, of the entity having a to-be-predicted transaction attribute in the corresponding relational network of the corresponding time period are separately inputted, and are respectively represented using  $X_{t-1}$ ,  $X_t$ , and  $X_{t+1}$ .  $S_{t-1}$ ,  $S_t$ , and  $S_{t+1}$  respectively represent the states of the neuron at time instances of  $t-1$ ,  $t$ , and  $t+1$ .  $C_{t-1}$ ,  $C_t$ ,  $C_{t+1}$  respectively represent the output at time instances of  $t-1$ ,  $t$ , and  $t+1$ , wherein:

$$S_t = g(\mathbf{U} * \mathbf{X}_t + \mathbf{W} * \mathbf{C}_{t-1} + b_s)$$

$$C_t = f(V * S_{t-1} + b_c)$$

$$S_{t+1} = g(U * X_t + W * C_t + b_s)$$

$$C_{t+1} = f(V * S_t + b_c)$$

where  $U$ ,  $W$ , and  $V$  are weights.

It can be seen that in the LSTM model, the current state of each neuron is jointly determined by the input at the current time instance and the output at the previous time instance. The current output of each neuron is related to the state at the previous time instance. The expression vectors of the entity at various time points are analyzed using the LSTM model to selectively memorize information and perform data mining for long-range dependence.

The main technical problems and implementation solutions involved under the technical concept of this specification are described in detail above. The complete technical solutions will be described below in conjunction with some embodiments.

FIG. 3 shows the process of training a time-series neural network according to some embodiments of this specification. The main entity executing this process may be a computer, device, or server having certain computing capabilities. It is to be understood that the model parameters in the time-series neural network may be adjusted by using multiple training samples to train the time-series neural network. As shown in FIG. 3, the process of training the time-series neural network comprises the following steps: step 301, obtaining a current training sample corresponding to a sample entity and an attribute tag of the sample entity with respect to the preset transaction attribute; step 302, executing, for each historical relational network of  $T$  historical relational networks sequentially arranged in accordance with the temporal order, a node vector fusion operation of a neighbor node, so as to determine  $T$  expression vectors corresponding to the sample entity under the  $T$  historical relational networks, the  $T$  historical relational networks being established in accordance with  $T$  time points having the preset time interval; step 303, sequentially inputting, in accordance with the temporal order of the corresponding historical relational networks, the  $T$  expression vectors into a selected time-series neural network, and obtaining an output sample result of the time-series neural network for a current sample entity; and step 304, adjusting, with an objective of matching the output sample result with the attribute tag, model parameters to train the time-series neural network.

First, in step 301, the current training sample, corresponding to the sample entity and

the attribute tag of the sample entity with respect to the preset transaction attribute, is obtained. It is to be understood that in a supervised machine learning model training process, training samples may be divided into multiple batches, and one model parameter adjustment operation may be performed for each batch. One batch of training samples may be one  
5 training sample or multiple training samples, which is not limited by the embodiments herein. The current training sample is the training sample of the current batch. The term “current” in the current batch corresponds to the current process. In practice, the current batch may be any batch.

One training sample may correspond to one piece of entity data. In the case that the  
10 sample entity is a sample user, one piece of entity data may indicate one sample user and a historical attribute tag of the sample user on the preset transaction attribute. The term “historical attribute tag” is used herein because, for the training sample, the corresponding attribute state or attribute category of the sample entity on the preset transaction attribute is already available.

15 Then, in step 302, for each historical relational network in the T historical relational networks arranged in accordance with temporal order, the node vector fusion operation of a neighbor node is performed to determine the corresponding T expression vectors of the sample entity under the T historical relational networks. T is an integer greater than 1.

