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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
09 February 2023 (09.02.2023)

WIPO | PCT

(10) International Publication Number
WO 2023/010502 A1

(51) International Patent Classification:
G06K 9/62 (2022.01) *G06N 5/02* (2006.01)
G06N 3/08 (2006.01) *G06F 17/16* (2006.01)
G06N 3/04 (2006.01)

(21) International Application Number:
PCT/CN2021/111102

(22) International Filing Date:
06 August 2021 (06.08.2021)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicants: **ROBERT BOSCH GMBH** [DE/DE]; Postfach 30 02 20, 70442 Stuttgart (DE). **TSINGHUA UNIVERSITY** [CN/CN]; Qinghua Yuan 1, Haidian District, Beijing 100084 (CN).

(72) Inventors: **KHARLAMOV, Evgeny**; Robert Bosch GmbH, Postfach 30 02 20, 70442 Stuttgart (DE). **TANG, Jie**; Tsinghua University, Qinghua Yuan 1, Haidian District, Beijing 100084 (CN). **CHEN, Bo**; Tsinghua University, Qinghua Yuan 1, Haidian District, Beijing 100084 (CN).

(74) Agent: **NTD PATENT & TRADEMARK AGENCY LTD.**; 10th Floor, Tower C, Beijing Global Trade Center, 36 North Third Ring Road East, Dongcheng District, Beijing 100013 (CN).

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

WO 2023/010502 A1

(54) Title: METHOD AND APPARATUS FOR ANOMALY DETECTION ON GRAPH

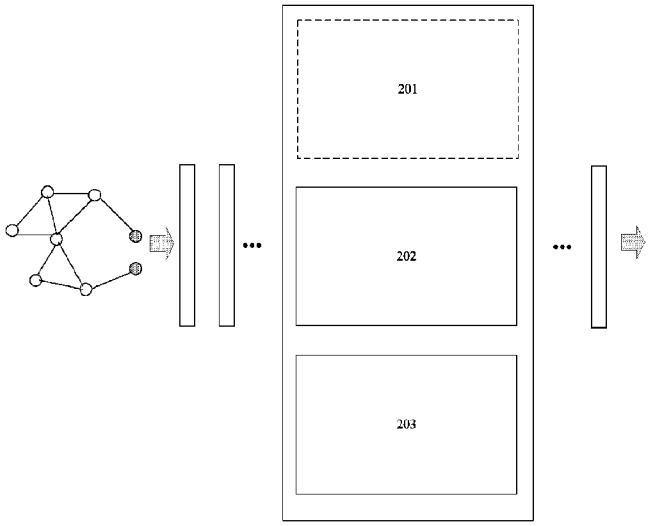


Fig.2

(57) Abstract: A method for training a Graph Neural Network (GNN) with multiple layers for anomaly detection is disclosed. The method comprising obtaining an embedding for each node of a graph and an embedding for the entire graph; inputting the embedding for each node and the embedding for the entire graph to the GNN, wherein the embedding for each node and the embedding for the entire graph are updated successively at each layer of the GNN; and updating the GNN based on a loss function.. The method further comprising edges of the graph are updated at each layer of the GNN before the embedding for each node and the embedding for the entire graph are updated. Numerous other aspects are provided.

[Continued on next page]

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

METHOD AND APPARATUS FOR ANOMALY DETECTION ON GRAPH

FIELD

[001] Aspects of the present disclosure relate generally to artificial intelligence, and more particularly, to a method and an apparatus provided for anomaly detection on a graph.

BACKGROUND

[002] Anomaly detection has profound impacts on preventing malicious activities in various applications, such as the detection of online review spams, financial frauds, fake users, and misinformation. The most promising developments have been the utilization of graph structures in machine learning models, as graphs can be used for naturally modeling the structural dependencies underlying the data.

[003] A significant line of works has been focused on discovering and incorporating the structural patterns of anomalies into learning models. Recently, the advances of graph neural networks (GNNs) have inspired and empowered various attempts to adopt GNNs for detecting anomalies. Existing attempts to address the graph-based anomaly detection have focused on structural feature engineering or learning. The main idea of GNN-based anomaly detection is to leverage the power of GNNs to learn expressive node representations with the goal of distinguishing abnormal and normal nodes in the embedding space, producing promising results.

[004] Contrastive learning is a widely used method to train an encoder to learn to produce vector representations of inputs such that representations of inputs in the same class will be more similar compared to representations of inputs in different classes. It is proven to be very useful for the downstream classification task, but is not specifically proposed to solve the anomaly detection problem.

[005] However, it may be not sufficient to only make inputs in the same class more similar, it could be more efficient to make inputs in different classes more distinguishing at the same time. It is promising to combine GNNs and contrastive learning for anomaly detection, with a finely designed objective.

SUMMARY

[006] The following presents a simplified summary of one or more aspects in order to provide a basic understanding of such aspects. This summary is not an extensive overview of all contemplated aspects, and is intended to neither identify key or critical elements of all aspects nor delineate the scope of any or all aspects. Its sole purpose is to present some concepts of one or more aspects in a simplified form as a prelude to the more detailed description that is presented later.

[007] In light of the contrastive learning, it is proposed in the present disclosure to contrast each node with the global context of the input graph. For the purpose of contrasting abnormal nodes with normal ones in terms of their distances to the global context, a context aware graph contrastive learning (CoGCL) model is disclosed based on graph contrastive learning and GNN. The underlying assumption is that abnormal nodes tend to be more distant from the majority of the nodes than normal ones.

[008] The present disclosure provides the context aware graph contrastive loss function in a supervised manner, i.e., labeled normal and abnormal nodes as positive and negative keys respectively. To achieve the contrastive objective, a graph neural network encoder is disclosed to learn the nodes' representations as well as learn the global context of the input graph during message passing. The graph neural network encoder is further disclosed to infer and remove suspicious links before learning the nodes' representations and learning the global context of the input graph. The disclosed encoder has corresponding modules for edge update, node update and graph update.

[009] In addition to the supervised model, it is also extended to an unsupervised model for handling cases with scarce labels. Straightforwardly, node labels that can be directly used to replace the ground-truth labels in the supervised contrastive loss function of CoGCL need to be synthesized.

[010] According to an aspect, a method for training a Graph Neural Network (GNN) with multiple layers for anomaly detection is disclosed. The method comprising: obtaining an embedding for each node of a graph and an embedding for the entire graph; inputting the embedding for each node and the embedding for the entire graph to the GNN, wherein the embedding for each node and the embedding for the entire graph are updated successively at each layer of the GNN; and updating the GNN based on a loss function.

[011] According to a further aspect, a portion of nodes is identified as abnormal nodes before input to the GNN based at least on ground-truth node labels.

[012] According to a further aspect, a portion of nodes is identified as abnormal nodes before input to the GNN based at least on generating the graph by: breaking a first graph into a plurality of sub-graphs; and injecting a set of nodes from sub-graphs other than a sub-graph which is to be used as the graph for inputting into the GNN, into the sub-graph as abnormal nodes.

[013] According to a further aspect, the loss function is to enforce maximizing the consistency between embeddings of normal nodes of the graph and the embedding for the entire graph compared with the consistency between embeddings of abnormal nodes of the graph and the embedding for the entire graph.

[014] According to a further aspect, edges of the graph are updated at each layer of the GNN before the embedding for each node and the embedding for the entire graph are updated.

[015] According to a further aspect, the edges of the graph are updated at each layer by estimating a suspicious probability of an edge between two nodes; and removing the edge based at least partially the suspicious probability of the edge.

[016] According to a further aspect, the embedding for each node is updated at each layer by aggregating embeddings of all neighboring nodes of the last layer for a node based at least on the edges of current layer; and combining the aggregated embedding with the embedding for the node of last layer.

[017] According to a further aspect, the embedding for the entire graph is updated at each layer by calculating a similarity between the embedding for each node of current layer and the embedding for the entire graph of last layer; and calculating the embedding for the entire graph of the current layer by weighted aggregating the embedding for each node of the current layer based on the corresponding similarities.

[018] A method for using a Graph Neural Network (GNN) trained with the method in the present disclosure for anomaly detection is disclosed. The method comprising obtaining an embedding for each node of a graph and an embedding for the entire graph; inputting the embedding for each node and the embedding for the entire graph to the GNN, wherein at least the embedding for each node and the embedding for the entire graph are updated successively at each layer of the GNN; and outputting, after final layer of the GNN, a cosine similarity between the updated embedding for a node and the updated embedding for the entire graph as an abnormal extent.

