Knowledge Graph Representation Learning Model Based On Capsule Network And Information Fusion

Chu Zhao[1,3], Gilja So[2*], and Rui Chen[3]

[1] Department of Computer and Information Engineering, Graduate School Youngsan University, South Korea
[2] Department of Cyber Security Youngsan University, South Korea
[3] Software College, Shenyang Normal University, Shenyang, 110034, China
[*] Corresponding author. E-mail: kjso@ysu.ac.kr

Received: Aug. 30, 2024; Accepted: Mar. 31, 2025

In recent years, knowledge representation learning has played a key role in intelligent recommendation, intelligent question-answering, and intelligent retrieval, and has been widely concerned. Knowledge representation learning aims to vectorize semantic information and deduce knowledge through mathematical formulas with the help of low dimensional embedding of entity and relation. Although knowledge representation learning based on knowledge graph can obtain entity structure and relational embedding, it lacks semantic information utilization of entity description text. In addition, with the increase of the scale of the knowledge graph, the categories and quantities of entities and relationships, as well as the content and sources of entity descriptions, the correspondence between the textual descriptions of entities and the triplet structure information becomes more difficult to obtain. Therefore, we propose a novel knowledge graph representation learning model based on capsule network and information fusion in this paper. Based on anchor node and neighbor node and the relational sampling strategy, each node on the knowledge graph is represented by the predicted operator graph. The capsule network is used to gather the image features for each node to obtain the node representation vector, which is finally input to the decoder to calculate the score. In particular, we construct a loss function for the multi-layer attention mechanism of entity structure and semantic fusion. Experimental results show that the proposed method can effectively deduce the hidden link relationship between entities containing complex entity descriptions, and has more accurate classification accuracy than other methods in triplet classification tasks.

Keywords: Knowledge representation learning; Capsule network; Information fusion; Multi-layer attention mechanism

http://dx.doi.org/10.6180/jase.202601_29(1).0009

1. Introduction

Knowledge graph [1] is a large multi-relationship graph that is widely used in the field of entity and relationship space. Where nodes correspond to entities, and relationships between entities are described by typed edges, and facts are often encoded into triples, which are presented in the form (entity, relationship, entity). Since the knowledge graph was first proposed to the present long-term development and evolution process, a large number of knowledge graphs had been constructed, such as DBPedia [2], Knowledge Cube [3], YAGO [4] and so on. These structured and large-scale knowledge bases cover a large number of fields and involve a large amount of knowledge, which not only greatly facilitates people's search for knowledge in application, but also drives the development of knowledge graph in representation learning. However, due to the continuous expansion of knowledge scale and the shortening of data update cycle, the phenomenon of incomplete knowledge is becoming more and more obvious. In order to solve the problem of incomplete knowledge, knowledge representation learning [5] is put forward. Its

main idea is to change the way of representing knowledge by triples (h, r, t), and use the knowledge of machine learning to convert the semantic information of triples (h, r, t) to achieve vectorization of entities and relations while keeping the structure of knowledge unchanged [6, 7]. A large number of mathematical formulas are used to carry out the calculation, so as to complete the work of knowledge graph.

Due to the high computational efficiency of knowledge representation learning, a large number of knowledge representation learning models have been established, such as TransE [8], TransH [9], HyTE, IPTransE, PTransE-RNN [10], puTransE, etc. Among the above typical models, TransE is considered to be the most promising model due to its few score function parameters, easy operation, and ease of data sparsity. However, TransE has certain limitations when dealing with complex relationships other than one-to-one. Specifically, in the face of complex relationships such as one-to-many, many-to-one, many-to-many and reflexive, TransE is prone to the problem of wrong differentiation between different entities with the same relationship, and the model expression ability is also limited. For example, suppose that there are two triples in the knowledge base, namely (apple, is, fruit) and (orange, is, fruit), which belong to the many-to-one relationship, TransE cannot correctly distinguish between the two entities of "apple" and "orange", and it is easy to conclude that the "apple" entity is equal to the "orange" entity. In fact, the attributes, roles, and status of the two entities are different, so the two entities are not the same.

The absence of links (relationships) between entities of the knowledge graph are called the incompleteness of the knowledge graph. Even the most advanced knowledge maps have imperfections, with over 70% of human entities in the largest and most widely used FreeBase knowledge Map having no known place of birth and over 99% having no known race. This has led researchers to come up with various techniques for supplementing knowledge or correcting errors and adding missing facts to the knowledge graph, which is also known as the knowledge graph completion task. One branch of this is completion by inferring missing facts from facts that already exist in the knowledge graph, called link prediction (LP) [11].

Researchers have carried out a wealth of knowledge representation learning work, combined with the proposed algorithm in this paper, elaborated two kinds of related knowledge representation learning methods.

1.1. Knowledge representation learning based on translation ideas

Early knowledge representation learning focused on structural embedding of triples, the most representative of which was TransE. It treats the relationship between entities as a mapping from the head entity to the tail entity. Although it is simple and effective to use TransE to describe the entities and relations of triples, it is difficult to accurately describe the complex entity correspondence, such as, an entity exists in multiple relations (denoted as $1-N$); multiple entities exist in the same relationship $(N-1)$; or multiple relationships exist between multiple entities (denoted as $N-M$). To solve the above problems, the researchers proposed a variety of derivative models of TransE. Where, TransH assumed that the entity vector and the relational vector were located in different hyper-planes, and mapped the head entity vector and tail entity vector to the hyperplane where the relational vector resided, respectively. TransR projected the head entity and tail entity vectors into a specific relation space through matrix transformation. Unlike TransR, TransD dynamically obtained the transformation matrix from entity vector to relational vector for the $N-M$ mapping between entity and relation. Based on the TransR model, TranSparse used sparse matrix to describe heterogeneous relationships, and different sparse projection matrices were used to realize the mapping from the head entity to the tail entity and solve the problem of relationship imbalance. TransA improved the loss function embedded in triples according to the adaptive strategy. KG2E represented entities and relations as random vectors extracted from a multidimensional Gaussian distribution, effectively reflecting the uncertainty of entities and relations in the knowledge graph. TransG [12] used Gaussian distribution to model entities and relationships, believing that a relationship could have multiple semantics. TransE-SNS generated negative example triples based on entity similarity to improve the quality of knowledge graph embedding.

The above improved models provide an effective solution for complex relation representation to some extent, but they still use triplet structure to embed information. It does not take full advantage of the rich semantics in the entity text description.

