Computing and Informatics, Vol. 42, 2023, 257–279, doi: 10.31577/cai_2023_2_257

ASIAM-HGNN: AUTOMATIC SELECTION AND INTERPRETABLE AGGREGATION OF META-PATH INSTANCES FOR HETEROGENEOUS GRAPH NEURAL NETWORK

Xiaojun Lou

Department of School of Mathematics and Computer
Zhejiang A & F University
Hangzhou 311300, China
e-mail: 1932707397@qq.com

Guanjun Liu

Department of Computer Science
Key Laboratory of Embedded System and Service Computing
Tongji University
Shanghai 201804, China
e-mail: liuguanjun@tongji.edu.cn

Jian Li*

Department of School of Mathematics and Computer
Zhejiang A & F University
Hangzhou 311300, China
e-mail: lijian0120@foxmail.com

Abstract. In heterogeneous information network (HIN)-based applications, the existing methods usually use Heterogeneous Graph Neural Networks (HGNN) to handle some complex tasks. However, these methods still have some shortcomings: 1) they manually pre-select some meta-paths and thus some important ones are

* Corresponding author

missing, while the missing ones still contains the information and features of the node in the entire graph structure; and 2) they have no high interpretability since they do not consider the logical sequences in an HIN. In order to deal with them, we propose ASIAM-HGNN: a heterogeneous graph neural network combined with the automatic selection and interpretable aggregation of meta-path instances. Our model can automatically filter important meta paths for each node, while preserving the logical sequence between nodes, so as to solve the problems existing in other models. A group of experiments are conducted on real-world datasets, and the results demonstrate that the models learned by our method have a better performance in most of task scenarios.

Keywords: Graph neural networks, meta-paths, network representation learning, heterogeneous graph

1 INTRODUCTION

A Heterogeneous Information Network (HIN) defines a group of entities and their relations, and this heterogeneous representation can describe the real world more precisely compared to those homogeneous graphs. The emergence of HIN has triggered new explorations in many application scenarios such as relationship prediction [1, 2], recommendation [3, 4] and node classification [5].

However, because manual feature selection methods are generally used in these scenarios, it is hard for them to precisely express the characteristics of nodes. Although, some methods are proposed which can automatically mine node representation from a graph structure, such as Deepwalk [6], Metapath2vec [7] and ASNE [8], their ability is still limited when rich neighborhood information is expected to capture [9]. To solve this problem, some graph neural network methods are presented, such as: GCN [10], GAT [11] and GraphSAGE [12]. They can make good use of the feature feedback of adjacent nodes, and the aggregation embedding function is more powerful. However, they aim at homogenous graphs rather than heterogeneous ones. Consequently, they cannot distinguish the difference of node and edge attributes so that they cannot obtain a good performance for heterogeneous graphs [13, 9].

For overcoming this defect, some heterogeneous graph neural networks have been developed which mainly fall into two categories. The first one employs meta-paths to split a heterogeneous graph structure, and then to represent the node embeddings based on the extracted meta-path instances, e.g. HAN [9], HAHE [14], DeepHGNN [15] and MAGNN [16]. Another one such as HetGNN [13] does not use meta-path, but they usually apply the random walk algorithm to dig out the node embedding. However, they still have some problems. In the first category, they artificially preselect the types of meta-paths which influence the effect of the trained model, but this process is not interpretable because of the high subjectivity of manual selection. Additionally, they are inevitable to ignore the influence of unselected meta-paths of

each node. In the second category, HetGNN [13] disorders the nodes selected by the random walk, and inputs them into a neural network. Obviously, it cannot maintain the logical sequence in the graph structure so that it does not have high interpretability.

Focusing on these problems, we propose ASIAM-HGNN, a method of Automatic Selection and Interpretable Aggregation of Meta-path instances used in Heterogeneous Graph Neural Network. ASIAM-HGNN can adaptively find "strongly correlated" meta-paths corresponding to nodes, and at the same time preserve the sequence relationship of the graph structure in the process of aggregating meta-paths, which solves:

1. The subjectivity caused by the artificial selection of meta-paths in existing models and Loss of information;
2. Low interpretability due to the loss of the original structure of the graph after random walk.