[019] According to a further aspect, edges of the graph are updated at each layer of the GNN before the embedding for each node and the embedding for the entire graph are updated.

[020] According to an aspect, a method for training an online acadamic system including a Graph Neural Network (GNN) with multiple layers for wrongly-assigned publication detection is disclosed. The method comprising: obtaining an embedding for each node of a graph and an embedding for the entire graph; inputting the embedding for each node and the embedding for the entire graph to the GNN, wherein the embedding for each node and the embedding for the entire graph are updated successively at each layer of the GNN; and updating the GNN based on a loss function.

[021] According to further aspect, the graph is constructed based on data of an online academic system, and wherein each node of the graph represents a researcher of the online academic system, and an edge exists if two publications have a certain common attribute, such as coauthors, venues or organizations.

[022] According to an aspect, a method for training an online spams recognition system including a Graph Neural Network (GNN) with multiple layers for spams detection is disclosed. The method comprising: obtaining an embedding for each node of a graph and an embedding for the entire graph; inputting the embedding for each node and the embedding for the entire graph to the GNN, wherein the embedding for each node and

the embedding for the entire graph are updated successively at each layer of the GNN; and updating the GNN based on a loss function.

[023] According to further aspect, the graph is constructed based on data of an online e-mail system, and wherein each node of the graph represents an account of the online e-mail system, and an edge exists if any e-mail exchange happens between two accounts.

[024] According to an aspect, a method for training a frauds recognition system including a Graph Neural Network (GNN) with multiple layers for anomaly detection is disclosed. The method comprising: obtaining an embedding for each node of a graph and an embedding for the entire graph; inputting the embedding for each node and the embedding for the entire graph to the GNN, wherein the embedding for each node and the embedding for the entire graph are updated successively at each layer of the GNN; and updating the GNN based on a loss function.

[025] According to further aspect, the graph is constructed based on data of a bank system, and wherein each node of the graph represents an account of the bank system, and an edge exists if any transfer happens between two accounts or the two accounts have a certain common attribute, such as account name.

[026] According to an aspect, a method for training a rumor prediction system including a Graph Neural Network (GNN) with multiple layers for anomaly detection is disclosed. The method comprising: obtaining an embedding for each node of a graph and an embedding for the entire graph; inputting the embedding for each node and the embedding for the entire graph to the GNN, wherein the embedding for each node and the embedding for the entire graph are updated successively at each layer of the GNN; and updating the GNN based on a loss function.

[027] According to further aspect, the graph is constructed based on data of a social network system, and wherein each node of the graph represents an account of the social network system, and an edge exists if any information flow happens between two accounts.

[028] The present disclosure can be used in a variety of applications, for example, in any scenario that inputs can be learned as node representations and connections between the nodes can be learned as edges, including but not limited to dataset preprocessing, detection of online review spams, financial frauds, insurance frauds, bot frauds, fake users and rumors in social media, fake news, etc..

BRIEF DESCRIPTION OF THE DRAWINGS

[029] The disclosed aspects will be described in connection with the appended drawings that are provided to illustrate and not to limit the disclosed aspects.

[030] Fig.1a illustrates an exemplary anomaly detection scenario, in accordance with various aspects of the present disclosure.

[031] Fig.1b illustrates exemplary similarity between normal nodes and global context versus abnormal nodes and the global context, in accordance with various aspects of the present disclosure.

[032] Fig.2 illustrates an exemplary module diagram of the GNN encoder for CoGCL at each layer, in accordance with various aspects of the present disclosure.

[033] Fig.3 illustrates an exemplary framework of CoGCL, in accordance with various aspects of the present disclosure.

[034] Fig.4 illustrates an exemplary process of creating corrupt graphs, in accordance with various aspects of the present disclosure.

[035] Fig.5 illustrates an exemplary flow chart of training a GNN for anomaly detection, in accordance with various aspects of the present disclosure.

[036] Fig.6 illustrates an exemplary flow chart of using the GNN trained with the method in Fig.5 for anomaly detection, in accordance with various aspects of the present disclosure.

[037] Fig.7 illustrates an exemplary computing system 700, in accordance with various aspects of the present disclosure.

DETAILED DESCRIPTION

[038] The present disclosure will now be discussed with reference to several example implementations. It is to be understood that these implementations are discussed only for enabling those skilled in the art to better understand and thus implement the embodiments of the present disclosure, rather than suggesting any limitations on the scope of the present disclosure.

[039] Various embodiments will be described in detail with reference to the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts. References made to particular examples and embodiments are for illustrative purposes, and are not intended to limit the scope of the disclosure.

[040] An important issue that data scientists are working to solve recently is anomaly detection, from network security to financial fraud, and further to misinformation propagated on social network, anomaly detection helps protect business, individuals and online communities. To this end, with mainly focused on structural features, graph-based anomaly detection has been widely used for detecting malicious activities in real-world applications.

[041] Fig.1a illustrates an exemplary anomaly detection scenario, in accordance with various aspects of the present disclosure.

[042] An anomaly detection problem is motivated by news reported in 2012 that a researcher named “Jun Lu" from Beijing University of Chemical Technology cheated the awards using the papers of another author who is also named “Jun Lu" but from

Yale University. This event caused by the wrongly-assigned papers of the same author name implies the importance of anomaly detection.

[043] To better understand the behavior of anomalies, a similar example is taken as a case study, by building a graph with papers assigned to a certain author as nodes, and connecting two papers with an edge if they share the same coauthors or venues. The goal in this example is to detect the wrongly-assigned papers (anomalies), i.e., those that are not authored by the certain author. As shown in Fig.1a, blank nodes represent papers authored by the certain author, and black nodes represent papers actually from other authors but wrongly-assigned.

[044] As another example, a graph can be constructed based on data of an online e-mail system, and wherein each node of the graph represents an account of the online e-mail system, and an edge exists if any e-mail exchange happens between two accounts. As yet another example, a graph can be constructed based on data of a bank system, and wherein each node of the graph represents an account of the bank system, and an edge exists if any transfer happens between two accounts or the two accounts have a certain common attribute, such as account name. And as other example, a graph can be constructed based on data of a social network system, and wherein each node of the graph represents an account of the social network system, and an edge exists if any information flow happens between two accounts.

[045] Fig.1a shows the t-SNE2 projection of each node's input features—the BERT embedding of its title, with blank as normal nodes and black as abnormal ones. It can be observed that both abnormal and normal nodes are distributed diversely, wherein abnormal ones being relatively more diverse. Intuitively, to quantify this observation, the similarity between each node and the global context, which is the average of all node features, is computed, which is showed in Figure 1b.

[046] Fig.1b illustrates exemplary similarity between normal nodes and global context versus abnormal nodes and the global context, in accordance with various aspects of the present disclosure. The similarity between normal nodes and global context is computed as a cosine similarity herein, and shown in the blank block. The similarity between abnormal nodes and the global context is computed as a cosine similarity herein, and shown with a shadowed block. On the left-side, it is the cosine similarity calculated based on the original input features, it suggests that though being relatively closed ($y \in [0.97,1]$), the two similarity distributions can be clearly distinguished. Inspired by these observations, it is desired a straightforward way to capture the differences for distinguishing abnormal and normal nodes. On the right-side, it is the cosine similarity calculated based on embeddings generated by CoGCL in the present disclosure, the two similarity distributions are more distant and more distinguishable as shown, illustrating that CoGCL is an ideal way for distinguishing the anomalies from the graph.

[047] A graph-based anomaly detection task can be formalized as following problem. With a graph denoted as $G = (V, X, A)$, where V is the set of N nodes, $A \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix, and X is the corresponding feature vectors with $x_i \in \mathbb{R}^d$ denoting the d-dimensional feature vector of $v_i \in V$. In consideration of generality, G is considered into an unweighted, undirected and single-relation graph, i.e., $A_{ij} = 1$ if an edge exists between v_i and v_j, and $A_{ij} = 0$ otherwise.

[048] Given a labeled graph $G = (V, X, A, Y)$, Y is the set of labels on nodes with $y_i \in Y$ takes value 1 if v_i is abnormal and 0 otherwise. In an example, the goal is to learn a classifier $g: \mathbb{R}^d \rightarrow \{0,1\}$ to determine whether a given node is abnormal *(1)* or normal *(0)*.

[049] To address this problem, most existing GNN-based models directly instantiate g as a binary classifier to predict the abnormal labels. Inspired by the observations about the distinguished distances to the global context (Fig.1b), contrastive learning is leveraged in the present disclosure for detecting the anomalies.

[050] In a first example, a CoGCL model is discussed in a supervised framework.