1.2. Knowledge representation learning method combining with entity description

In order to more rationally describe the complex mapping between entities and relationships, researchers try to combine the structural embedding of triples with the semantic information of text descriptions. Tang et al. [13] proposed a

neural tensor network (NTN) model, which embedded entity description and triplet separately, but did not model the interaction between them. Considering the incompleteness of data in knowledge graph, An et al. [14] adopted natural language processing technology to extract relational knowledge directly from plain text. Zhao et al. [15] applied deep neural networks to the extraction of text relational vectors. Li et al. [16] referred to the Paragraph Vector model and used entity description to assist entities in triplets for vector expression, but did not distinguish the text word order corresponding to entity description. Li et al. [17] built three kinds of neural network models to realize semantic information extraction in entity description, and designed a gating mechanism to map structural information and semantic information to the same vector space. Hao et al. [18] proposed the description embodied knowledge representation learning (DKRL) model and adopted the continuous bag of words (CBOW) model to encode the entity description. Mousselly-Sergieh et al. [19] proposed a TEKE (text enhanced knowledge embedding) model and used word2vec and TransH to obtain the text description corresponding to the entity and the vector representation of triple structure information, respectively. Passalis et al. [20] proposed a TransET model to learn the semantic information contained in the corresponding categories of entities. Chen et al. [21] introduced a graph convolutional neural network (GCN) to learn the semantic information between the entity's corresponding nodes and their neighbors, applied it to the entity embedding, and then combined it with the entity description. Gan et al. [22] proposed a model based on reference sentences, which selected high-quality reference sentences through the attention mechanism and then integrated the semantic information contained in structural embedment and entity description. Wang et al. [23] enriched the semantic expression of triples by integrating the word order features in entity description with the structural information of triples through long short-term memory (LSTM). Zhao et al. [24] used convolutional neural networks (CNNs) to extract semantic relationships from text descriptions corresponding to entities, and used attention mechanisms to distinguish the credibility of different semantic relationships, but the semantic relationships obtained from text descriptions were not accurate enough. Zhang et al. [25] used semantic information in entity description to embed entity and relation into hyperbolic space for training. Niu et al. [26] used meta-path to evaluate the relationship between entities and the similarity of text information, so as to improve the accuracy of the model. Liu et al. [27] combined medical imaging information, text information and triplet structure information according to specific medical data sets, which significantly improved model performance. Shen et al. [28] proposed an adaptive sampling algorithm to realize the aggregation of neighbor node features and further combine it with entity description. Seo et al. [29] proposed dimensional attention composition (DAC) method to adjust model parameters that fused structure and entity description information, effectively solving the sparsity of knowledge graph. Fan et al. [30] proposed a multi-scale capsule-based embedding model incorporating entity descriptions model, which used capsule network and attention mechanism to obtain semantic information in entity descriptions.

These methods combine knowledge representation learning with entity semantic description to give fusion representation of triples. However, the above methods only propose the semantic information acquisition method of single source entity description, and the expansion of the scale of knowledge graph will inevitably lead to the emergence of complex multi-source entity description. Therefore, it is of great significance to obtain the semantic information of complex multi-source entity description better and combine it with triplet structure information.

In knowledge representation learning, in addition to the structural information of triples, the text description corresponding to the entities in triples can also be used, which contains rich semantic information to improve the knowledge connotation of triples. However, with the increase of the scale of knowledge graph, the categories and quantities of entities and relationships, as well as the content and sources of entity descriptions, increase accordingly, and the correspondence between entity text descriptions and triplet structure information becomes more difficult to obtain. However, the existing knowledge representation learning methods combining entity descriptions do not solve this problem [31].

The innovations of the proposed method are as follows: (1) The subgraph sampling method based on anchor node is further developed, and the anchor node, neighbor, context relationship and itself of the target node are simultaneously sampled to obtain subgraphs containing more information; (2) A context information transfer method using capsule network is proposed to fuse the neighbor and context information in the image, so that data types can be unified into node types before entering Transformer aggregator; (3) A deeper and more efficient knowledge graph representation learning model is constructed, with a total of two convolutional layers, one multi-attention layer and four linear layers, forming a 7-layer representation learning model.

2. Materials and methods

As shown in Fig. 1, the proposed model consists of an encoder and a decoder. The sampling graph of each node is calculated in advance and then the subgraph features of the node are aggregated by the encoder according to the sampling graph. The final input to the decoder calculates the prediction probability of the score function output knowledge triples, training the parameters of the embedding vector, encoder and decoder.

As shown in Fig. 2, instead of sampling subgraphs of nodes during training, each node is pre-assigned a sampling graph consisting of anchor nodes and context neighbors, relationships, and central nodes (themselves) before training. The appearance map contains the structure topological information and rich neighborhood information of the graph, which can be improved by encoder processing. The neighborhood information is encoded through the capsule network, and finally feature aggregation is carried out through Transformer encoder and average pooling layer.

2.1. Subgraph sampling

Given a directed knowledge graph data set $G = (N, E, R)$. $|N|$ Indicates the number of nodes. $|E|$ represents the number of relationships. $|R|$ indicates the number of relationship types. It adds an inverse edge and a $[PAD]$ edge for each directed edge of a digraph, and adds a self-ring edge. So $|R| = |R|_{direct} + |R|_{inwerse} + 2$. For nodes, it add a $[PAD]$ node, so $|N| = |N| + 1$. It samples a subgraph $g(T) = sample(T)$ for each target node T that can represent its features.

For a knowledge graph G, the direct application of graph encoder will result in too much computation or even impossible computation. In order to make use of the information of nodes and their neighbors, each node is serialized by means of subgraph generated by precomputed nodes, and the whole subgraph generation process is set outside the calculation and training of the model, so as to avoid problems such as too much computation and too complicated. In the field of natural language processing, some excellent methods such as WordPiece [32] and BERT [28] propose to sample a fixed number of vocabularies to represent words in natural language. Inspired by NodePiece, subgraphs are generated for each node according to a certain policy.

Firstly, it is considered that the anchor node is mainly a node that provides general information, and it should be a node with a higher degree in the graph. Therefore, a fixed number of anchor nodes are sampled according to the descending order of degrees (PageRank algorithm, and the method combining three random strategies (4:4:2)).