Specifically, we wander from each node by the random walk, and then get k "end-to-end" (the types of nodes at both ends of meta-path instances are the same) meta-path instances with the highest frequency from each node. In this way, we can collect the "strongly associated" meta-path instances for each node while preserving the connectivity and heterogeneity of a HIN. Meanwhile, our method can find out them automatically instead of manually. Additionally, the structural information retained in the meta-path is also saved in the embedded information when it passes through bi-LSTM, which solves the previous problem of weak interpretation. At last, k embedded path instances are aggregated through the attention layer which can make our model learn the influence factors of different meta-path instances on the target node, and help us express the accurate expression of the node in the graph structure. In the process of training, according to the triples corresponding to various types of nodes, our loss value calculation is based on all types of nodes, so the parameters optimized by the loss function will better reflect the real network situation. Through a group of experiments with real-world data, ASIAM-HGNN has a good effect in most task scenarios.

Our contributions can be summarized as follows:

- We employ a random walk-based approach to automatically search for "strongly correlated" meta-path instances. Our method can automatically and inclusively select all strongly associated meta-path instances around the node. Compared with artificial meta-path selection, our method is more objective and scientific, and will not cause information loss caused by artificial selection of meta-paths.
- We convert the node and meta-path instances of HIN into words and sentences respectively, and finally learn their embedding representations via bi-LSTM. Through the objective relationship sequence of "meta-path" and the learning of these sequence by bi-LSTM, we can preserve the original structure of graph data and the process of information transfer to the greatest extent on the basis of random walk, so as to solve the problem of interpretability low problem.

- We apply the attention mechanism to learn the attention coefficients of different nodes for different meta-path instances, and aggregate them together. In this process, multiple meta-path instances corresponding to the target node (including different meta path instances under the same meta path and meta path instances under different meta paths) are aggregated together. We learn the different effects of different meta-path instances corresponding to each node through the attention mechanism. Through this method, we try to find a more accurate representation for each node, and experiments prove that our model has a great performance.

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 introduces some key definitions we used in this paper. Section 4 shows our Heterogeneous Graph Neural Network model in details. Section 5 demonstrates our model's performance and compares our model with the baselines. Section 6 indicates the result of Ablation Study. Section 7 summarizes our work.

2 RELATED WORK

The related work including 3 parts:

1. Graph Network Embedding,
2. Homogeneous GCNs,
3. Heterogeneous GCNs.

2.1 Graph Network Embedding

Graph network embedding is developed to extract the embedded information of nodes from the graph structure, so as to use this information for downstream tasks. For example, there are methods based on random walk and deep learning network, but these methods are all aimed at homogeneous graph networks [9].

For heterogeneous graph network, many methods have also been proposed by researchers. ESim [17] is proposed to use the pre-selected meta-path to learn graph structure. EOE [18], PTE [19] and HEER [20] have processed heterogeneous graph into several bipartite graphs. Then, LINE [21] is employed to learn the representation by preserving the first-order or the second-order proximities. SHNE [22] and Metapath2Vec [7] both employ the improved Skip-Gram model to learn representation of the node after processing the graph structure. HERec [23] and MCRec [3] are recommending models for heterogeneous graph. HERec employs the method of type restriction to seize semantic information in heterogeneous graph structures, while MCRec adopts convolutional neural network (CNN) to get paths and then trains them in the form of constructing triples (user, item, meta-path). However, none of these models consider the influence of all meta-path instances on different nodes. Additionally, above-methods require artificially pre-selection of multiple meta-paths.