[051] The basic idea of the CoGCL model is to determine a node's abnormality by contrasting it with the global context (e.g., the average of all nodes) of the entire graph. This is motivated by the discovery that there exists a significant difference of distance to the global context between the normal and the abnormal nodes, that is, a node is more abnormal if it deviates farther away from the majority of nodes in the feature space.

[052] In view of this, the CoGCL model is disclosed to distinguish abnormal and normal nodes in the embedding space by leveraging graph contrastive learning. Specifically, given a graph G, our objective is to learn a GNN encoder that can output an embedding $\boldsymbol{h}_i$ for each node v_i and also embedding $\boldsymbol{q}$ for the entire graph (global context), i.e., $(H, \boldsymbol{q}) = f_{\text{GNN}}(X, A)$ with $H = \{\boldsymbol{h}_i\}_{i=1}^N$. One of the main ideas of the model is to take the graph embedding $\boldsymbol{q}$ as the query, a normal node's embedding as the positive key that matches with $\boldsymbol{q}$, and the embeddings of all the abnormal nodes as the negative keys that don't match with $\boldsymbol{q}$. These representations in the embedding space may then be used for downstream tasks, such as node classification, anomaly detection, etc.

[053] In an embodiment, a concrete loss function is designed as equation (1):

$$\mathcal{L} = \sum_{i:y_i=0} \left[-\log \frac{\exp(\mathbf{q}^\top \boldsymbol{h}_i/\tau)}{\sum_{j:y_j=1} \exp(\boldsymbol{q}^\top \boldsymbol{h}_j/\tau) + \exp(\boldsymbol{q}^\top \boldsymbol{h}_i/\tau)} \right] \quad (1)$$

where τ is the temperature hyperparameter. The objective function is to enforce maximizing the consistency between the positive pairs (normal nodes and global context) compared with the negative pairs (abnormal node, global context).

[054] Generally, for the purpose of optimizing an objective, a GNN model contains a message passing process and a neighbor aggregation process. Specifically, to optimize the context aware contrastive objective in equation (1), the CoGCL model contains several steps. For example, the CoGCL model contains a node update process and graph update process. For another example, the CoGCL model contains an edge update process, a node update process and graph update process. As a result, the CoGCL's GNN encoder f_{GNN} contains modules for corresponding processes.

[055] Fig.2 illustrates an exemplary module diagram of the GNN encoder for CoGCL at each layer, in accordance with various aspects of the present disclosure. As described below, some or all illustrated features may be omitted in a particular implementation within the scope of the present disclosure, and some illustrated features may not be required for implementation of all embodiments. It should be understood that modules shown with dashed lines in Fig. 2 indicate optional modules.

[056] A graph's representation in embedding space is input into the GNN. In an aspect, the graph is a labeled graph with a set of labels on nodes. In an aspect the graph is input into the GNN at least in the form of an embedding for each node and an embedding for the entire graph. In an aspect, the graph is input into the GNN with an adjacent matrix representing edges.

[057] As shown in Fig.2, the encoder f_{GNN} contains several modules to optimize the context aware contrastive objective in equation (1). In an aspect, the encoder f_{GNN} contains two modules, a node update module 202 and a graph update module 203. At each layer of GNN, the two modules are performed successively. In another aspect, the encoder f_{GNN} contains three modules, an edge update module 201, a node update module 202 and a graph update module 203. At each layer of GNN, the three modules are performed successively.

[058] An edge update module 201 is to estimate the suspicious probability of an edge and modify the adjacent matrix according to the probabilities, wherein

$$A^{(updated)} = f_{\text{edge}}\left(H^{(input)}, A^{(input)}, \boldsymbol{q}^{(input)}\right) \tag{2}$$

The suspicious edges are removed to reduce the negative influence from them thoroughly.

[059] An node update module 202 is then performed after the edge update module, to aggregate the neighboring information based on the modified adjacent matrix $A^{(updated)}$ and then combine the aggregated result with the concerned node, wherein

$$H^{(updated)} = f_{\text{node}}\left(H^{(input)}, A^{(updated)}\right) \tag{3}$$

[060] An graph update module 203 is then performed after the node update module, to update the context of the whole graph based on the updated node embeddings $H^{(updated)}$ and the input graph embedding $\boldsymbol{q}^{(input)}$ via a readout function:

$$\boldsymbol{q}^{(updated)} = f_{\text{graph}}\left(H^{(updated)}, \boldsymbol{q}^{(input)}\right) \quad (4)$$

[061] After all the layer convolutions, the updated node embedding $H^{(updated)}$, graph embedding $\boldsymbol{q}^{(updated)}$ and graph edges $A^{(updated)}$ are output for further processing.

[062] Fig.3 illustrates an exemplary framework of CoGCL model, in accordance with various aspects of the present disclosure. As described below, some or all illustrated features may be omitted in a particular implementation within the scope of the present disclosure, and some illustrated features may not be required for implementation of all embodiments. It should be understood that modules shown with dashed lines in Fig. 3 indicate optional modules.

[063] As shown in Fig.3, a graph G is input into the CoGCL model at 301. In an aspect, the graph G is a labeled graph with a set of labels Y on nodes. In an aspect, the graph G is input into the CoGCL model with an embedding for each node $H = \{\boldsymbol{h}_i\}_{i=1}^{N}$ and an embedding for the entire graph $\boldsymbol{q}$. The embedding for the entire graph $\boldsymbol{q}$ is obtained based at least on embeddings for each node $H = \{\boldsymbol{h}_i\}_{i=1}^{N}$ via a readout function. In an aspect, the graph G is input into the CoGCL model with an adjacent matrix A representing edges. As in Fig.3, the blank nodes are shown as normal nodes, while the black nodes are shown as abnormal nodes.

[064] It is illustrated in Fig.3 that the CoGCL model includes a plurality of repeat layers, with each layer at least includes corresponding modules of the encoder f_{GNN} as discussed with Fig.2. It is noted that the number of layers would not limit the scope of the invention.

[065] For example, layer 302-l includes two modules for node update and graph update respectively. For another example, layer 302-l includes three modules for edge update, node update and graph update respectively. At each layer, edge update as the first step, which is optional, is to estimate the likelihood of each link being a suspicious one and then remove the most suspicious ones. Then the node update is to update the node embeddings by message passing on the updated adjacency matrix. Finally, the global context is updated via graph update. After L-layer graph convolutions, the context aware contrastive loss function in equation (1) would be optimized, and the updated nodes' embeddings and graph embedding are output to 303 for further processing, such as calculating loss function for training, and calculating cosine similarity when for anomaly prediction, etc.

[066] In an embodiment, the encoder f_{GNN} contains an edge update module 302-l-1 in layer 302-l to remove suspicious links. Suspicious links are the ones that connect nodes with opposite labels. For example, almost every fraudster in business website can be linked to some benign users by the homophily assumption between neighbors – two neighboring nodes tend to have similar features and same labels – and impact the performance of the traditional message-passing along with the links.

[067] To predict the likelihood of a link, instead of based on the node features regardless of the node labels, both the node features $\boldsymbol{h}$ and one's distance to the global context $\mathbf{h} - \boldsymbol{q}$ in the feature space are considered. In the present disclosure, the distance can serve as implicit labels for guiding the model to remove suspicious links – those connected by the nodes with distinguished distance to the global context.

[068] In an aspect, a context aware link predictor is provided in the present disclosure to estimate the suspicious linkage likelihood between two nodes. The link predictor accepts the representations of two nodes as the input, where each node is represented by its own embedding and its relative distance to the global context embedding, i.e.,

$$\widetilde{\boldsymbol{h}}_i^{(l-1)} = \mathrm{MLP}\left(\boldsymbol{h}_i^{(l-1)} \oplus \left(\boldsymbol{h}_i^{(l-1)} - \boldsymbol{q}^{(l-1)}\right)\right) \tag{5}$$

Where $\oplus$ is the concatenation operator and MLP is a projection layer. $\left(\boldsymbol{h}_i^{(l-1)} - \boldsymbol{q}^{(l-1)}\right)$, the distance from node v_i to the global context, indicates that two nodes can be highly probably linked given their relative positions to the global context are similar, even if their local features are slightly different. In an aspect, the linkage likelihood between node v_i and v_j can be calculated as:

$$p_{ij}^{(l)} = \mathrm{ReLU}\left(\mathrm{cosine}\left(\widetilde{\boldsymbol{h}}_i^{(l-1)}, \widetilde{\boldsymbol{h}}_j^{(l-1)}\right)\right) \tag{6}$$

Where ReLU normalizes the cosine similarities into [0,1].