Compared with the anchor node, the neighbor node is closer to the target node, so it is defined as a node that provides unique information, and also samples a fixed number of one-hop neighbor nodes in descending order. In addition, in order to ensure a strong degree of differentiation between the generated subgraphs, the target node itself is sampled as a supplement to provide unique information.

In the stage of pre-calculation and subgraph generation, a fixed number of anchor nodes of target nodes are selected from the anchor node set according to width priority, and the distance between anchor nodes and target nodes is recorded. A fixed number of neighbor nodes and their corresponding relationships are also sampled, and finally, their own nodes and self-loop edges are added to form a subgraph representation of target nodes. This process can be expressed as:

$$g(T) = sample(T) = A, D, N, R \tag{1}$$

$$A = a_1, a_2, \cdots, a_k \tag{2}$$

$$D = d_1, d_2, \cdots, d_k \tag{3}$$

$$N = n_0, n_1, n_2, \cdots, n_m \tag{4}$$

$$R = r_0, r_1, r_2, \cdots, r_m \tag{5}$$

Where A represents the set of sampled k anchor nodes. D represents the distance of each anchor node relative to the target node T. N represents the set of m neighbors sampled. R represents the set of relationships corresponding to each neighbor. n_0 and r_0 represent their own nodes and their own loop edges, respectively. For nodes with less than m neighbors, the $[PAD]$ nodes and $[PAD]$ edges are used for filling, and $[PAD]$ nodes and edges are not involved in training.

2.2. Feature graph mapping based on capsule network

In this paper, text description information (h_d, t_d) and triplet structure information (h_s, t_s) are fused and trained jointly. Two vectors of different types of entities are mapped into the same semantic space and share the same relational vector. The energy function is shown in Eq. (6).

$$\begin{aligned} E(h, r, t) =& \alpha_1 ||h_s + r - t_s|| + \alpha_2 ||h_s + r - t_d|| \\ &+ \alpha_3 ||h_d + r - t_s|| + \alpha_4 ||h_d + r - t_d|| \end{aligned} \tag{6}$$

Among them, α_1, α_2, α_3, α_4 are the hyperparameters that control each function. The fused results are the triplet vector V_h, V_r, V_t as shown in Eqs. (7) to (9). Where W_s, and W_d are shared parameters, $\odot$ is the Hadamard product, and it is the matrix multiplication of the same order.

$$V_h = W_s \odot h_s + W_d \odot h_d \tag{7}$$

$$V_r = r \tag{8}$$

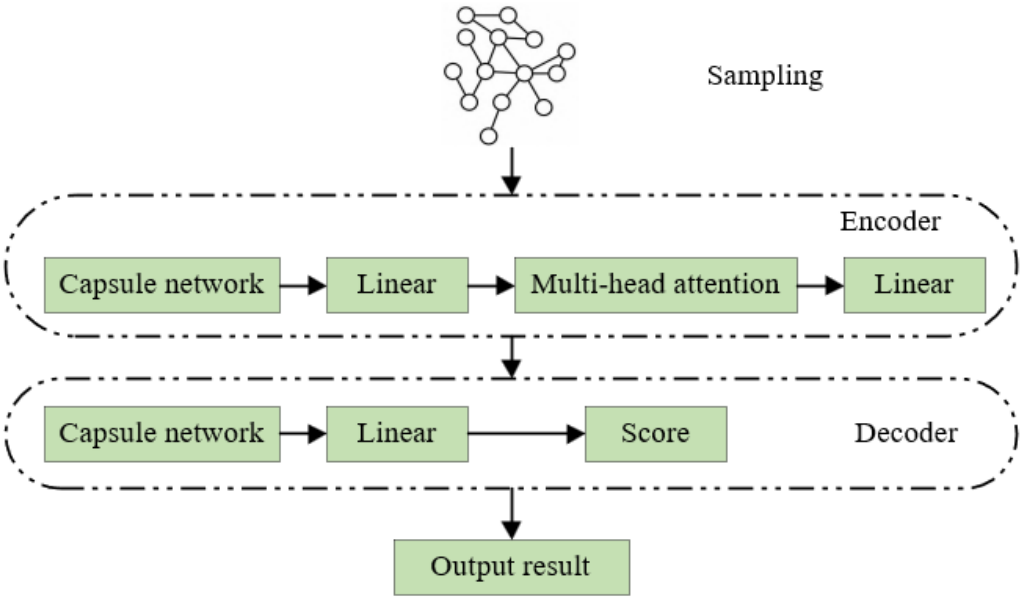

Fig. 1. Proposed model flowchart

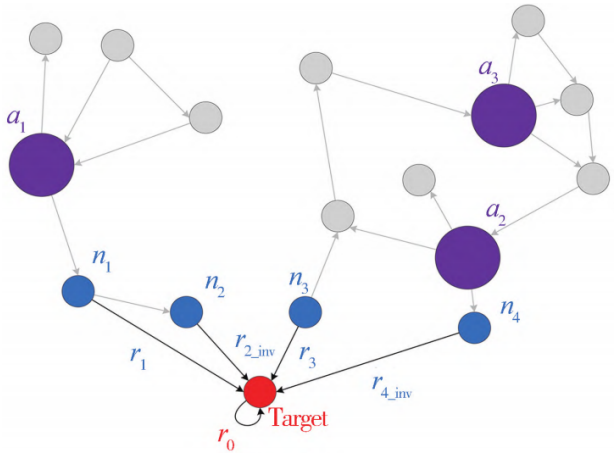

Fig. 2. Subgraph sampling

$$V_t = W_s \odot t_s + W_d \odot t_d \tag{9}$$

We input the fused triplet vector as matrix $A = (V_h, V_r, V_t) \in R^{k\times 3}$ into the capsule neural network model. Where A_i represents the $i-th$ of matrix A. k is the dimension of the word vector, which is 100. The capsule neural network model is divided into two parts: the convolutional layer and the capsule layer.

2.2.1. *Convolutional layer*

Matrix A is taken as input to the convolution layer and three different windows are taken with sizes $j \in (1,2,3)$. Each window uses two convolution kernels ω_j to convolve the matrix A, and for each convolution kernels, the operation is performed by Eqs. (10) to (12).

$$\mu_{j,i} = \xi(\omega_i \cdot A_i, : +b) = ReLU(\omega_j \cdot A_i, : +b) \tag{10}$$

$$q_j = [\mu_{j1}, \mu_{j2}, \cdots, \mu_{jk}] \in R^k \tag{11}$$

$$q = [q_1, q_2, q_3] \tag{12}$$

Where $\cdot$ is the dot product operation and $b \in R$ is the bias term. ξ is the activation function ReLU. $\mu_{j,i}$ represents the characteristic element of the convolution kernel ω_j at each position of step 1 on row i of matrix A. q_j is the feature graph obtained under convolution operations of different windows. q is the set of feature graphs obtained by the convolution operation. Triplet vectors passing through the convolution layer produce six feature maps as inputs to the first capsule layer.