2.2 Homogeneous GCNs

Due to the achievements of CNN in the field of image processing, many researchers also apply the idea of convolution to graph structure, forming homogeneous graph convolutional neural network. Homogeneous graph convolution models are roughly divided into two categories. The first one is in the spectral domain, Bruna et al. [24] propose a graph convolution method based on graph Laplace transform for the first time. Chebnet [25] applies k-order Chebyshev Polynomials, and on this basis, Kipf et al. [10] employ localized first-order approximation to design a graph model called graph convolutional network, which also achieves good performance. The second one is the spatial approach, Veličković et al. propose GAT [11], they employ the attention mechanism to learn the influence coefficients, which is the weights between different nodes, so as to aggregate nodes. Hamilton et al. propose GraphSAGE [12], which samples neighbor nodes to aggregate the target. However, these models are aim at homogeneous network. In the face of complex node and edge relations, they cannot distinguish them. Therefore, these models have not received good results in dealing with the information of heterostructures [9].

2.3 Heterogeneous GCNs

In recent years, due to the poor performance of several traditional graph neural network models on heterogeneous graph, many heterogeneous graph neural network models have emerged. The methods, such as: HAN [9], HAHE [14], DeepHGNN [15] and GraphInception [5], decompose heterogeneous graphs into multiple homogeneous graphs through different type of meta-paths, then use homograph neural network to conduct convolution aggregation. MAGNN [16] first proposes the internal aggregation of meta-paths. It proposes an aggregator to aggregate the information of meta-paths, and then uses attention mechanism to perform node aggregation. HIN-DRL [26] proposes a method of dynamic acquisition of sequence through meta-path and timestamp, then use skip-gram to learn the sequence's representation. HetGNN [13] first employs random walk with restart to sample "strongly associated" nodes on the heterogeneous graph, then aggregated nodes of various types, and finally aggregates different types through the attention mechanism to obtain the final representation of nodes. RANCH [27] and HIN2Grid [28] both combine the graph attention network with the convolutional neural network for more accurate embedding learning. RGCN [29] first performs in-type aggregation, and then aggregates according to the type of the edge. MBRep [30] learns triangle motif embedding in the graph structure to get the representation of the nodes. [31] proposes a model to calculate the entropy between different meta-paths. What HetSANN [32] and HGT [33] do was to directly apply the GAT method to the heterogeneous graph structure, calculate the attention coefficient between each node, and then perform aggregation according to the attention coefficient. However, compared with our model [9, 14, 15, 5, 16, 26, 33, 31] needs to manually select meta-paths, which is prone to subjectivity, and may miss some meta-paths that also contain important

information. [13, 28, 27, 32] loses the objective sequence of the graph structure, which lead to low interpretability. [28, 30, 29, 32] does not take into account the structure of the meta-path, resulting in the information features contained in the meta-path being ignored.

3 PRELIMINARIES

We list three key definitions employed in our work for readability, which are from [34].

Definition 1 (HIN)**.** A heterogeneous information network (HIN) is a huge network with complex node types and relationships. A HIN is denoted as $G = (\mathcal{V}, \mathcal{E}, \phi, \psi, \mathcal{A}, \mathcal{R})$ where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ represents the feature set, $\mathcal{A}$ and $\mathcal{R}$ denote the sets of node and edge types such that $|\mathcal{A}| + |\mathcal{R}| > 2$, $\phi : \mathcal{V} \to \mathcal{A}$ and $\psi : \mathcal{E} \to \mathcal{R}$ are object type mapping function and link type mapping function.

Definition 2 (Meta-path)**.** A meta-path P is a path structure defined on the heterogeneous network and is represented in the form of $P_1 \xrightarrow{L_1} P_2 \xrightarrow{L_2} \ldots \xrightarrow{L_l} P_{l+1}$, which defines a relationship collection $L = L_1 \circ L_2 \circ \cdots \circ L_l$ between types P_1 and P_{l+1}, and intermediate nodes on the meta-path are connected by these relationships.

Definition 3 (Meta-path instance)**.** Given a graph G and a meta-path set M, each specific path instance $p \in M$ that conforms to meta-path structure is called meta-path instance.