[069] In an aspect, the suspicious links are removed to reduce the negative influence from them thoroughly by employing Bernoulli approximation, the binary case of the Gumbel-Softmax reparameterization trick. Specifically, for each link $\left(\mathrm{A}_{\mathrm{ij}}^{(l-1)} = 1\right)$, a Gumbel noise $\varepsilon \sim$ Gumbel $(0,1)$ is first sampled, and then is added to the logarithm of the linkage likelihood $\mathrm{p}_{\mathrm{ij}}^{(l)}$ and is applied a sigmoid function. Then, the result is rounded as 1 if it is larger than 0.5 and 0 otherwise to obtain the updated $\mathrm{A}_{\mathrm{ij}}^{(l)}$, i.e.,

$$A_{ij}^{(l)} = \left\lfloor \frac{1}{1+\exp\left(-\left(\log p_{ij}^{(l)} + \varepsilon\right)/\lambda\right)} + \frac{1}{2} \right\rfloor \tag{7}$$

Where λ indicates the temperature of Gumbel-Softmax distribution. In an aspect, according to straight-through gradient estimator, the gradients is passed to the relaxed value $p_{ij}^{(l)}$ instead of the discrete value $A_{ij}^{(l)}$ during the backward procedure.

[070] In another aspect, to accelerate the optimization of the link, $\mathcal{L}_{\mathrm{link}} = \sum_{\mathrm{i,j}:\mathrm{A}_{\mathrm{ij}}^{(l-1)}=1} - \log \mathrm{p}_{\mathrm{ij}}^{(l)}$ is also added, which is the cross-entropy loss between the estimated

likelihood $p_{ij}^{(l)}$ and the observation $A_{ij}^{(l-1)}$ of the last layer, as the constraint of the loss function in equation (1).

[071] In an embodiment, the encoder f_{GNN} in layer 302-l contains a node update module 302-l-2 for updating node embeddings. In an aspect, the node embeddings are updated based on the updated edges. In another aspect, the node embeddings are updated without updating the edges of the graph.

[072] In an aspect, the operation in equation (3) is divided into an AGGREGATION and a COMBINE operations as:

$$a_i^{(l)} = \text{AGGREGATION}\left(\left\{\boldsymbol{h}_j^{(l-1)} : A_{ij}^{(l)} = 1\right\}\right) \tag{8}$$

$$\boldsymbol{h}_i^{(l)} = \text{COMBINE}\left(\boldsymbol{h}_i^{(l-1)}, \boldsymbol{a}_i^{(l)}\right) \tag{9}$$

Where equation (8) is to aggregate the neighboring information of the last layer based on the updated adjacency matrix and equation (9) is to combine the aggregated neighboring information with the concerned node. In an aspect, the AGGREGATION operation is the sum, the same as GIN. In an aspect, the COMBINE operation is concatenation.

[073] In an embodiment, the encoder f_{GNN} in layer 302-l contains a graph update module 302-l-2. The graph embedding $\boldsymbol{q}$ which is the global context, needs to be updated at each layer after all nodes' embeddings are updated.

[074] In an aspect, the graph embedding $\boldsymbol{q}$ can be obtained by straightforwardly performing one of sum pooling, max pooling or average pooling of all nodes in the graph. However, these kinds of operations do not distinguish the normal and abnormal nodes, which result in the inaccurate global context being deviated from the normal pattern.

[075] In another aspect, a memory buffer $\boldsymbol{m}$ is introduced to register the global context $\boldsymbol{q}^{(l-1)}$ of the last layer, based on which the contribution of each node to the global context is calculated. The main idea here is a node which is deviated from the current global context should be weakened in the next context update step. The concrete strategy can be formulated as:

$$s_i^{(l)} = \text{cosine}\left(\boldsymbol{h}_i^{(l)}, \boldsymbol{m}\right), \quad \alpha_i^{(l)} = \frac{\exp\left(s_i^{(l)}\right)}{\sum_{j=1}^{N} \exp\left(s_j^{(l)}\right)},$$

$$\boldsymbol{q}^{(l)} = \sum_{i=1}^{N} \alpha_i^{(l)} \cdot \boldsymbol{h}_i^{(l)}, \quad \boldsymbol{m} = \boldsymbol{q}^{(l)} \tag{10}$$

The memory are read out and the attention $\alpha_i^{(l)}$ for each node v_i is calculated based at least on its cosine similarity with the memory vector, which value is the global context $\boldsymbol{q}^{(l-1)}$ of the last layer. Then the global context $\boldsymbol{q}^{(l)}$ is updated by weighted

aggregating different nodes, and the updated global context is written into the memory to update the memory vector $\boldsymbol{m}$.

[076] After L-layer graph convolutions, the updated nodes' embeddings $H^{(L)}$ and graph embedding $\boldsymbol{q}^{(L)}$ are output to 303 for further processing.

[077] In an embodiment, for the graphs used for training, at 303, the loss function is calculated based at least on the updated nodes' embeddings $H^{(L)}$ and graph embedding $\boldsymbol{q}^{(L)}$.

[078] In another embodiment, for the graphs used for test, at 303, the updated nodes' embeddings $H^{(L)}$ and graph embedding $\boldsymbol{q}^{(L)}$ are output to 303 for anomaly prediction.

[079] In an aspect, for node $v_i \in V$ to be predicted, a binary classifier is used at 303 to determine whether a given node is abnormal or normal with a binary label indicating 1 or 0.

[080] In another aspect, instead of directly predicting a binary label for each node, for node $v_i \in V$ to be predicted, with its embedding $\boldsymbol{h}_i^{(L)}$ and the graph embedding $\boldsymbol{q}^{(L)}$ of the final layer, the cosine similarity between them is calculated as the abnormal extent of v_i. The lower the cosine similarity is, the more anomalous the v_i is predicted to be.

[081] In the above example, the CoGCL model is discussed in a supervised framework. While sufficient labels of anomalies are often arduously expensive to obtain, the CoGCL model can be extended as an unsupervised model to handle real-world applications with scarce or even no labels.

[082] In a second example, the CoGCL model is discussed in an unsupervised framework, especially focus on a pre-training strategy.

[083] Inspired by the idea of the context aware contrastive objective, it is proposed to construct pseudo anomalies via corrupting the original graph. Considering one small part of the original graph, nodes outside of this small part can be injected into it to act as pseudo anomalies. The underlying assumption is that as nodes in different parts of the graph follow different feature distributions, they can serve as the pseudo anomalies to the current original graph context.

[084] A corrupt graph is defined as follows. Given a graph $G = (V, X, A)$, which is broken into M parts $\{G_i\}_{i=1}^{M}$. For each part G_i, a corrupt graph $\tilde{G}_i$ is created by injecting a set of nodes $\bar{V}_i$ from the parts except G_i, i.e., $\bar{V}_i = \{v_j \in G \setminus G_i\}$, thus the corrupt nodes $\tilde{V}_i$ are the union of V_i and $\bar{V}_i$, i.e., $\tilde{V}_i = V_i \cup \bar{V}_i$. The corrupt adjacency matrix $\tilde{A}_i$ of $\tilde{G}_i$ is obtained by slicing A using the indexes in $\tilde{V}_i$.

[085] Fig.4 illustrates an exemplary process of creating corrupt graphs, in accordance with various aspects of the present disclosure. Firstly, the graph is broken into several

sub-graphs. There are different ways for breaking the graph into multiple parts, such as using community detection or clustering algorithms. In an aspect, the hierarchical agglomerative clustering algorithms can be used to cluster nodes into sub-graphs based on the initial features X and determine the size of the clusters via finding the optimal Silhouette Coefficient, which is a well-adopted clustering metric.

[086] Then the sub-graphs are injected into nodes from other sub-graphs as abnormal nodes. In an aspect, each corrupt sub-graph is input into the CoGCL model respectively.

[087] In an aspect, given the corrupt sub-graphs with injected nodes as pseudo anomalies and original nodes as normal ones, the same context aware contrastive loss function defined in equation (1) can be directly optimized to train f_{GNN}.

[088] In another aspect, the context aware contrastive loss function defined in equation (1) can be further fine-tuned on the target graph if the ground truth labels are also available.

[089] Fig.5 illustrates an exemplary flow chart of training a GNN for anomaly detection, in accordance with various aspects of the present disclosure. As described below, some or all illustrated features may be omitted in a particular implementation within the scope of the present disclosure, and some illustrated features may not be required for implementation of all embodiments. In some examples, the method may be carried out by any suitable apparatus or means for carrying out the functions or algorithm described below. It should be understood that operations shown with dashed lines in Fig. 5 indicate optional operations.