2.2.2. *Capsule layer*

Two capsule layers are used in the capsule model, and the first layer constructs k capsules for each feature map list. We encapsulate features of the same dimension in the feature graph set to capture features at different locations in the triplet vector. For each capsule, the corresponding vector u_{ji} is generated, and then the vector u_{ji} is multiplied by the weight matrix W_{ji} to obtain the vector U_{ji}, multiplied by the coupling coefficient c_i, weighted sum of the vector U_{ji} to obtain the input vector s_j of the second layer capsule. Finally, the nonlinear compression function $Squash()$ is executed on the vector s_j to produce an output vector e_j. Vectors e_1, e_2, e_3 are weighted and summed to obtain vector e, its length represents the fraction of the triplet. The above

process is shown in Eqs. (13) to (15).

$$e_j = Squash(s_j), s_j = \sum_i c_i U_{ji} = \sum_i c_i W_{ji} u_{ji} \tag{13}$$

$$Squash(s_j) = \frac{||s_i||^2}{1+||s_i||^2} \frac{s_i}{||s_i||} \tag{14}$$

$$e = \sum_{j=1,2,3} e_j \tag{15}$$

2.3. Encoder

Different from Nodepiece and StarGraph, which directly transfer nodes on subgraphs to the graph neural network model after generating subgraphs, the proposed method further optimizes the propagation process [33–35]. For neighbor nodes and relationships that express unique information, two-dimensional convolution with more expressive power is used for calculation, and the expression is as follows:

$$f_{conv}(N,R) = FC(vec(f_w(cat[\bar{e}_N, \bar{e}_R]))) \tag{16}$$

For anchor node embedding and anchor node distance embedding expressing general information, it directly adds them together, then,

$$f_{add}(A,D) = e_A + e_D \tag{17}$$

In order to obtain an embedded representation with strong differentiation and prevent over-fitting, the embedding dimension of neighbor nodes is reduced to reduce the unique information capacity, and the embedding dimension of anchor nodes is increased to increase the general information capacity. Before entering Transformer encoder, the neighbor node embed is mapped to the same dimension as the anchor node embed through linear transformation, and the node type embed information is added to distinguish itself, neighbor and anchor node three node types.

In this case, the set of anchor nodes and neighbor nodes on the subgraph can be represented as:

$$cat[f_{conv}(N,R); f_{add}(A,D)] \in R^{(1+k+m)\times d} \tag{18}$$

For information aggregation on subgraphs, the self-attention module of Transformer is used to calculate the target node's attention to itself and other nodes providing information on subgraphs as a "virtual edge".

$$attn = Attention(cat[f_{conv}(N,R); f_{add}(A,D)]) \in R^{(1+k+m)\times d} \tag{19}$$

The subgraph then becomes a "fully connected subgraph" and has a manageable scale to ensure feasible computational efficiency. The entire subgraph is computed by the Transformer encoder, and all node information can be fully and effectively mixed in the end, i.e.,

$$encode(T) = Transformer(cat[f_{conv}(N,R); f_{add}(A,D)]) \in R^{(1+k+m)\times d} \tag{20}$$

Single-layer Transformer block is adopted, that is, one attention layer plus two linear layers. After encoding in single-layer Transformer, the output subgraph nodes are embedded and the final target nodes are embedded in e_T by means of average aggregation.

2.4. Decoder

First, TransE, a scoring function based on translation distance, is adopted as the scoring function.

$$f_r(h,t) = -||h + r - t|| \tag{21}$$

$$score(h,r,t) = tanh(f_r(h,t)) + 1 \tag{22}$$

If the TransE score function is less than 0, the *tanh* activation function is used to constrain the score at $(-1,0)$, and 1 is added to constrain the score at $(0,1)$ as the prediction probability value.

The greater the probability value, the closer the head entity h of the knowledge triplet is to the tail entity t under the transformation of relation r, the higher the rationality of the knowledge triplet, that is, the higher the probability of predicting the existence of relation r between the head entity h and the tail entity t.

We use the inner product as a scoring function, then,

$$f_r(h,t) = ||f_{conv}(h,r) \times t^T|| \tag{23}$$

$$score(h,r,t) = \sigma(||f_{conv}(h,r) \times t^T||) \tag{24}$$

The sigmoid function is used to constrain the score at $(0,1)$ as the predicted probability value.

The greater the probability, the greater the probability value of the convolution result of the head entity h and relation r representing the knowledge triplet and the inner product of the tail entity t obtained by the sigmoid activation function. The higher the rationality of the knowledge triplet, the higher the probability of predicting the relationship r between the head entity h and the tail entity t.

Training with a $1-N$ model, unlike a $1-1$ model that calculates the score of a single knowledge triplet (h,r,t), the $1-N$ model computes the score of each head entity relation pair (h,r) and all entities $t \in N$ simultaneously. This method can train the model more efficiently and reduce the

hardware load [27]. Using the cross entropy loss function as the training target, there are,

$$L(score, label) = -\frac{1}{N}\sum_{i\in N}(label_i \times \log(score_i) + (1 - label_i) \times \log(1 - score_i)) \quad (25)$$

Where N represents the number of all entities. $label \in R^{|N|}$ indicates the label vector.

3. Results and discussion

In order to verify the effectiveness of the proposed method, this paper analyzes two tasks of link prediction and triplet classification on four data sets. All experiments are conducted based on TensorFlow framework, NVIDIA RTX 2060 GPU and Intel(R) i5-9400F CPU environment.

Five data sets, FB15k, FB15K-237, WN11, MCMK and FinKG, are used in this paper. Among them, the MCMK data set integrates the "Chinese Symptom Database", "Medical question and Answer Knowledge Map" in OpenKG and other traditional Chinese and Western medicine books and pharmacopoeies. The sentences in the data set contain a maximum of 1458 words with an average word number of 728, which is complex and multi-source, and the semantic difference of entity description is significant. The sentences in FB15k contain 343 words and the average number of words is 69. Fb15k-237 is a subset of FB15K, which reduces some repetitive relations and integrates some entities. FinKG is the most complex and diverse document in the financial industry. The report is usually written by researchers of financial research institutions, involving a large number of macro data, industry research data, industry research data and fundamental and technical data of individual stocks/companies, with comprehensive information collection, in-depth research, high quality, reliable content and other characteristics. The core parameters of the four data sets are shown in Table 1.