4 MODEL ASIAM-HGNN

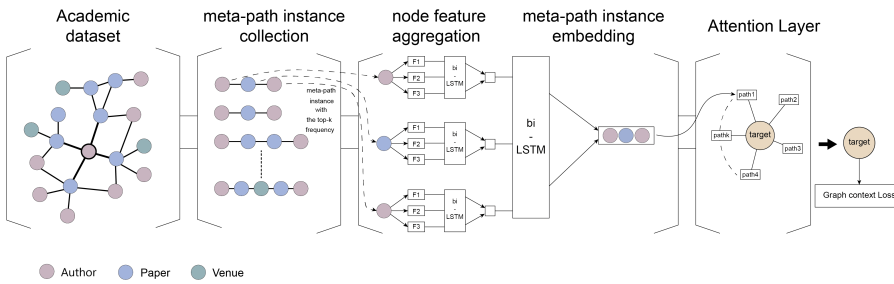


Figure 1. An overview of ASIAM-HGNN (Take academic data sets for example). ASIAM-HGNN consists of four parts: a) Meta-path instances collection, b) The aggregation of node features, c) Aggregation of the meta-path instances, d) Attention layer.

In this section, we will introduce four parts of our ASIAM-HGNN: Meta-path Instances Collection (collecting different meta-path instances for a group of target nodes and select the most correlated instance set), Node Features Aggregation

(aggregating different types of node features), Meta-path Instance Embedding (converting meta-path a instance to a vector representation), Attention Layer (learning the influence factors of different meta-paths on the target node and aggregating them).

4.1 Meta-Path Instances Collection

In a complicated heterogeneous network structure, each node has a large number of neighboring nodes and associated edges from which the node's embedded information come. For mining the structural information around each node, we design a random walk based method that can adaptively find the "strongly associated" meta-path instances for each node and can dig out the node embedded information hidden in the graph structure. Meanwhile, we do not need to pre-select multiple types of meta-paths. The algorithm mainly contains the following two steps:

Step 1: Collecting meta-path instances for each node. When we employ random walk to reach a node whose type is the same as the target node's, we record this meta-path instance and then continue to start a new random walk from the target node. In order to ensure that the collected meta-path instances can accurately express the spatial and structural information of the target node, we will repeat this step many times. For example, for the Academic-II dataset, we have to repeat this step 100 times for each node, which means that 100 meta-path instances are stored for each node.

Step 2: Selecting instances with higher relevance from the collected meta-path instances. For the meta-path instances of a node collected by step one, we will select the k path instances with the highest occurrence frequency as the "strongly associated" meta-path instances of the node.

This strategy can solve the aforementioned defects, because it has the following two characteristics:

1. Intermediate nodes will be stored during the wandering process, the information of this part will not be discarded, this characteristic preserves the order and structure of the network to the greatest extent, so that the structural information of the network can be better mined;
2. Each random walk starts from the target node, so that all the information around the target node in the graph structure can be obtained to the greatest extent.

So random walk method can automatically select the meta-path instances that are "strongly associated" with around the target node, and the selected meta-path instances of each node are different which also preserves the heterogeneity of the network structure, this characteristic ensures that the information mined by our method is more rich and heterogeneous compared to the existing models.

Figure 2 shows the process of selecting meta path instances for target nodes under academic dataset. From the figure, we can clearly see that through our