It is to be understood that each historical relational network describes the  
20 instantaneous state of each entity at the corresponding historical time point. In order to predict, from entity states corresponding to the T historical time points, the attribute state (e.g., a scaled income number) or attribute category (e.g., the gender attribute category being male, etc.) of the sample entity on the preset transaction attribute, mining previous state information before the corresponding attribute state or attribute category on the preset  
25 transaction attribute is acquired by the user is required. Therefore, the T historical relational networks of the sample entity are established at the T historical time points prior to the generation of the attribute tag. Since the relational network closer to the time that the sample entity generates the attribute tag can reflect the potential attribute state or attribute category of the sample entity better, in some embodiments, the historical time point corresponding to the  
30 T-th historical relational network thus does not differ from the generation time of the attribute tag of the sample entity by a preset time threshold.

In one embodiment, the sample entity and the attribute tag of the sample entity on the preset transaction attribute may be determined first, then the T historical relational networks may be established per a preset time interval by counting backward. For example, the time

instance corresponding to the attribute tag of the sample entity is June 1, 2020, the preset time threshold is 1 week, and the preset time interval is 1 month. Then, a time point within the period of one week before June 1, 2020 is selected as the T-th historical time point, e.g., May 31, 2020. The (T-1)-th historical time points are selected by counting backward using the preset time interval, e.g., selecting April 30, 2020, March 31, 2020, and the like, which are (T-1) time points in total. Corresponding relational networks are respectively established for the T time points, including the T-th time point and the T-1 time points. The manner of establishing a relational network is as described above and will not be further elaborated herein.

In some embodiments, the relational network at each time point may also be pre-established in accordance with a time interval. For example, the 15th of every month is taken as a time point for establishing corresponding relational networks under a time interval of 1 month. After the time instance corresponding to the attribute tag of the sample user is determined, T corresponding historical relational networks are selected under the order of the corresponding historical time points from the closest to the most distant in time.

Further, for each of the T historical relational networks, one expression vector may be determined under the manner described above, which yields a total of T expression vectors. It can be understood that since the T historical relational networks are arranged under the temporal order, these T expression vectors also have time-series characteristics. In the case that there are multiple current sample entities, the attribute tags corresponding to these multiple sample entities may have different time instances, which means that each of the sample entities may correspond to its own attribute tag time instance. Similarly, each sample entity may correspond to its own T historical relational networks and T expression vectors.

It is worth noting that for these T historical relational networks, one graph neural network may process the relational networks, or T graph neural networks may respectively process the relational networks, which is not limited by the embodiments herein. The process of processing a relational network with a graph neural network has been described above and will not be further elaborated herein.

Then, in step 303, in accordance with the temporal order of the corresponding historical relational networks, the T expression vectors is sequentially inputted into a selected time-series neural network to obtain a sample output result of the time-series neural network for the sample entity.

It is to be understood that the time-series neural network can process data having time-series characteristics. For one exemplary training sample, the to-be-processed data is T

expression vectors arranged under the time series. The time-series neural network may include one or more neurons, and the one or more neurons may respectively receive the T expression vectors at T different time instances. One output result may be obtained through the processing manner described above. The output result referred herein corresponds to the above-described sample entity, and therefore may be referred to as the sample output result.

The sample output result may be an output result obtained after the above-described one or more neurons having processed the T-th expression vector or may be a processing result obtained by combining T output results, which is not limited by the embodiments herein. The output result may be a vector, a score, and the like. The sample output result herein may indicate a prediction result of the time-series neural network for the attribute state or attribute category of the sample entity on the preset transaction attribute. It is to be understood that in the case that a batch of training samples includes multiple sample entities, a prediction result of the attribute state or attribute category on the preset transaction attribute may be obtained for each of the sample entities.

Then, in step 304, with an objective of matching the output sample result with the attribute tag, model parameters are adjusted to train the time-series neural network. It is to be appreciated that according to the form of the output sample result and the attribute tag, the consistency between these two may be described by a vector or a numerical value.

In some embodiments, in the case that the sample output result is a vector, the similarity level between the vector corresponding to the corresponding attribute tag and the vector of the sample output result (e.g., a similarity level expressed by the dot product of the vectors) may be used to describe the consistency between the sample output result and the sample attribute tag.