In order to fully compare the performance of the proposed methods, the TEKE model, Text-Graph model, TransE model, TransH model, TransR model, TransD model and TransA model are selected as the comparison algorithms. In the proposed method in this paper, Word2Vec is used to obtain word embedding v_{ij}. In the model training, the random gradient descent method is used to optimize the loss function. $L2$ regularization is used to prevent overfitting of the model. The experimental parameter settings are shown in Table 2.

3.1. Parameter sensitivity analysis

In proposed model, the training learning rate λ, boundary parameter μ, vector dimension d, loss function weight α and Batch size have direct effects on model performance. Therefore, this section analyzes the sensitivity of the above parameters according to the accuracy of triplet classification. For the three data sets FB15K, WN11 and MCMK, the learning rate λ, boundary parameter μ, vector dimension d, loss function weight α and Batch Size respectively adopt the value ranges listed in Table 2, and the obtained classification results are shown in Figs. 3 to 5.

It can be seen from the experimental results that the boundary parameter μ is insensitive to the data set, and the best triplet classification accuracy is obtained when $\mu = 1.0$. For different data sets, the parameters and corresponding training time for obtaining the best classification performance are shown as follows. FB15K and FB15K-237 data sets: $\lambda = 0.001$, $d = 100$, $\alpha = 0.4$, $BatchSize = 128$; The training time is 138min and 102 min. Data set WN11: $\lambda = 0.001$, $d = 200$, $\alpha = 0.6$, $BatchSize = 128$; The training time is 192min. MCMK data set: $\lambda = 0.01$, $d = 200$, $\alpha = 0.6$, $BatchSize = 256$; The training time is 275min.

3.2. Link prediction

Link prediction is used to predict the missing relationship between two entities, i.e., to predict triples (h, r, t). In this paper, two evaluation indexes mean rank (MR), mean reciprocal ranking (MRR) and Hits@10 are used to evaluate the accuracy of link prediction. A lower MR Value or a higher MRR/Hits@10 means that the model has better link prediction performance.

The average ranking represents the average ranking of triples to obtain the correct relationship. Recording T as the set of triples and $|T|$ as the size of the set of triples. k_i indicates the link prediction ranking of the $i-th$ triplet, i.e.,

$$MR = \frac{1}{|T|}\sum_{i=1}^{|T|} k_i \quad (26)$$

MRR is calculated as follows. Where S is the set of triples, $|S|$ is the number of triples, and $rank_i$ refers to the link prediction ranking of the $i-th$ triple.

$$MRR = \frac{1}{|S|}\sum_{i=1}^{|S|}\frac{1}{rank_i} \quad (27)$$

Hits@10 represents the proportion of triples that obtained the correct relationship in the first 10 predictions. It defines $II(\cdot)$ as the indicator function, i.e.,

$$Hits@10 = \frac{1}{|T|}\sum_{i=1}^{|T|} II(k_i \leq 10) \quad (28)$$

Table 3 shows the link prediction results of all comparison algorithms, and the optimal prediction results

Table 1. Parameters of the data sets

Data set	Entity	Relation	Training set	Validation set	Test set
FB15k	14953	1347	483142	50000	59071
FB15K-237	14542	238	272117	17535	20467
WN11	38696	11	112582	2609	10543
MCMK	32187	24	12692	2716	4729
FinKG	12668	20707	35687	46127	6574

Table 2. Experimental parameters

Learning rate λ	0.0001, 0.001, 0.01, 0.1
Layer number in Transformer	8
Number of self-attention mechanism heads in Transformer	12
Boundary parameter μ	0.5, 1.0, 1.5, 2.0
Batch Size	64, 128, 256, 512
Dropout	0.6
Vector dimension d of entities and relations	50, 100, 200
Loss function weight α	0.2, 0.4, 0.6, 0.8

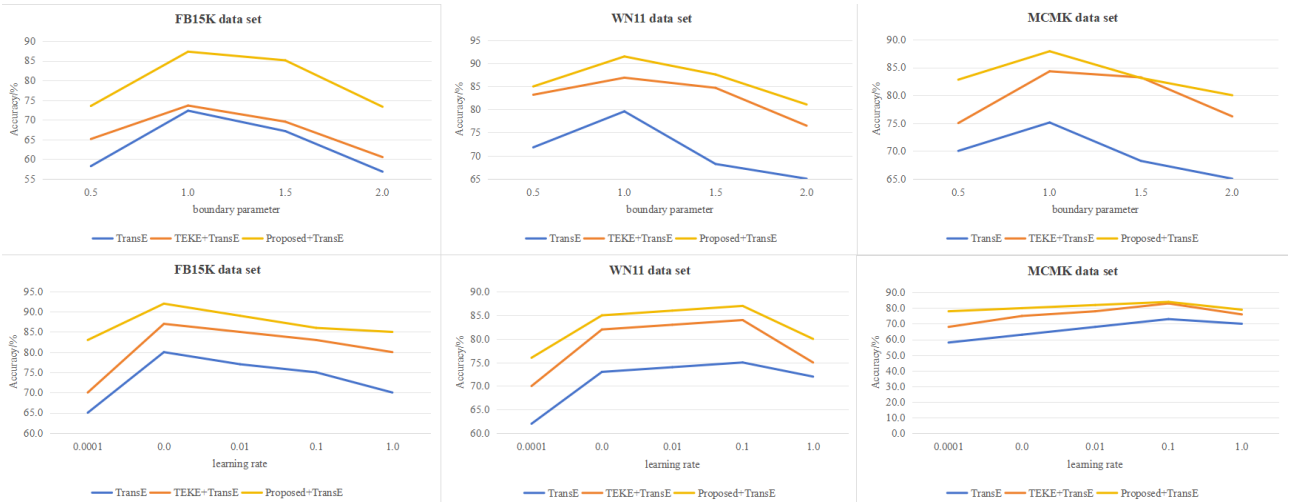

Fig. 3. The effect of boundary parameters and learning rate on algorithm performance

are shown in bold. Among them, proposed+TransE, proposed+TransH and proposed+TransR represent the combination of the multi-layer attention mechanism proposed in this paper with the corresponding structural embedding model respectively. ESKA+TransE, ESKA+TransH, and ESKA+TransR are based on the above methods by removing multiple layers of attention mechanisms. In addition, in order to make a more intuitive comparison between the experimental results of the proposed model and the comparison model, based on three basic models, TransE, TransH and TransR, Hits@10 and MR of all comparison methods are shown in Fig. 6.