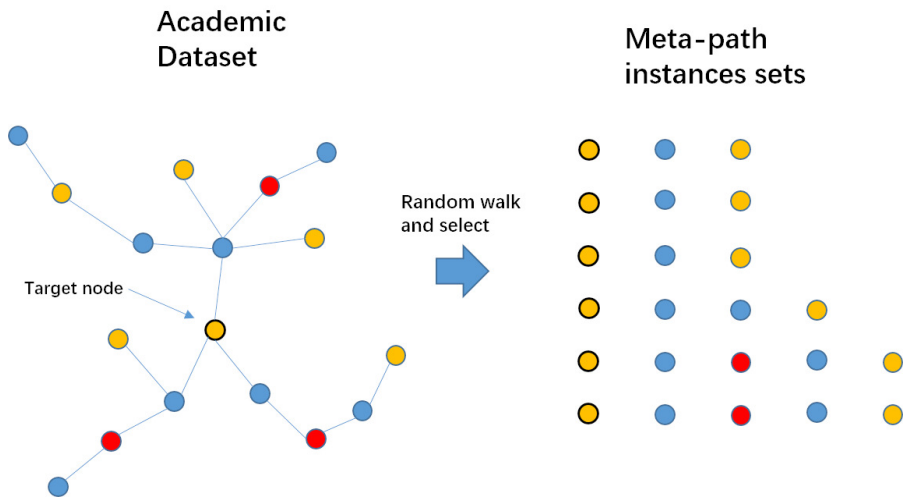


Figure 2. The example of meta-path instances selection for target nodes

selection of meta-path instances, most instances close to the target node are selected, including different meta-path and different instances under the same meta-path. In this way, we can find appropriate Metapath instances for each target node according to the original graph structure. Then we aggregate these meta-path instances in the same layer, which is helpful for our model to learn the different effects of different meta-path instances on each node.

4.2 Aggregation of Node Features

Given a meta-path instance p, it may contains different type of nodes, each type of node has different features and thus their dimensions are also different, which means that different type of nodes are in different semantic spaces. Therefore, we need some methods to aggregate their features and unify their dimensions. Here, we follow the method in the HetGNN [13].

Specifically, we first preprocess features of nodes in the network. For example, we use CNN to process image features, use ParVec [35] to process text features, change the feature dimensions to 128 dimensions, store these features in a three-dimensional matrix F, and then use the bi-LSTM network aggregating these features to obtain a node feature vector with a length dimension of 128 dimensions. This step can be understood as mapping the nodes with different lengths and different features to the same semantic spaces. The formula can be expressed as:

$$f(v) = \frac{\sum_{n \in C(n)} \left[\overrightarrow{LSTM} \left\{ f_1(n) \right\} \| \overleftarrow{LSTM} \left\{ f_1(n) \right\} \right]}{|C(n)|}, \tag{1}$$

where $C(n)$ denotes a set of node types, $f_1(n)$ represents the features of node n. The calculation process of $LSTM$ can be expressed as:

$$\begin{aligned} y_i &= \sigma\left(\mu_y f\left(x_i\right) + w_y h_{i-1} + b_y\right), \\ q_i &= \sigma\left(\mu_q f\left(x_i\right) + w_q h_{i-1} + b_q\right), \\ e_i &= \sigma\left(\mu_e f\left(x_i\right) + w_e h_{i-1} + b_e\right), \\ \hat{d}_i &= \tanh\left(\mu_d f\left(x_i\right) + w_d h_{i-1} + b_d\right), \\ d_i &= q_i \circ d_{i-1} + y_i \circ \hat{d}_i, \\ j_i &= \tanh\left(d_i\right) \circ e_i, \end{aligned} \tag{2}$$

where $\circ$ denotes the Hadamard product, y_i, q_i and e_i are the vectors of forget gate, b is the parameter that need to learn through the network, and j_i denotes the hidden output of the i^{th} unit Refering to HetGNN, this process can ensure a better effect on the feature aggregation of nodes.

4.3 Meta-Path Instance Embedding

Before dealing with meta-path instances, we must again emphasize the function of meta-path instances in this paper, which will help one understand our next work. We believe that the "end-to-end" meta-path instances in the heterogeneous network is responsible for most of the information connection and transmission. In the real world, the end-to-end information transfer method includes most of the information in the network, such as the relationship between authors in academic datasets, the relationship between articles, etc. At the same time, the methods in previous models can also prove this, for example, the HAN model directly discards the intermediate nodes of meta-path, only retains both ends' nodes, and finally achieves good performance. However, we find that information transfer in heterogeneous networks is bidirectional. For instance, in the Academic dataset, there is a meta-path instance $p(P^1 - V^1 - P^2)$ such that P^1 affects P^2 through V^1 to a certain extent, P^2 also affects P^1 in this way. We hope that this bidirectional information can also be reflected in the aggregation of meta-path instances. To this end, we choose bi-LSTM. This bidirectional LSTM network can train semantic information in two directions. Unlike HetGNN that inputs the LSTM after finding the nodes out of order, we retain the original structure of the heterogeneous network when inputting. The preservation of this natural structure makes our model more convincing.