In some other embodiments, in the case that the sample output result is a score, the similarity level between these two may be described as the absolute difference between the score value corresponding to the sample attribute tag and the score of the sample output result. Generally, a smaller absolute difference indicates a greater similarity level between the sample attribute tag and the sample output result.

It is to be appreciated that if the sample output result is inconsistent with the attribute tag, then a difference between them may be defined as a loss, which may be described by a model parameter. The time-series neural network may be trained by adjusting the model parameters to reduce the loss.

In some embodiments, the sample output result is one score. Sequential adjustment operations of the model parameters are performed based on a batch of training samples

including multiple sample entities, and the loss may be determined in the form of cross-entropy. In some embodiments, take the binary classification result (which may correspond to a standard value of 0 or 1) as an example, the loss  $L$  may be determined in the following manner:

$$\mathcal{L} = -\frac{1}{|V|} \sum_{u \in V} y_u \log(h_u) + (1 - y_u) \log(1 - h_u)$$

where  $V$  represents the sample entity sets respectively corresponding to the training samples of the current batch,  $h_u$  represents the output score corresponding to node  $u$  (or sample entity  $u$ ), and  $y_u$  represents the attribute tag corresponding to node  $u$  (such as sample user  $u$ ). It can be understood that in the case that the binary classification result is represented by 0 or 1, one of  $y_u$  or  $1 - y_u$  is 0. Using  $y_u$  being 0 as an example, then  $y_u \log(h_u)$  is 0, and  $1 - y_u$  is 1. Therefore, the closer  $h_u$  to 1, the closer  $\log(1 - h_u)$  to 0, and the smaller the loss. The closer  $h_u$  to 0, the closer  $\log(1 - h_u)$  to negative infinity, and the greater the loss. The loss is negatively correlated with this term. The same reasoning may be applied to the case of  $y_u$  being 1, which will not be further elaborated herein.

In some embodiments, in step 302, the parameters involved in the node vector fusion operation of the neighbor node executed for each historical relational network in the  $T$  historical relational networks also need to be adjusted. Therefore, the node fusion operation may be combined with the time-series neural network to form an integrated machine learning model, and the model parameters are adjusted according to the above-described loss.

In another aspect, FIG. 4 shows a process of determining an entity transaction attribute according to some embodiments of this specification. This process is used for determining, through a plurality of historical relational networks based on a temporal order, an attribute state or an attribute category of a to-be-predicted entity on a preset transaction attribute. The main entity executing this process may be a computer, device, or server having certain computing capabilities. It can be understood that the preset transaction attribute of the to-be-predicted entity has a temporal characteristic (i.e., time-sensitive). In other words, the attribute state or attribute category of the preset transaction attribute is valid for a certain time point or time period. In some embodiments, an income transaction attribute of a user in a performance-based salary system may only be valid for a certain month. A loan overdue risk transaction attribute of a user in the credit financing field may also be only valid for the current month, or even be valid only up to the payment due date.

As shown in FIG. 4, the process of determining a transaction attribute of a to-be-

predicted entity may comprise: step 401, obtaining a plurality of historical relational networks sequentially arranged in accordance with a temporal order, wherein the historical relational networks respectively correspond to historical time points having a preset time interval. Each of the historical relational networks comprises a plurality of nodes, each of the nodes corresponds to a plurality of entities comprising the to-be-predicted entity. Each of the historical relational networks is configured to describe association relationships between various entities which correspond to a corresponding historical time point. Entities having an association relationship are connected by a connecting edge therebetween, and each of the nodes corresponds to a node vector composed of transaction characteristics extracted based on description information of the corresponding entities; step 402, determining, for each of the historical relational networks and through vector fusion of the neighbor nodes, each expression vector of the to-be-predicted entity; step 403, processing, through a pre-trained time-series neural network, each expression vector to obtain an output result; and step 404, determining, according to the output result, the attribute state or the attribute category of the to-be-predicted entity on the preset transaction attribute.