It can be seen from the link prediction experiment results of the proposed model that the proposed model has the smallest MR Index and the largest Hits@10 on the three data sets. In particular, for the FB15K dataset, the prediction result of the proposed+TransE model is more accurate than that of the TransE model. The evaluation index MR is reduced by 45% and Hits@10 is increased by 42%, indicating that semantic information in entity description can effectively improve the quality of knowledge representation learning. In addition, the prediction result of the proposed+TransE model is more accurate than that of the TEKE+TransE model. The evaluation index MR is reduced by 13% and Hits@10 is increased by 18%, indicating that the proposed model has better performance than the knowledge representation learning model combined with entity description. For FB15K-237 and MCMK data sets, Hits@10 of the proposed+TransR model is 8% and 37% higher than TEKE+TransR model, respectively. It can be seen that the

Fig. 4. The effect of vector dimension and Batch size on algorithm performance

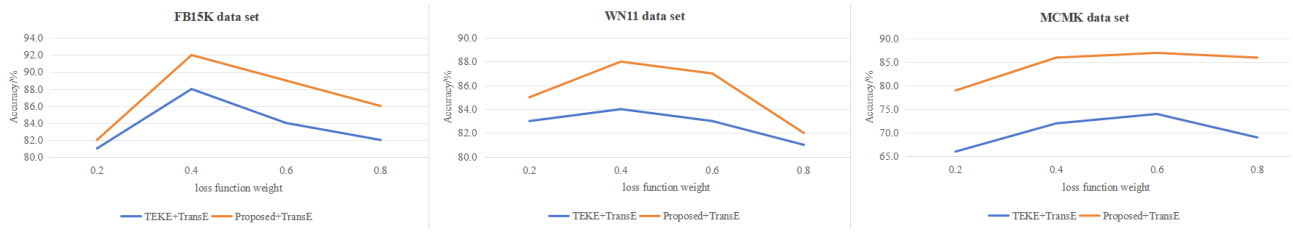

Fig. 5. The effect of loss function weight on algorithm performance

Table 3. Experimental results of link prediction

Data	FB15k	FB15k	FB15k	FB15K-237	FB15K-237	FB15K-237	MCMK	MCMK	MCMK
Model	MR	Hits@10	MRR	MR	Hits@10	MRR	MR	Hits@10	MRR
TransD	91	77.3	26.7	244	48.0	25.4	462	35.0	24.9
TransH	84	58.5	27.5	348	45.2	27.1	451	42.7	28.5
TransA	74	80.4	30.4	451	49.1	31.2	437	38.0	31.1
TransR	78	65.5	32.4	252	47.8	31.5	467	43.6	32.2
TransE	125	47.1	33.7	347	46.5	32.8	412	38.6	33.9
TEKE+TransE	79	67.6	35.6	171	48.9	36.4	367	45.0	35.8
Text-Graph+TransE	72	76.3	35.9	169	48.3	37.2	287	46.5	37.4
ESKA+TransE	75	70.8	36.7	174	48.5	38.5	374	42.8	38.2
Proposed+TransE	69	**82.5**	**50.3**	**151**	51.6	**51.4**	**214**	56.3	**52.7**
TEKE+TransH	75	70.4	49.7	259	48.8	48.3	386	62.0	49.1
ESKA+TransH	77	76.0	48.1	312	43.7	47.9	423	54	49.2
Proposed+TransH	70	**82.5**	47.2	260	52.7	46.3	274	**85.0**	47.4
TEKE+TransR	79	68.5	46.5	233	48.1	47.0	365	59.8	47.3
ESKA+TransR	74	73.4	46.1	217	45.7	40.8	437	40.5	39.6
Proposed+TransR	**68**	81.0	39.4	187	**54.5**	38.7	292	65.4	36.5

proposed model is better suited to handle data sets that contain complex entity descriptions.

By comparing and analyzing the link prediction experiment results of ESKA model, it can see that the prediction accuracy of ESKA+TransH model is worse than that of the proposed+TransH model on the three data sets, indicating that the multi-layer attention mechanism can better obtain the semantic information contained in the entity description and combine it with structural embeddedness. In particular, HITS@10 for FB15k-237 and ESKA+TransE models increases by 12% compared to TransR; For MCMK dataset, HITS@10 decreases by 7% compared with TransR model,

Fig. 6. Hits@10 and MR values for different base models

which indicates that the lack of multi-layer attention mechanism will reduce the quality of knowledge representation learning.

3.3. Triplet classification

Triplet classification is a typical binary classification task designed to determine whether a given triplet (h, r, t) is correct. For this classification task, a specific relational threshold σ is set in this paper. If the distance score of the triplet (h, r, t) is less than the threshold σ, the triplet is considered correct; Otherwise, it is wrong. In the experiment, the threshold to achieve maximum classification accuracy on the verification set was selected, and the other parameters were selected with the same values as the link prediction task. The accuracy of triplet classification was evaluated by obtaining the triplet proportion of the correct relationship [36].

Table 4 shows the triplet classification results of all comparison algorithms, and the optimal classification results are shown in bold. Obviously, the proposed model has the highest triplet classification accuracy on all three data sets. In particular, for the WN11 data set, the accuracy of the proposed+TransH model is 9.7% higher than the TransH model and 2.9% higher than the TEKE+TransH model. In addition, the accuracy of the proposed model is higher than that of the ESKA model. For example, the proposed+TransR model's triple classification accuracy on FB15k data sets is 3.8% higher than that of ESKA+TransE model, indicating that multi-layer attention mechanism plays an important role in knowledge representation learning. In addition, TEKE, ESKA and proposed methods combined with entity description generally have higher triplet classification accuracy than TransE and other models using only structural information on the three data sets, and the improvement is most obvious on MCMK data sets, which proves that entity description can effectively improve the quality of knowledge representation learning.