[090] The method begins at 501, with a graph G is given. The graph G is a graph with labels, which is denoted as $G = (V, X, A, Y)$, where V is the set of N nodes, $A \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix, and X is the corresponding feature vectors with $x_i \in \mathbb{R}^d$ denoting the d-dimensional feature vector of $v_i \in V$, Y is the set of labels on nodes with $y_i \in Y$ takes value 1 if v_i is abnormal and 0 otherwise.

[091] In an aspect, the set of labels Y is a set of ground-truth labels. In another aspect, the set of labels Y is obtained through a pre-training procedure, which would be discussed with the optional operation 500. In other aspect, the set of labels Y is a set of labels including both ground-truth labels and pre-trained labels.

[092] As shown in Fig.5, operations in 500 are optional. At 500, a graph G is generated by breaking a large graph G' into M parts $\{G_i\}_{i=1}^{M}$. Then for each part G_i, injecting a set of nodes $\bar{V}_i$ from the parts except G_i to create corresponding corrupt graph $\tilde{G}_i$. The corresponding corrupt graphs $\tilde{G}_i$ are input into the model respectively, each of the corrupt graphs is treated as an independent graph G.

[093] The method proceeds to 502 from 501, with obtaining an embedding for each node of the graph G and an embedding for the entire graph.

[094] The method proceeds to 503, with inputting the embedding for each node and the embedding for the entire graph to a Graph Neural Network (GNN) with multiple layers.

[095] In an aspect, at each layer of the GNN, the embedding for each node and the embedding for the entire graph are updated successively.

[096] As an example, the embedding for each node is updated in a message passing and neighbor aggregation process. In an embodiment, the embedding for each node is updated at each layer by aggregating embeddings of all neighboring nodes of the last layer for a concerned node based at least on the edges of current layer; and combining the aggregated embedding with the embedding for the concerned node of last layer. In an embodiment, the aggregating operation comprises summation, and the combining operation comprises concatenation.

[097] As a further example, the embedding for the entire graph is updated at each layer after the embedding for each node is updated at the current layer. In an embodiment, the embedding for the entire graph is obtained by performing one of sum pooling, max pooling or average pooling of all nodes in the graph. In another embodiment, the embedding for the entire graph is updated by calculating a similarity between the embedding for each node of current layer and the embedding for the entire graph of last layer; and calculating the embedding for the entire graph of the current layer by weighted aggregating the embedding for each node of the current layer based on the corresponding similarities. In a further embodiment, the similarity is cosine similarity, and the aggregating operation comprises summation.

[098] In another aspect, at each layer of the GNN, edges of the graph, the embedding for each node and the embedding for the entire graph are updated successively.

[099] As an example, the edges of the graph are updated at each layer by estimating a suspicious probability of an edge between two nodes; and removing the edge based at least partially the suspicious probability of the edge.

[100] As a further example, the embedding for each node is updated in a message passing and neighbor aggregation process. In an embodiment, the embedding for each node is updated at each layer after the edges is updated at current layer by aggregating embeddings of all neighboring nodes of the last layer for a concerned node based at least on the edges of current layer; and combining the aggregated embedding with the embedding for the concerned node of last layer. In an embodiment, the aggregating operation comprises summation, and the combining operation comprises concatenation.

[101] As a further example, the embedding for the entire graph is updated at each layer after the embedding for each node is updated at the current layer. In an embodiment, the embedding for the entire graph is obtained by performing one of sum pooling, max pooling or average pooling of all nodes in the graph. In another embodiment, the

embedding for the entire graph is updated by calculating a similarity between the embedding for each node of current layer and the embedding for the entire graph of last layer; and calculating the embedding for the entire graph of the current layer by weighted aggregating the embedding for each node of the current layer based on the corresponding similarities. In a further embodiment, the similarity is cosine similarity, and the aggregating operation comprises summation.

[102] After all layer graph convolution, the method proceeds to 504, with updating the GNN based on a loss function.

[103] In an aspect, the loss function is to enforce maximizing the consistency between embeddings of normal nodes of the graph and the embedding for the entire graph compared with the consistency between embeddings of abnormal nodes of the graph and the embedding for the entire graph.

[104] Fig.6 illustrates an exemplary flow chart of using the GNN trained with the method in the Fig.5 for anomaly detection, in accordance with various aspects of the present disclosure. As described below, some or all illustrated features may be omitted in a particular implementation within the scope of the present disclosure, and some illustrated features may not be required for implementation of all embodiments. In some examples, the method may be carried out by any suitable apparatus or means for carrying out the functions or algorithm described below.

[105] The method begins at 601, with a graph G is given for anomaly detection. The graph G is denoted as $G = (V, X, A)$, where V is the set of N nodes, $A \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix, and X is the corresponding feature vectors with $x_i \in \mathbb{R}^d$ denoting the d-dimensional feature vector of $v_i \in V$. In a further aspect, the graph G could be given with ground-truth labels as $G = (V, X, A, Y)$, wherein Y is the set of labels on nodes with $y_i \in Y$ takes value 1 if v_i is abnormal and 0 otherwise.

[106] The method proceeds to 602, with obtaining an embedding for each node of the graph G and an embedding for the entire graph.

[107] The method proceeds to 603, with inputting the embedding for each node and the embedding for the entire graph to a Graph Neural Network (GNN) with multiple layers.

[108] In an aspect, at each layer of the GNN, the embedding for each node and the embedding for the entire graph are updated successively.

[109] As an example, the embedding for each node is updated in a message passing and neighbor aggregation process. In an embodiment, the embedding for each node is updated at each layer by aggregating embeddings of all neighboring nodes of the last layer for a concerned node based at least on the edges of current layer; and combining the aggregated embedding with the embedding for the concerned node of last layer. In an embodiment, the aggregating operation comprises summation, and the combining operation comprises concatenation.

[110] As a further example, the embedding for the entire graph is updated at each layer after the embedding for each node is updated at the current layer. In an embodiment, the embedding for the entire graph is obtained by performing one of sum pooling, max pooling or average pooling of all nodes in the graph. In another embodiment, the embedding for the entire graph is updated by calculating a similarity between the embedding for each node of current layer and the embedding for the entire graph of last layer; and calculating the embedding for the entire graph of the current layer by weighted aggregating the embedding for each node of the current layer based on the corresponding similarities. In a further embodiment, the similarity is cosine similarity, and the aggregating operation comprises summation.

[111] In another aspect, at each layer of the GNN, edges of the graph, the embedding for each node and the embedding for the entire graph are updated successively.

[112] As an example, the edges of the graph are updated at each layer by estimating a suspicious probability of an edge between two nodes; and removing the edge based at least partially the suspicious probability of the edge.

[113] As a further example, the embedding for each node is updated in a message passing and neighbor aggregation process. In an embodiment, the embedding for each node is updated at each layer after the edges is updated at current layer by aggregating embeddings of all neighboring nodes of the last layer for a concerned node based at least on the edges of current layer; and combining the aggregated embedding with the embedding for the concerned node of last layer. In an embodiment, the aggregating operation comprises summation, and the combining operation comprises concatenation.

[114] As a further example, the embedding for the entire graph is updated at each layer after the embedding for each node is updated at the current layer. In an embodiment, the embedding for the entire graph is obtained by performing one of sum pooling, max pooling or average pooling of all nodes in the graph. In another embodiment, the embedding for the entire graph is updated by calculating a similarity between the embedding for each node of current layer and the embedding for the entire graph of last layer; and calculating the embedding for the entire graph of the current layer by weighted aggregating the embedding for each node of the current layer based on the corresponding similarities. In a further embodiment, the similarity is cosine similarity, and the aggregating operation comprises summation.

[115] After all layer graph convolution, the method proceeds to 604, with outputting indication for determining whether a node is anomaly.

[116] In an aspect, the indication is a binary label for the node to determine whether a given node is abnormal (1) or normal (0).

[117] In another aspect, the indication is a similarity between the updated embedding for the node and the updated embedding for the entire graph as an abnormal extent. For example, the similarity can be a cosine similarity.