First, in step 401, the multiple historical relational networks sequentially arranged in accordance with the temporal order is obtained. Each of the historical relational networks includes multiple nodes. Each of the nodes corresponds to multiple entities. These entities include the to-be-predicted entity. These entities may belong to the same category as the to-be-predicted entity or related categories. In some embodiments, in the case that the to-be-predicted entity is a user, the multiple entities in an individual relational network may include multiple users, and may further include multiple other entities associated with the user, such as an APP installed on a user device, an IP address for a user device to access the network, etc. An individual historical relational network may be configured to describe the association relationships between various entities at the corresponding historical time point, the entities having an association relationship are connected by a connecting edge therebetween. Each node corresponds to a node vector composed of transaction characteristics extracted based on the corresponding description information. The construction process of the relational network has been described above, which will not be further elaborated herein.

It is worth noting that because the preset transaction attribute of the to-be-predicted entity has a temporal characteristic, the multiple historical relational networks herein may be associated with the temporal characteristic of the to-be-predicted transaction attribute. In other words, the multiple historical relational networks herein may be relational networks respectively corresponding to multiple historical time points that are close to the current time.

Optionally, these historical time points may be selected in accordance with a preset time interval. In order to ensure validity, the time difference between the last historical time point and the current time may not exceed a preset time threshold.

It is worth noting that the multiple historical relational networks may be pre-established under the preset time interval, and be selected according to the temporal characteristic when the process in FIG. 4 is executed, or be established by selecting multiple historical time points at the instant when the process in FIG. 4 is executed. The number of historical relational networks herein may be preset, for example, as T or other values in the process in FIG. 3. The process of determining the multiple historical relational networks is similar to that described in step 302 and will not be further elaborated herein.

Then, in step 402, for each of the historical relational networks and through vector fusion of a neighbor node, each expression vector of the to-be-predicted entity is determined. The vector fusion of the neighbor nodes was described above and will not be further elaborated herein.

Then, in step 403, through a pre-trained time-series neural network, each expression vector is processed to obtain an output result. In step 403, a time-series neural network having the model parameter(s) adjusted by the process in FIG. 3 may sequentially process each expression vector under the temporal order of each historical time point corresponding to each historical relational network, thereby obtaining the output result of the time-series neural network. The output result may be an output after the neurons of the time-series neural network have processed the last expression vector, or an outputted fusion result after each expression vector has been processed, which is not limited by the embodiments herein.

In addition, the output result of the time-series neural network may be a vector or a score. In one embodiment, a binary classification task outputs a score to indicate the probability of being classified into a certain category. In another embodiment, each category is represented by a vector, and the similarity level between the output vector and each category vector describes the category likelihood. In yet another embodiment, the output result may be a vector which can be mapped to an exemplary classification category through the processing of a fully connected network processing. Various similar examples exist but are not elaborated herein.

Furthermore, in step 404, according to the output result, the attribute state or the attribute category of the to-be-predicted entity on the preset transaction attribute is determined. It is to be understood that the output result of the time-series neural network may be a vector or a score according to the description of step 403. Corresponding to the output

result, the manner of determining the preset transaction attribute of the to-be-predicted entity according to the output result may be as follows.

In one embodiment, the output result for a binary classification task is a score indicating the probability of being classified into a certain category. Then, the attribute state or attribute category of the to-be-predicted entity on the preset transaction attribute during a preset time period may be determined according to a preset score threshold.

In another embodiment, each category is represented by a vector, the similarity level between the output vector and each category vector describes the category likelihood. Then, an attribute state or attribute category corresponding to the category vector having the highest similarity level with the output vector may be determined as the attribute state or attribute category of the to-be-predicted entity on the preset transaction attribute.

In yet another embodiment, the output result may be a vector, which may be processed by a fully connected network to map the vector to an exemplary classification category.

In other embodiments, other manners may be adopted to determine, according to the output result, the attribute state or attribute category of the to-be-predicted entity on the preset transaction attribute during the preset time period. For example, the preset transaction attribute is the scaled income number, the output result is a score, and different scores may correspond to different scaled income numbers (for example, a score of 5 corresponds to an income of 5,000 RMB, etc.), and so on.