Table 4. Experimental results of link prediction

Model	WN11	FB15K	MCMK	FinKG
TransE	75.9	79.8	72.4	73.5
TransH	78.8	87.7	74.3	77.4
TransR	85.9	83.9	82.6	82.9
TransD	86.4	88.0	85.7	87.9
TEKE+TransE	84.1	87.9	74.1	86.7
ESKA+TransE	82.4	88.4	71.9	81.1
Proposed+TransE	87.5	91.2	**87.3**	85.8
TEKE+TransH	84.8	89.2	78.4	88.3
ESKA+TransH	82.7	88.4	75.8	83.5
Proposed+TransH	**88.5**	**92.5**	86.9	**90.8**
TEKE+TransR	86.1	85.6	84.5	88.6
ESKA+TransR	85.1	82.7	80.7	87.8
Proposed+TransR	88.4	89.1	87.1	86.7

3.4. Ablation experiment

In order to illustrate the influence of different modules on the proposed method, we conducted comparative experiments on the FB15K-237 dataset, and the results are shown in Table 5. From Table 5, the numerical results of the module with the attention mechanism have a significant advantage over the baseline. The MR Value is 15 lower than baseline, and the Hits@10 value is 19.1 higher than baseline. However, it is obvious that there is still a gap compared with the proposed method in this paper, because the key feature extraction of capsule network is lost.

Table 5. Ablation experiment with different module

Model	MR	Hits@10	MRR
Baseline	87	63.4	38.7
Baseline+capsule network	75	79.7	40.6
Baseline+Multi-head attention	72	81.4	42.7
Proposed	**68**	**82.5**	**51.3**

4. Conclusions

Knowledge graph contains a wide range of entities, and the corresponding entity description contains rich semantic information, which can be used to improve the quality of knowledge representation learning. However, these semantic information are difficult to obtain accurately and combine with knowledge representation learning. Therefore, a novel knowledge graph representation learning model based on capsule network and information fusion is proposed in this paper, which adopts the multi-layer attention mechanism and uses the structural embeddings of entities to enhance the semantic expression in the entity description. Then, the Transformer model integrates the semantic relations of complex entity descriptions and uses the semantic relations to enhance the relational embeddings. Faced with four data sets, the new model performs better in link prediction and triplet classification tasks than the only considering structural information models (TransE, TransH, TransR, TransD, and TransA), as well as the latest knowledge representation learning models (TEKE and Text-Graph). In the future, we will try to optimize the knowledge graph representation of low-resource languages, further expand the scale of knowledge base, supplement the performance of data test models of other languages, and try to introduce more external auxiliary information to achieve multi-modal embedding, so as to enhance the learning effect of knowledge representation.

5. Acknowledgments.

This work was supported by the Special Fund of Basic scientific Research Business expenses of undergraduate universities in Liaoning Province. Project name: Application of a large language model for enhancing career ability map driven by knowledge base in education and teaching scenarios. Project number: LJ232410166062.

References

[1] Z. Chen, Y. Wang, B. Zhao, J. Cheng, X. Zhao, and Z. Duan, (2020) *"Knowledge graph completion: A review"* **Ieee Access** *8*: 192435–192456. DOI: 10.1109/ACCESS.2020.3030076.

[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. "Dbpedia: A nucleus for a web of open data". In: *international semantic web conference*. Springer. 2007, 722–735. DOI: 10.1007/978-3-540-76298-0_52.

[3] A. Madkour, W. G. Aref, and S. Basalamah. "Knowledge cubes—A proposal for scalable and semantically-guided management of Big Data". In: *2013 IEEE International Conference on Big Data*. IEEE. 2013, 1–7. DOI: 10.1109/BigData.2013.6691800.

[4] F. M. Suchanek, G. Kasneci, and G. Weikum, (2008) *"Yago: A large ontology from wikipedia and wordnet"* **Journal of Web Semantics** ***6***(3): 203–217. DOI: 10.1016/j.websem.2008.06.001.

[5] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun. "Representation learning of knowledge graphs with entity descriptions". In: *Proceedings of the AAAI conference on artificial intelligence*. ***30***. 1. 2016. DOI: 10.1609/aaai.v30i1.10329.

[6] H. Zhu, D. Xu, Y. Huang, Z. Jin, W. Ding, J. Tong, and G. Chong, (2024) *"Graph structure enhanced pre-training language model for knowledge graph completion"* **IEEE Transactions on Emerging Topics in Computational Intelligence**: DOI: 10.1109/TETCI.2024.3372442.

[7] S. Feng, C. Zhou, Q. Liu, X. Ji, and M. Huang, (2024) *"Temporal Knowledge Graph Reasoning Based on Entity Relationship Similarity Perception"* **Electronics** ***13***(12): 2417. DOI: 10.3390/electronics13122417.

[8] H. Yang and J. Liu. "Knowledge graph representation learning as groupoid: unifying TransE, RotatE, QuatE, ComplEx". In: *Proceedings of the 30th ACM international conference on information & knowledge management*. 2021, 2311–2320. DOI: 10.1145/3459637.3482442.

[9] C. Jin, R. Cui, and Y. Zhao. "Research on Chinese-Korean Entity Alignment Method Combining TransH and GAT". In: *China Conference on Knowledge Graph and Semantic Computing*. Springer. 2021, 134–144. DOI: 10.1007/978-981-16-6471-7_10.

[10] F. Liu, Y. Shen, T. Zhang, and H. Gao, (2020) *"Entity-related paths modeling for knowledge base completion"* **Frontiers of Computer Science** ***14***: 1–10. DOI: 10.1007/s11704-019-8264-4.

[11] M. Zhang and Y. Chen, (2018) *"Link prediction based on graph neural networks"* **Advances in neural information processing systems** ***31***: DOI: 10.1007/978-981-16-6054-2_10.

[12] H. Xiao, M. Huang, Y. Hao, and X. Zhu, (2015) *"TransG: A generative mixture model for knowledge graph embedding"* **arXiv preprint arXiv:1509.05488**: DOI: 10.48550/arXiv.1509.05488.

[13] X. Tang, L. Chen, J. Cui, and B. Wei, (2019) *"Knowledge representation learning with entity descriptions, hierarchical types, and textual relations"* **Information Processing & Management** ***56***(3): 809–822. DOI: 10.1016/j.ipm.2019.01.005.

[14] B. An, B. Chen, X. Han, and L. Sun. "Accurate text-enhanced knowledge graph representation learning". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 2018, 745–755. DOI: 10.18653/v1/N18-1068.