[118] Fig.7 illustrates an exemplary computing system 700, in accordance with various aspects of the present disclosure. The computing system 700 may comprise at least one processor 710. The computing system 700 may further comprise at least one storage device 720. In an aspect, the storage device 720 may store computer-executable instructions that, when executed, cause the processor 710 to perform the method for training a Graph Neural Network (GNN) with multiple layers for anomaly detection, comprising: obtaining an embedding for each node of a graph and an embedding for the entire graph; inputting the embedding for each node and the embedding for the entire graph to the GNN, wherein the embedding for each node and the embedding for the entire graph are updated successively at each layer of the GNN; and updating the GNN based on a loss function. In a further aspect, the storage device 720 may store computer-executable instructions that, when executed, cause the processor 710 to perform the method for using a Graph Neural Network (GNN) trained with the present disclosure for anomaly detection, comprising: obtaining an embedding for each node of a graph and an embedding for the entire graph; inputting the embedding for each node and the embedding for the entire graph to the GNN, wherein at least the embedding for each node and the embedding for the entire graph are updated successively at each layer of the GNN; and outputting, after final layer of the GNN, a cosine similarity between the updated embedding for a node and the updated embedding for the entire graph as an abnormal extent.

[119] It should be appreciated that the storage device 720 may store computer-executable instructions that, when executed, cause the processor 710 to perform any operations according to the embodiments of the present disclosure as described in connection with FIGs.1-6.

[120] The embodiments of the present disclosure may be embodied in a computer-readable medium such as non-transitory computer-readable medium. The non-transitory computer-readable medium may comprise instructions that, when executed, cause one or more processors to perform any operations according to the embodiments of the present disclosure as described in connection with FIGs.1-6.

[121] The embodiments of the present disclosure may be embodied in a computer program product comprising computer-executable instructions that, when executed, cause one or more processors to perform any operations according to the embodiments of the present disclosure as described in connection with FIGs.1-6.

[122] It should be appreciated that all the operations in the methods described above are merely exemplary, and the present disclosure is not limited to any operations in the

methods or sequence orders of these operations, and should cover all other equivalents under the same or similar concepts.

[123] It should also be appreciated that all the modules in the apparatuses described above may be implemented in various approaches. These modules may be implemented as hardware, software, or a combination thereof. Moreover, any of these modules may be further functionally divided into sub-modules or combined together.

[124] The previous description is provided to enable any person skilled in the art to practice the various aspects described herein. Various modifications to these aspects will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other aspects. Thus, the claims are not intended to be limited to the aspects shown herein. All structural and functional equivalents to the elements of the various aspects described throughout the present disclosure that are known or later come to be known to those of ordinary skill in the art are expressly incorporated herein by reference and are intended to be encompassed by the claims.

CLAIMS

1. A method for training a Graph Neural Network (GNN) with multiple layers for anomaly detection, the method comprising:

obtaining an embedding for each node of a graph and an embedding for the entire graph;

inputting the embedding for each node and the embedding for the entire graph to the GNN, wherein the embedding for each node and the embedding for the entire graph are updated successively at each layer of the GNN; and

updating the GNN based on a loss function.

2 The method of claim 1, wherein a portion of nodes is identified as abnormal nodes before input to the GNN based at least on ground-truth node labels.

3. The method of claim 1, wherein a portion of nodes is identified as abnormal nodes before input to the GNN based at least on generating the graph by:

breaking a first graph into a plurality of sub-graphs; and

injecting a set of nodes from sub-graphs other than a sub-graph which is to be used as the graph for inputting into the GNN, into the sub-graph as abnormal nodes.

4. The method of claims 2 or 3, wherein the loss function is to enforce maximizing the consistency between embeddings of normal nodes of the graph and the embedding for the entire graph compared with the consistency between embeddings of abnormal nodes of the graph and the embedding for the entire graph.

5. The method of claim 1, the method further comprising:

edges of the graph are updated at each layer of the GNN before the embedding for each node and the embedding for the entire graph are updated.

6. The method of claim 5, wherein the edges of the graph are updated at each layer by:

estimating a suspicious probability of an edge between two nodes; and

removing the edge based at least partially the suspicious probability of the edge.

7. The method of claim 6, wherein estimating the suspicious probability of the edge between two nodes comprising:

estimating the suspicious probability of the edge between two nodes based at least on a similarity of the two nodes, wherein each node is represented by its own

node embedding and its distance to the embedding for the entire graph of the last layer.

8. The method of claim 5, wherein the embedding for each node is updated at each layer by:

aggregating embeddings of all neighboring nodes of the last layer for a node based at least on the edges of current layer; and

combining the aggregated embedding with the embedding for the node of last layer.

9. The method of claim 8, wherein the aggregating operation comprises summation, and the combining operation comprises concatenation.

10. The method of claim 5, wherein the embedding for the entire graph is obtained by performing one of sum pooling, max pooling or average pooling of all nodes in the graph.

11. The method of claim 5, wherein the embedding for the entire graph is updated at each layer by:

calculating a similarity between the embedding for each node of current layer and the embedding for the entire graph of last layer;

calculating the embedding for the entire graph of the current layer by weighted aggregating the embedding for each node of the current layer based on the corresponding similarities.

12. The method of claim 1, wherein the graph is constructed based on data of an online e-mail system, and wherein each node of the graph represents an account of the online e-mail system, and an edge exists if any e-mail exchange happens between two accounts.

13. The method of claim 1, wherein the graph is constructed based on data of an online academic system, and wherein each node of the graph represents a publication of a researcher, and an edge exists if two publications have a certain common attribute.

14. A method for using a Graph Neural Network (GNN) trained with one of claims 1-13 for anomaly detection, comprising:

obtaining an embedding for each node of a graph and an embedding for the entire graph;

inputting the embedding for each node and the embedding for the entire graph to the GNN, wherein at least the embedding for each node and the embedding for the entire graph are updated successively at each layer of the GNN; and

outputting, after final layer of the GNN, a cosine similarity between the updated embedding for a node and the updated embedding for the entire graph as an abnormal extent.

15. A computer system, comprising:

one or more processors; and

one or more storage devices storing computer-executable instructions that, when executed, cause the one or more processors to perform the operations of the method of one of claims 1-14.

16. One or more computer readable storage media storing computer-executable instructions that, when executed, cause one or more processors to perform the operations of the method of one of claims 1-14.

17. A computer program product comprising computer-executable instructions that, when executed, cause one or more processors to perform the operations of the method of one of claims 1-14.