To sum up the above process, on the one hand, the method provided by the embodiments of this specification processes expression vectors of a to-be-predicted entity at different historical time points through a time-series neural network to take into account time-series characteristics of the to-be-predicted entity. On the other hand, node vector aggregation of neighbor nodes is performed for the expression vector of a single time point based on a relational network determined in accordance with corresponding entity states at the corresponding historical time points, fully considering influences from other entities associated with the to-be-predicted entity on the entity state, thereby generating an expression vector having a better expression capability for the to-be-predicted entity. Such expression vector having the time-series characteristic and the characteristics of being fused with neighbor nodes enables sufficient mining of the influences of environmental changes on entity attributes, thereby improving the accuracy of predicting entity transaction attributes.

According to embodiments of another aspect, an entity transaction attribute determination apparatus is further provided. The apparatus may be used for determining,

through a plurality of historical relational networks based on a temporal order, an attribute state or an attribute category of a to-be-predicted entity on a preset transaction attribute.

As shown in FIG. 5, an entity transaction attribute determination apparatus 500 may comprise:

- 5           a determining unit 51, configured to obtain a plurality of historical relational networks sequentially arranged in accordance with a temporal order, wherein the historical relational networks respectively correspond to historical time points having a preset time interval; each of the relational networks comprises a plurality of nodes, each of the nodes corresponds to a plurality of entities comprising the to-be-predicted entity; each of the historical relational  
10   networks is configured to describe association relationships between the plurality of entities which correspond to a corresponding historical time point; the entities having an association relationship are connected by a connecting edge therebetween, and each of the nodes corresponds to a node vector composed of transaction characteristics extracted based on description information of the corresponding entities;
- 15           a fusion unit 52, configured to determine, for each of the relational networks and through vector fusion of neighbor nodes, a plurality of expression vectors of the to-be-predicted entity;
- a time-series data processing unit 53, configured to process, through a pre-trained time-series neural network, the expression vectors to obtain an output result; and  
20           a prediction unit 54, configured to determine, according to the output result, the attribute state or the attribute category of the to-be-predicted entity on the preset transaction attribute.

According to an embodiment, the transaction characteristics in the node vector corresponding to a single node are extracted through description information at a  
25   corresponding historical time point of a corresponding relational network, or extracted through description information within a time interval between the corresponding historical time point and a previous historical time point of the corresponding relational network.

According to an embodiment, in the case that the to-be-predicted entity is a user, the entities in each of the historical relational networks comprise a plurality of users, and further  
30   comprise at least one of the following: an IP identifier or a WIFI identifier for a user device to access a network, and an application installed on the user device.

According to an embodiment, the historical relational networks include a first relational network, the to-be-predicted entity corresponds to a first node in the first relational network. The fusion unit 52 is further configured to:

process the relational network through a multilayer graph neural network, the multilayer graph neural network updates, after processing of a current layer is done, the node vector of the first node based on the following result: a weighted result from a product of the previous layer-updated node vector of each neighbor node and an auxiliary weight matrix;

5 and

use the node vector of the first node updated by the last layer of the graph neural network as a first expression vector of the first node corresponding to the first relational network.

According to a further embodiment, the neighbor nodes of the first node include a second node, and at the current layer, the fusion unit 52 is further configured to determine the following result as a weight of the second node relative to the first node:

an exponential form-based normalization result of a similarity level between the previous layer-updated node vector of the first node and the previous layer-updated node vector of the second node relative to a sum of similarity levels between the previous layer-updated node vector of the first node and the previous layer-updated node vector of each of the neighbor nodes.