The aggregation formula for the meta-path is:

$$\begin{aligned} &n_1, n_2, n_3 \in P, \\ &F = \text{concat}(f(n_1), f(n_2), f(n_3)), \\ &\textit{embedding} = \left[\overrightarrow{\text{LSTM}}\{F(i)\} \| \overleftarrow{\text{LSTM}}\{F(i)\}\right], \end{aligned} \tag{3}$$

where n_1, n_2, n_3 are the nodes that makes up the meta path p, $F(i)$ denotes the instance's vector which have concated different vector of nodes, and || denotes concatenation.

When using bi-LSTM, we understand a meta-path instance as a sentence with semantic meaning and the nodes as the words in the sentence. Different from the previous aggregation of node vectors, the previous step is to average the output of each unit. When aggregating the meta-path, since we want to get the representation of this path, we take the output of the last unit of the meta-path sequence in the bi-LSTM network as the embedding representation for that path.

4.4 Attention Layer

After getting the vector representation of each path, we also need to aggregate them to get the final representation of the target node. At this time, we must consider the imbalance of the heterogeneous network, which means different meta-path instances have totally different influence on the target node. We employ an attention layer to complete this step of aggregation operation. Among them, the feature aggregation of node $Node_v$ is expressed as:

$$\gamma_v = a^{v,v} f_1(v) + \sum_{x \in Path_v} a^{v,x} f_2^x(v), \tag{4}$$

where $a^{v,x}$ denotes the importance between v and x, $f_1(v)$ is the embedded representation of target node which is calculated in the Section 4.2, $f_2^x(v)$ is the embedded representation of meta-path instance which is aggregated in the Section 4.3.

The calculation process of the attention coefficient a between the node and the meta-path is:

$$a^{v,x} = \frac{\exp\left\{LeakyRelu\left(u^T\left[f_i \| f_1(x)\right]\right)\right\}}{\sum_{f_j \in Path} \exp\left\{LeakyRelu\left(u^T\left[f_j \| f_1(x)\right]\right)\right\}}, \tag{5}$$

where $Path$ represents the path instance set corresponding to the node, and u is a learnable parameter.

4.5 Loss Function

Since most of the data in many real-world data sets are not labeled, in view of this situation, we think that the unsupervised training method can better meet the needs of reality. We collect a certain number of triples by negative sampling [36]. Then, based on those triples, we can optimize the parameters and weights of the model according to the following loss function:

$$\mathcal{L} = \sum_{(n,n_v,n_{v'}) \in triples} \log \sigma\left(\mathcal{E}_{n_v} \cdot \mathcal{E}_n\right) + \log \sigma\left(-\mathcal{E}_{n_{v'}} \cdot \mathcal{E}_n\right), \tag{6}$$

where v denotes the target, n_v denotes the positive sample, $n_{v'}$ denotes the negative sample, σ represents the sigmoid function, $\mathcal{E}_n$ is the calculated embed of target node n.

Algorithm 1: Training process of ASIAM-HGNN

Input: pre-trained content features of nodes $n \in N$
trpiples sets $T(n, n_{pos}, n_{neg})$
meta-path instaces sets M for each nodes
Output: node embeddings

```
while not done do
    calculate mixed content features of nodes n ∈ N by Equation (1)
    learn embeddings of each meta-path instances in sets M by Equation (3)
    Aggregated mutiple meta-path instances through the attention layer to
     obtain the embedded representation of nodes by Equation (5)
    Compute loss by sending learned node embeddings and trpiples sets
     T(n, n_pos, n_neg) into Equation (6)
    update parameters by Adam
end
```

5 EXPERIMENT

We do experiments to evaluate the performance of ASIAM-HGNN on different tasks (link prediction and node classification).