[15] Y. Zhao, H. Feng, H. Zhou, Y. Yang, X. Chen, R. Xie, F. Zhuang, and Q. Li, (2022) *"EIGAT: Incorporating global information in local attention for knowledge representation learning"* **Knowledge-Based Systems** ***237***: 107909. DOI: 10.1016/j.knosys.2021.107909.

[16] Z. Li, X. Liu, X. Wang, P. Liu, and Y. Shen, (2023) *"Transo: a knowledge-driven representation learning method with ontology information constraints"* **World Wide Web** ***26***(1): 297–319. DOI: 10.1007/s11280-022-01016-3.

[17] Z. Li, X. Jin, W. Li, S. Guan, J. Guo, H. Shen, Y. Wang, and X. Cheng. "Temporal knowledge graph reasoning based on evolutional representation learning". In: *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 2021, 408–417. DOI: 10.1145/3404835.3462963.

[18] J. Hao, M. Chen, W. Yu, Y. Sun, and W. Wang. "Universal representation learning of knowledge bases by jointly embedding instances and ontological concepts". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, 1709–1719. DOI: 10.1145/3292500.3330838.

[19] H. Mousselly-Sergieh, T. Botschen, I. Gurevych, and S. Roth. "A multimodal translation-based approach for knowledge graph representation learning". In: *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*. 2018, 225–234. DOI: 10.18653/v1/S18-2027.

[20] N. Passalis, M. Tzelepi, and A. Tefas, (2020) *"Probabilistic knowledge transfer for lightweight deep representation learning"* **IEEE Transactions on Neural Networks and learning systems** ***32***(5): 2030–2039. DOI: 10.1109/TNNLS.2020.2995884.

[21] T. Chen, L. Lin, R. Chen, Y. Wu, and X. Luo, (2018) *"Knowledge-embedded representation learning for fine-grained image recognition"* **arXiv preprint arXiv:1807.00505**: DOI: 10.48550/arXiv.1807.00505.

[22] W. Gan, Y. Sun, and Y. Sun, (2022) *"Knowledge structure enhanced graph representation learning model for attentive knowledge tracing"* **International Journal of Intelligent Systems** ***37***(3): 2012–2045. DOI: 10.1002/int.22763.

[23] B. Wang, T. Shen, G. Long, T. Zhou, Y. Wang, and Y. Chang. "Structure-augmented text representation learning for efficient knowledge graph completion". In: *Proceedings of the Web Conference 2021*. 2021, 1737–1748. DOI: 10.1145/3442381.3450043.

[24] C. Zhao, J. Jiang, Y. Guan, X. Guo, and B. He, (2018) *"EMR-based medical knowledge representation and inference via Markov random fields and distributed representation learning"* **Artificial intelligence in medicine** ***87***: 49–59. DOI: 10.1016/j.artmed.2018.03.005.

[25] J. Zhang, S. Liang, Y. Sheng, and J. Shao, (2022) *"Temporal knowledge graph representation learning with local and global evolutions"* **Knowledge-Based Systems** ***251***: 109234. DOI: 10.1016/j.knosys.2022.109234.

[26] G. Niu, Y. Zhang, B. Li, P. Cui, S. Liu, J. Li, and X. Zhang. "Rule-guided compositional representation learning on knowledge graphs". In: *Proceedings of the AAAI conference on artificial intelligence*. ***34***. 03. 2020, 2950–2958. DOI: 10.1609/aaai.v34i03.5687.

[27] H. Liu, C. Li, Y. Li, and Y. J. Lee. "Improved baselines with visual instruction tuning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, 26296–26306. DOI: 10.1109/CVPR52733.2024.02484.

[28] T. Shen, Y. Mao, P. He, G. Long, A. Trischler, and W. Chen, (2020) *"Exploiting structured knowledge in text via graph-guided representation learning"* **arXiv preprint arXiv:2004.14224**: DOI: 10.48550/arXiv.2004.14224.

[29] S. Seo, B. Oh, and K.-H. Lee, (2020) *"Reliable knowledge graph path representation learning"* **IEEE Access** ***8***: 32816–32825. DOI: 10.1109/ACCESS.2020.2973923.

[30] M. Fan, Q. Zhou, T. F. Zheng, and R. Grishman, (2017) *"Distributed representation learning for knowledge graphs with entity descriptions"* **Pattern Recognition Letters** ***93***: 31–37. DOI: 10.1016/j.patrec.2016.09.005.

[31] P. Wang, S. Li, and R. Pan. "Incorporating gan for negative sampling in knowledge representation learning". In: *Proceedings of the AAAI conference on artificial intelligence*. ***32***. 1. 2018. DOI: 10.1609/aaai.v32i1.11536.

[32] Z. Feng, D. Tang, X. Feng, C. Zhou, J. Liao, S. Wu, B. Qin, Y. Cao, and S. Shi, (2024) *"Pretraining without wordpieces: learning over a vocabulary of millions of words"* **International Journal of Machine Learning and Cybernetics *15***(9): 3989–3998. DOI: 10.1007/s13042-024-02132-4.

[33] Y. Jiang and S. Yin, (2023) *"Heterogenous-view occluded expression data recognition based on cycle-consistent adversarial network and K-SVD dictionary learning under intelligent cooperative robot environment"* **Computer Science and Information Systems *20***(4): 1869–1883. DOI: 10.2298/CSIS221228034J.

[34] Z. Zhang, L. Cao, X. Chen, W. Tang, Z. Xu, and Y. Meng, (2020) *"Representation learning of knowledge graphs with entity attributes"* **IEEE Access *8***: 7435–7441. DOI: 10.1109/ACCESS.2020.2963990.

[35] S. Yin, H. Li, A. A. Laghari, T. R. Gadekallu, G. A. Sampedro, and A. Almadhor, (2024) *"An Anomaly Detection Model Based on Deep Auto-Encoder and Capsule Graph Convolution via Sparrow Search Algorithm in 6G Internet of Everything"* **IEEE Internet of Things Journal *11***(18): 29402–29411. DOI: 10.1109/JIOT.2024.3353337.

[36] Z. Li, Q. Zhang, F. Zhu, D. Li, C. Zheng, and Y. Zhang, (2023) *"Knowledge graph representation learning with simplifying hierarchical feature propagation"* **Information Processing & Management *60***(4): 103348. DOI: 10.1016/j.ipm.2023.103348.