According to a yet further embodiment, the fusion unit 52 is configured to determine the similarity level between the previous layer-updated node vector of the first node and the previous layer-updated node vector of the second node in the following manner:

20 concatenating the previous layer-updated node vector of the first node and the previous layer-updated node vector of the second node to form a concatenation vector;

performing, through a spatial auxiliary weight matrix, a dimensionality reduction process on the concatenation vector to obtain an intermediate vector having preset dimensions; and

25 **processing**, through an auxiliary vector having the preset dimensions, the intermediate vector to obtain the similarity level between the previous layer-updated node vector of the first node and the previous layer-updated node vector of the second node.

According to an embodiment, the prediction unit 54 is further configured to:

process the output vector through a fully connected neural network; and

30 determine, through the obtained processing result, the attribute state or the attribute category of the to-be-predicted entity on the preset transaction attribute.

According to an embodiment, in the case that the output result is an output vector, attribute states or attribute categories of the to-be-predicted entity on the preset transaction attribute further respectively correspond to description vectors; and the prediction unit 54 is

further configured to:

determine respective corresponding similarity levels between the output vector and each of the description vectors; and

5 determine an attribute state or attribute category corresponding to a description vector having the highest similarity level as the attribute state or attribute category of the to-be-predicted entity on the preset transaction attribute during a preset time period.

10 It is worth noting that the apparatus 500 in FIG. 5 is an apparatus embodiment corresponding to the method embodiment in FIG. 4, and the corresponding description for the method embodiment in FIG. 4 is also applicable to the apparatus 500, which is not further elaborated herein.

According to an embodiment of another aspect, a computer-readable storage medium is further provided and has a computer program stored thereon, and when the computer program is executed in a computer, the computer is caused to execute the method described in conjunction with FIG. 3 or FIG. 4.

15 According to an embodiment of yet another aspect, a computing device is further provided and comprises a memory and a processor, wherein executable code is stored in the memory; and when executing the executable code, the processor implements the method described in conjunction with FIG. 3 or FIG. 4.

20 One skilled in the art should appreciate that in one or more of the above-mentioned examples, the functions described in the embodiments of this specification may be implemented by hardware, software, firmware, or any combination thereof. When implemented by software, these functions may be stored in a computer-readable medium or transmitted as one or more instructions or code on the computer-readable medium.

25 The exemplary implementations described above illustrate, in further detail, the objective, technical solutions, and beneficial effects of the technical concept of this specification. It should be appreciated that the above are some implementations of the technical concept of this specification and is not used to limit the protection scope of the technical concept of this specification. Any modification, equivalent alternatives, improvement, and the like made on the basis of the technical solutions of the embodiments of this specification shall fall within the protection scope of the technical concept of this specification.

30

**What is claimed is:**

1. An entity transaction attribute determination method for determining an attribute state or an attribute category of a to-be-predicted entity on a preset transaction attribute, the method comprising:

- 5           obtaining a plurality of historical relational networks sequentially arranged in accordance with a temporal order, wherein the historical relational networks respectively correspond to historical time points having a preset time interval, each of the historical relational networks comprises a plurality of nodes, each of the nodes corresponds to a plurality of entities comprising the to-be-predicted entity, each of the historical relational
- 10   networks is configured to describe association relationships between the plurality of entities which correspond to a corresponding historical time point, the entities having an association relationship are connected by a connecting edge therebetween, and each of the nodes corresponds to a node vector composed of transaction characteristics extracted based on description information of the corresponding entities;
- 15           determining, for each of the historical relational networks and through vector fusion of neighbor nodes, a plurality of expression vectors of the to-be-predicted entity;
- processing, through a pre-trained time-series neural network, the expression vectors to obtain an output result; and
- determining, according to the output result, the attribute state or the attribute category
- 20   of the preset transaction attribute for the to-be-predicted entity,
- wherein the historical relational networks comprise a first relational network, the to-be-predicted entity corresponds to a first node in the first relational network, and
- the determining, for each of the historical relational networks and through vector fusion of the neighbor nodes, a plurality of expression vectors of the to-be-predicted entity
- 25   comprises:
- processing the relational network through a multilayer graph neural network, the multilayer graph neural network updating, after processing of a current layer is done, the node vector of the first node based on the following result: a weighted result from a product of the previous layer-updated node vector of each neighbor node and an auxiliary weight
- 30   matrix; and
- using the node vector of the first node updated by the last layer of the graph neural network as a first expression vector of the first node corresponding to the first relational network.