5.1 Dataset

Academic: We follow the preprocessed method of HetGNN [13] to deal with the Academic Heterogeneous graph datasets from the Aminer database [37], and the detailed structure of the data is visible in Table 1.

DBLP: We used the DBLP dataset processed by [9], we use the BOW representations of keywords as the content features of the author nodes, and the detailed structure of the data is visible in Table 1.

IMDB: It is an online movie dataset, including three types of nodes: movie, actor and director, We use the BOW representations of plots as the content features of the movies, and the detailed structure of the data is visible in Table 1.

5.2 Baseline

We use six representative graph structure models as the baseline.

Dataset	Node Type	Edge Type
Academic-I	Paper:16073 Author:111409 Venue150	A–P P–P P–V
Academic-II	Paper:28646 Author:21044 Venue:18	A–P P–P P–V
DBLP	Paper:14328 Author:4057 Venue:20	A–P P–V
IMDB	Movie:3627 director:1714 actor:4340	M–D M–A

These are all Heterogeneous information Network.

Table 1. Dataset

Metapath2vec (mp2vec) [7]: This is a very advanced graph embedding model which applies the random walk method to graph structures most early.

ASNE [8]: This is an attribute graph embedding model that learns node features by combining "latent" features and attribute features.

SHNE [22]: This is also an attribute graph model, which learns node embedding by combining the tightness of the graph and semantic relevance.

GraphSAGE [12]: It is a neural network model that obtains node characteristics by sampling surrounding nodes and is widely used in application fields.

GAT [11]: It learns the degree of influence between nodes through the attention mechanism, thereby obtaining node representation.

HetGNN [13]: It samples "strongly associated" nodes through random walks and obtains node embeddings through node aggregation, which have great performance on Heterogeneous structure.

HAN [9]: It divides the heterogeneous graph into multiple homogeneous graphs through meta-path, and then aggregates through the double-layer attention mechanism to obtain the final representation of the nodes.

MAGNN [16]: It is a heterogeneous graph neural network model, which supplements the node information in the middle of the meta-path on the basis of HAN.

For homogeneous graph neural networks such as [11] and [12], we unify the nodes in the heterogeneous graph as the degree of advancement, and then input the data as a homogeneous graph.

5.3 Link Prediction

Link prediction is to learn the composition and structure of the existing edges in the structure, and then use the learned information to judge the possibility of the existence of unknown edge relationships. This task is widely applied in many fields, so we will use it as the first task scenario. The evaluation indicators we have chosen are AUC and $F1$, which can be expressed by formula:

$$AUC = \frac{\sum_{ins_i \in \text{positive Class}} \text{rank}_{ins_i} - \frac{P(1+P)}{2}}{P \times N}, \tag{7}$$

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP + FP}, \\ \text{Recall} &= \frac{TP}{TP + FN}, \\ F1\text{score} &= \frac{2\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}, \end{aligned} \tag{8}$$

In this task, we employ the same evaluation method as [13]. Specially, the first step of link prediction is to apply our model to train the embedding representation of each node, the second step is to employ the original links in the structure to train the binary classifier and the third step utilizes this binary classifier to evaluate the equal number of non-link relationships. In this process, we will only evaluate new links in the training data, the link embedding between the node and node is obtained by multiplying the embeddings of the nodes at both ends.

In Table 2, the experimental results for this task are shown, with the best results for each experiment marked in bold. According to these data: our model is better than all benchmark models in link prediction between nodes which have same type (Author-Author). Compared with HetGNN (after random wandering, all types of nodes are out of order), our model also uses the random walk method to preprocess the graph structure, but on the basis of random walk, we retain the original graph structure by means of meta path. Random walk gives our structure strong adaptability (adaptively selecting meta path instances for each node). The structure of meta path maintains the process of information transmission to a certain extent, The experimental results also show that our effect is better than HetGNN.