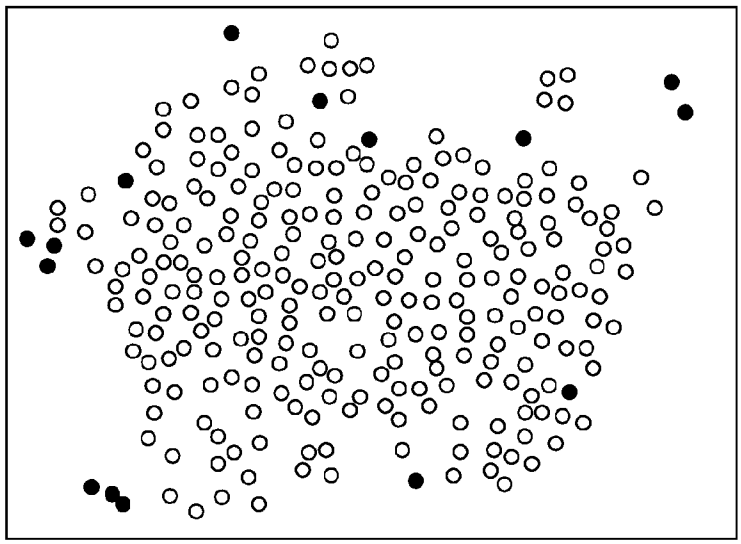


Fig.1a

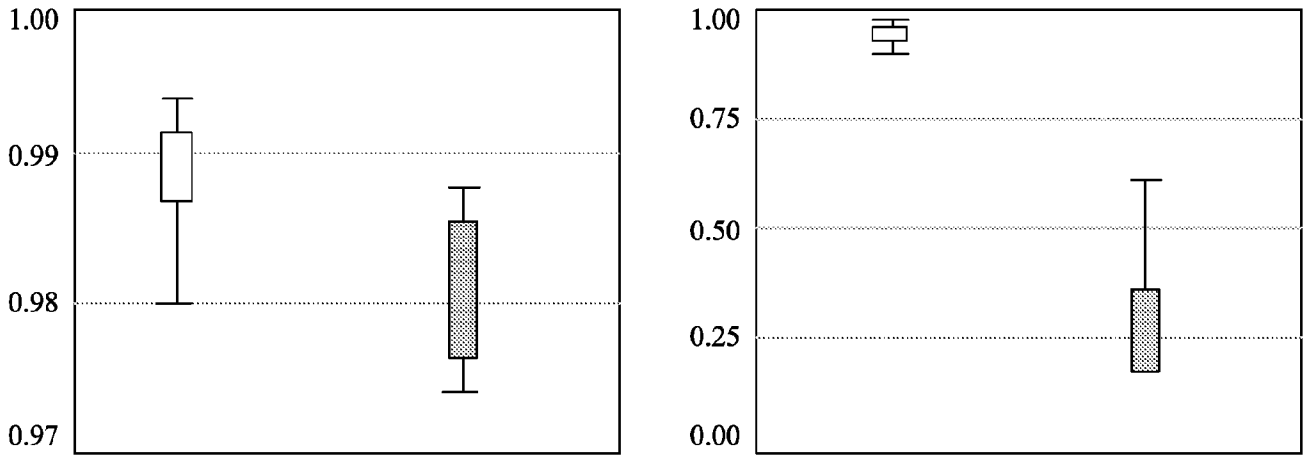


Fig.1b

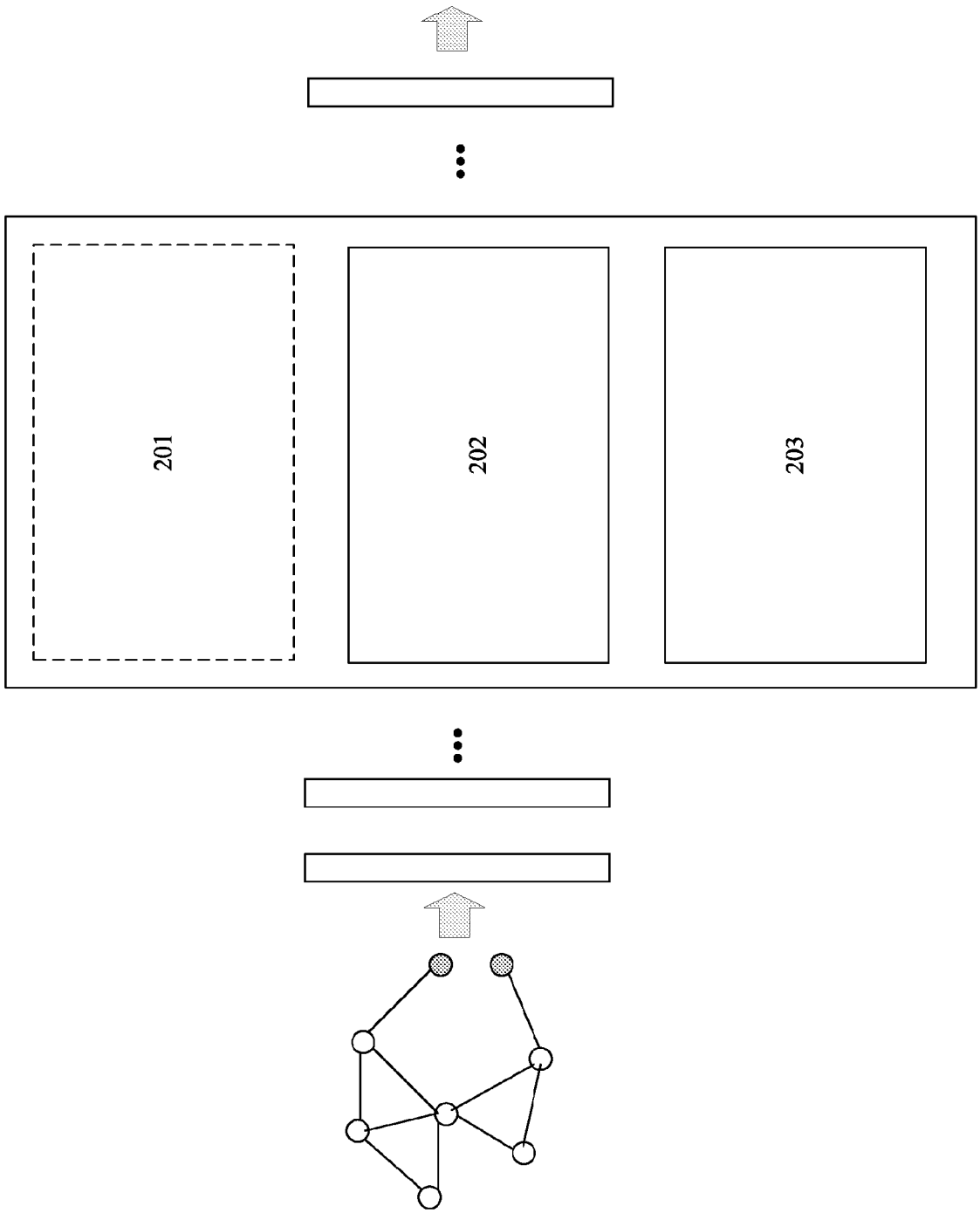


Fig.2

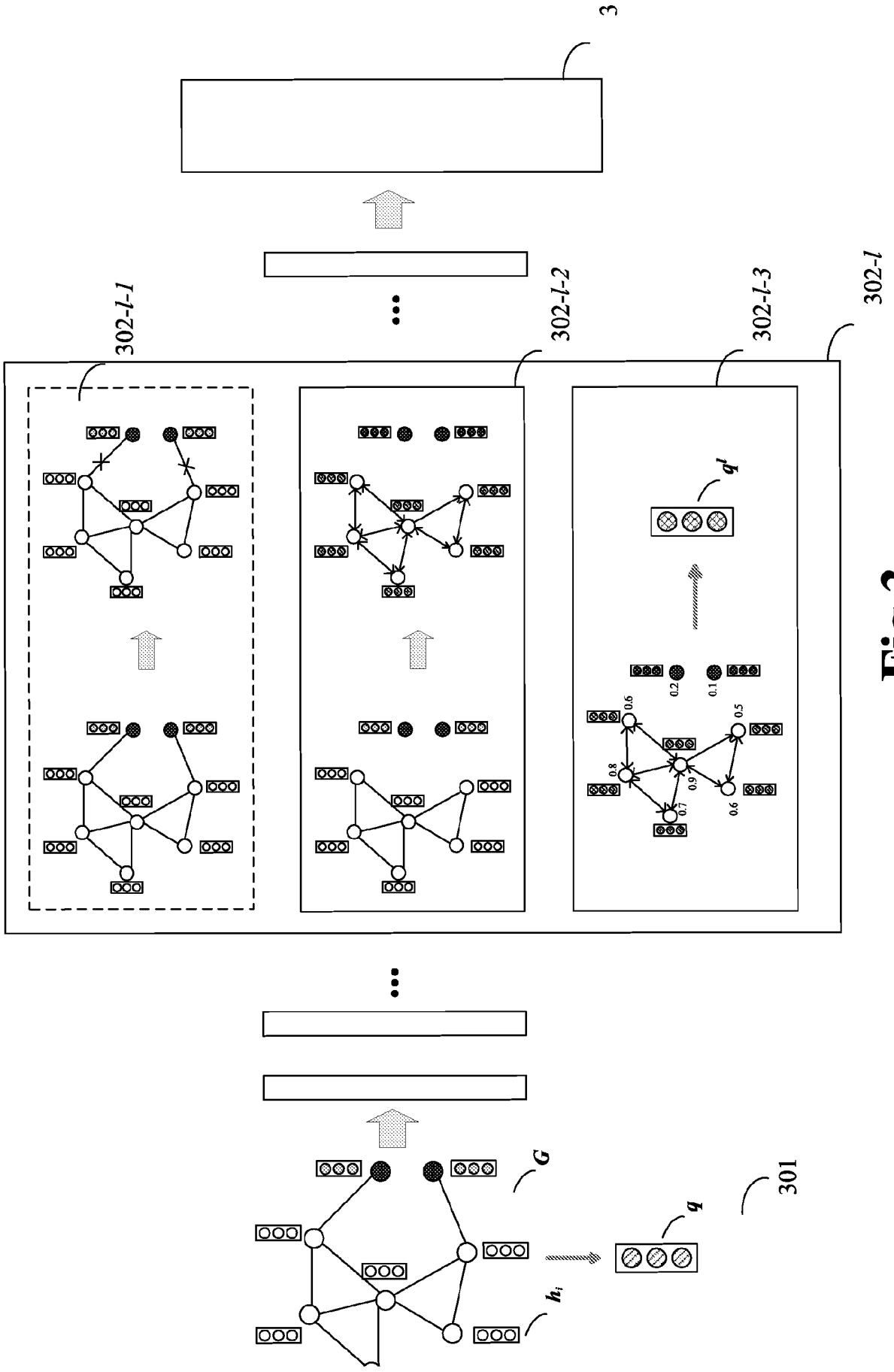


Fig.3

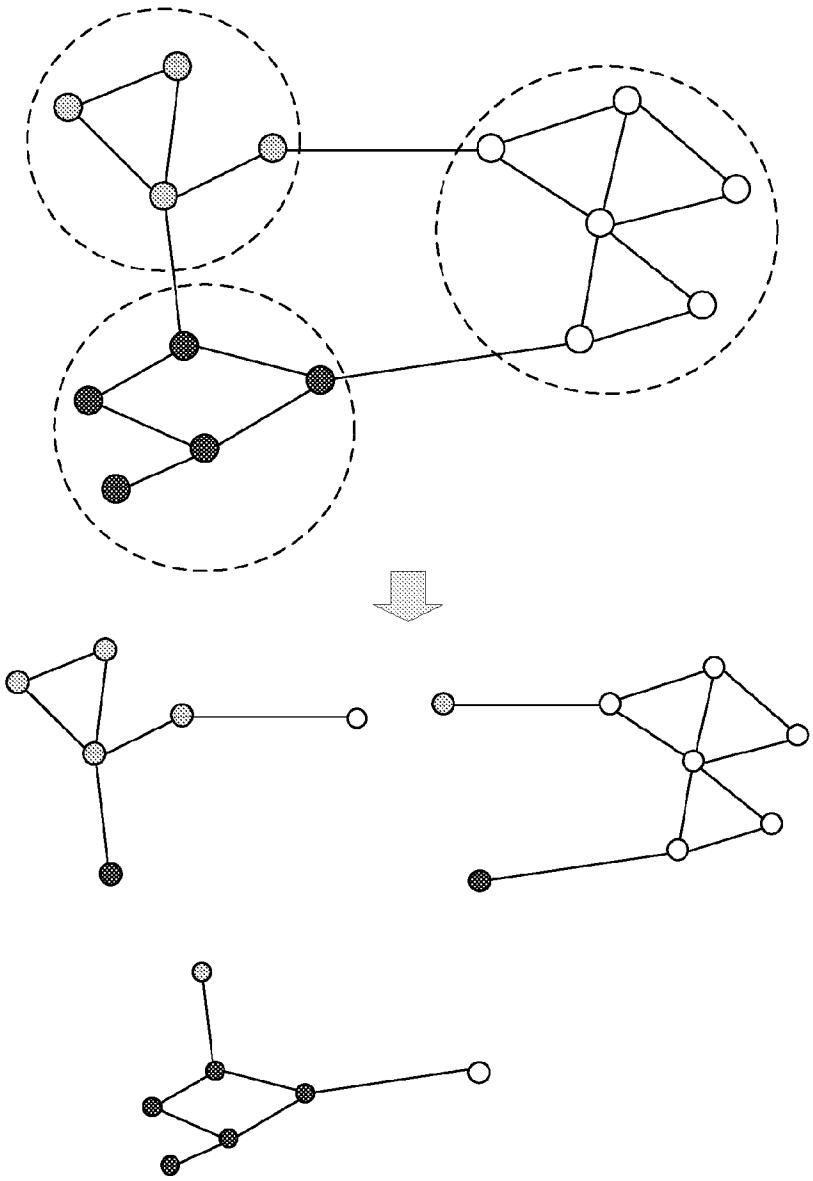


Fig.4

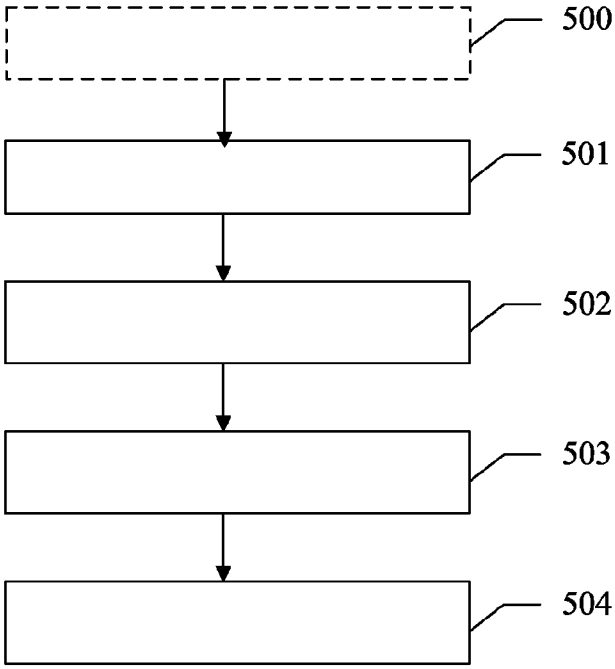


FIG.5

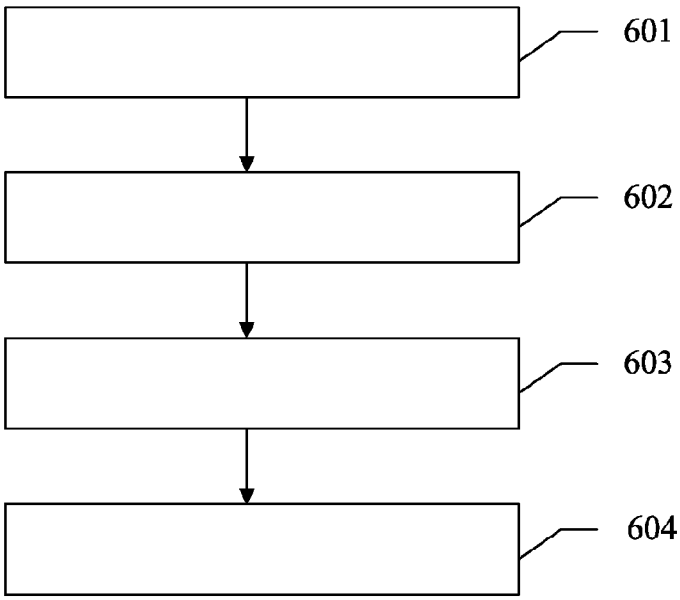


FIG.6

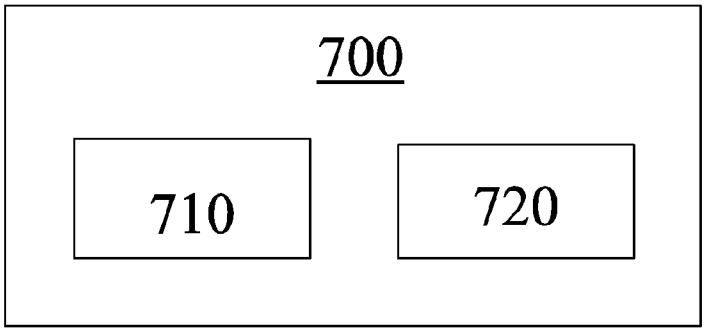


FIG.7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2021/111102

A. CLASSIFICATION OF SUBJECT MATTER

G06K 9/62(2022.01)i; G06N 3/08(2006.01)i; G06N 3/04(2006.01)i; G06N 5/02(2006.01)i; G06F 17/16(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06K; G06N; G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WPI, EPODOC, CNPAT, CNKI, IEEE: GNN, graph, neural, network, abnormal, detection, embedding, node, update, loss, function, label, edge, layer

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2021081717 A1 (BENEVOLENTAI TECHNOLOGY LIMITED) 18 March 2021 (2021-03-18) claims 1-43	1-17
A	US 2021103797 A1 (LUNIT INC.) 08 April 2021 (2021-04-08) the whole document	1-17
A	US 2021232918 A1 (NEC LABORATORIES AMERICA, INC.) 29 July 2021 (2021-07-29) the whole document	1-17
A	US 2021004210 A1 (X DEVELOPMENT LLC) 07 January 2021 (2021-01-07) the whole document	1-17
A	US 2021067549 A1 (NEC LABORATORIES AMERICA, INC.) 04 March 2021 (2021-03-04) the whole document	1-17

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
09 March 2022	**01 April 2022**
Name and mailing address of the ISA/CN **National Intellectual Property Administration, PRC** **6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing** **100088, China** Facsimile No. **(86-10)62019451**	Authorized officer **ZHANG,Tao** Telephone No. **86-(10)-53961356**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2021/111102

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
US 2021081717 A1	18 March 2021	CN 112119412 A	22 December 2020
		WO 2019220128 A1	21 November 2019
		EP 3794511 A1	24 March 2021
US 2021103797 A1	08 April 2021	KR 102068277 B1	11 February 2020
US 2021232918 A1	29 July 2021	None	
US 2021004210 A1	07 January 2021	None	
US 2021067549 A1	04 March 2021	None	