2. The method according to claim 1, wherein the time-series neural network is trained in the following manner:

obtaining a current training sample corresponding to a sample entity and an attribute  
5 tag of the sample entity with respect to the preset transaction attribute;  
executing, for each historical relational network of T historical relational networks  
sequentially arranged in accordance with the temporal order, a node vector fusion operation  
of the neighbor nodes, to determine T expression vectors respectively corresponding to the  
sample entity under the T relational networks, the T historical relational networks being  
10 established in accordance with T time points having the preset time interval;  
sequentially inputting, under the temporal order of the corresponding historical  
relational networks, the T expression vectors into a selected time-series neural network, and  
obtaining a sample output result of the time-series neural network for the sample entity; and  
adjusting, with an objective of matching the sample output result with the attribute  
15 tag, model parameters to train the time-series neural network.

3. The method according to claim 1, wherein the transaction characteristics in the node vector  
corresponding to a single node are extracted through description information at a  
corresponding historical time point of a corresponding relational network, or extracted  
20 through description information within a time interval between the corresponding historical  
time point and a previous historical time point of the corresponding relational network.

4. The method according to claim 1, wherein in the case that the to-be-predicted entity is a  
user, the entities in each of the historical relational networks comprise a plurality of users,  
25 and further comprise at least one of the following:

an IP identifier or a WIFI identifier for a user device to access a network, and  
an application installed on the user device.

5. The method according to claim 1, wherein the neighbor nodes of the first node comprise a  
30 second node, and at the current layer, a weight of the second node relative to the first node is:  
an exponential form-based normalization result of a similarity level between the previous  
layer-updated node vector of the first node and the previous layer-updated node vector of the  
second node relative to a sum of similarity levels between the previous layer-updated node  
vector of the first node and the previous layer-updated node vector of each of the neighbor

nodes.

6. The method according to claim 5, wherein the similarity level between the previous layer-updated node vector of the first node and the previous layer-updated node vector of the

5 second node is determined in the following manner:

concatenating the previous layer-updated node vector of the first node and the previous layer-updated node vector of the second node to form a concatenation vector;

performing, through a spatial auxiliary weight matrix, a dimensionality reduction process on the concatenation vector to obtain an intermediate vector having preset

10 dimensions; and

processing, through an auxiliary vector having the preset dimensions, the intermediate vector to obtain the similarity level between the previous layer-updated node vector of the first node and the previous layer-updated node vector of the second node.

15 7. The method according to claim 1, wherein in the case that the output result is an output vector, the determining, according to the output result, the attribute state or the attribute category of the preset transaction attribute for the to-be-predicted entity comprises:

processing the output vector through a fully connected neural network; and

20 determining, through an obtained processing result, the attribute state or the attribute category of the preset transaction attribute for the to-be-predicted entity.

8. The method according to claim 1, wherein in the case that the output result is an output vector, the attribute states or attribute categories of the preset transaction attribute for the to-be-predicted entity further respectively correspond to description vectors, and the

25 determining, according to the output result, the attribute state or the attribute category of the preset transaction attribute for the to-be-predicted entity comprises:

determining respective corresponding similarity levels between the output vector and each of the description vectors; and

30 determining an attribute state or attribute category corresponding to a description vector having the highest similarity level as the attribute state or attribute category of the preset transaction attribute for the to-be-predicted entity during a preset time period.

9. An entity transaction attribute determination apparatus, configured for determining, through a plurality of historical relational networks based on a temporal order, an attribute

state or an attribute category on a preset transaction attribute for a to-be-predicted entity, the apparatus comprising:

5 a determining unit, configured to obtain a plurality of historical relational networks sequentially arranged in accordance with a temporal order, wherein the historical relational networks respectively correspond to historical time points having a preset time interval; each of the relational networks comprises a plurality of nodes, each of the nodes corresponds to a plurality of entities comprising the to-be-predicted entity, each of the historical relational networks is configured to describe association relationships between a plurality of entities which correspond to a corresponding historical time point, the entities having an association  
10 relationship are connected by a connecting edge therebetween, and each of the nodes corresponds to a node vector composed of transaction characteristics extracted based on description information of the corresponding entities;

a fusion unit, configured to determine, for each of the relational networks and through vector fusion of neighbor nodes, a plurality of expression vectors of the to-be-predicted  
15 entity;

a time-series data processing unit, configured to process, through a pre-trained time-series neural network, the expression vectors to obtain an output result; and

a prediction unit, configured to determine, according to the output result, the attribute state or the attribute category of the preset transaction attribute for the to-be-predicted entity,  
20 wherein the historical relational networks comprise a first relational network, the to-be-predicted entity corresponds to a first node in the first relational network, and the fusion unit is further configured to:

process the relational network through a multilayer graph neural network, the multilayer graph neural network updates, after processing of a current layer is done, the node  
25 vector of the first node based on the following result: a weighted result from a product of the previous layer-updated node vector of each neighbor node and an auxiliary weight matrix; and

use the node vector of the first node updated by the last layer of the graph neural network as a first expression vector of the first node corresponding to the first relational  
30 network.

10. The apparatus according to claim 9, wherein the transaction characteristics in the node vector corresponding to a single node are extracted through description information at a corresponding historical time point of a corresponding relational network, or extracted

through description information within a time interval between the corresponding historical time point and a previous historical time point of the corresponding relational network.

11. The apparatus according to claim 9, wherein in the case that the to-be-predicted entity is a user, the entities in each of the historical relational networks comprise a plurality of users, and further comprise at least one of the following:

- an IP identifier or a WIFI identifier for a user device to access a network, and
- an application installed on the user device.

12. The apparatus according to claim 9, wherein the neighbor nodes of the first node comprise a second node, and at the current layer, the fusion unit is further configured to determine the following result as a weight of the second node relative to the first node:

- an exponential form-based normalization result of a similarity level between the previous layer-updated node vector of the first node and the previous layer-updated node vector of the second node relative to a sum of similarity levels between the previous layer-updated node vector of the first node and the previous layer-updated node vector of each of the neighbor nodes.

13. The apparatus according to claim 12, wherein the fusion unit is configured to determine the similarity level between the previous layer-updated node vector of the first node and the previous layer-updated node vector of the second node in the following manner:

- concatenating the previous layer-updated node vector of the first node and the previous layer-updated node vector of the second node to form a concatenation vector;

- performing, through a spatial auxiliary weight matrix, a dimensionality reduction process on the concatenation vector to obtain an intermediate vector having preset dimensions; and

- processing, through an auxiliary vector having the preset dimensions, the intermediate vector to obtain the similarity level between the previous layer-updated node vector of the first node and the previous layer-updated node vector of the second node.

14. The apparatus according to claim 9, wherein the prediction unit is further configured to:

- process the output vector through a fully connected neural network; and
- determine, through an obtained processing result, the attribute state or the attribute category of the preset transaction attribute for the to-be-predicted entity.

15. The apparatus according to claim 9, wherein in the case that the output result is an output vector, the attribute states or attribute categories of the preset transaction attribute for the to-be-predicted entity further respectively correspond to description vectors, and the prediction unit is further configured to:
- determine respective corresponding similarity levels between the output vector and each of the description vectors; and
  - determine an attribute state or attribute category corresponding to a description vector having the highest similarity level as the attribute state or attribute category of the preset transaction attribute for the to-be-predicted entity during a preset time period.
16. A computer-readable storage medium having a computer program stored thereon, and when the computer program is executed in a computer, the computer is caused to execute the method of any one of claims 1-8.
17. A computing device, comprising a memory and a processor, wherein executable code is stored in the memory, and when executing the executable code, the processor implements the method of any one of claims 1-8.