The main reason is that we obtain the representation of nodes through the aggregation of meta-path instances, and both ends of the meta-path instances are nodes with the same type. This structure can help our model to learn the relationship between the same types of nodes in the heterogeneous information network. Because

Dataset	Metric	M2vec	ASNE	SHNE	GSAGE	GAT	HetG	ASIAM-HGNN
$A-I_{2003}$	AUC	0.636	0.683	0.696	0.694	0.701	0.714	**0.735**
	F1	0.435	0.584	0.597	0.586	0.606	0.620	**0.633**
$A-I_{2002}$	AUC	0.626	0.667	0.688	0.681	0.691	0.710	**0.733**
	F1	0.412	0.554	0.590	0.567	0.589	0.615	**0.631**
$A-II_{2013}$	AUC	0.596	0.689	0.683	0.695	0.678	0.717	**0.732**
	F1	0.348	0.643	0.639	0.615	0.613	0.669	**0.683**
$A-II_{2012}$	AUC	0.586	0.671	0.672	0.676	0.655	0.701	**0.724**
	F1	0.318	0.615	0.612	0.573	0.560	0.642	**0.659**

$A-I_{2012}$ denotes that we use the data before 2012 as the train set, namely, the data after 2012 is the test set.
10 % of the test set is divided into the validation set.

Table 2. Experimental results of link prediction

of this characteristic, we believe that our model can have a great performance in many fields such as:

1. searching for user relationships in social networks;
2. disease relationship prediction;
3. compound relationship prediction.

5.4 Node Classification

Node classification is to classify nodes according to the existing node characteristics and labels, thereby predicting the category of nodes which don't have labels. The indicators used to evaluate the classification effect are: Micro-F1 and Macro-F1, which can be expressed by formula:

$$\text{Precision}_{\text{micro}} = \frac{\sum_{i=1}^{n} \text{TP}_i}{\sum_{i=1}^{n} \text{TP}_i + \sum_{i=1}^{n} \text{FP}_i},$$

$$\text{Recall}_{\text{micro}} = \frac{\sum_{i=1}^{n} \text{TP}_i}{\sum_{i=1}^{n} \text{TP}_i + \sum_{i=1}^{n} \text{FN}_i}, \tag{9}$$

$$F1_{\text{micro}} = 2 \cdot \frac{\text{Precision}_{\text{micro}} \cdot \text{Recall}_{\text{micro}}}{\text{Precision}_{\text{micro}} + \text{Recall}_{\text{micro}}}, \tag{10}$$

$$\text{Precision}_{\text{macro}} = \frac{\sum_{x=1}^{m} \text{Precision}_x}{m},$$

$$\text{Recall}_{\text{macro}} = \frac{\sum_{x=1}^{m} \text{Recall}_x}{m}, \tag{11}$$

$$F1_{\text{macro}} = 2 \cdot \frac{\text{Precision}_{\text{macro}} \cdot \text{Recall}_{\text{macro}}}{\text{Precision}_{\text{macro}} + \text{Recall}_{\text{macro}}}, \tag{12}$$

where i is the category of the node, and Precision and Recall are calculated in the same way as $F1$.

In this task, we follow the mathod of GrapgSAGE [12]. The data set itself does not have labels, and if magazines are designated as labels, there are too many types of labels. Therefore, we divide journals into four broad categories based on their characteristics and publications, and our criteria for labeling authors are the areas in which most of his papers are published. We learn the node representation through the ASIAM-HGNN model, and then input node representation and label into the logistic regression model. The dataset is divided into training set and test set according to different ratios, and 10 % of the test set is divided into the validation set.

Data-set	Train (%)	Metrics	M2-VEC	ASNE	SHNE	GSAGE	GAT	HetG	HAN	MAGNN	ASIAM-HGNN
A-I	10	Mac-F1	0.972	0.965	0.939	0.978	0.962	0.978	0.971	0.975	**0.982**
		Mic-F1	0.973	0.967	0.941	0.978	0.963	0.978	0.972	0.975	**0.983**
	30	Mac-F1	0.975	0.969	0.939	0.979	0.965	0.981	0.972	0.976	**0.983**
		Mic-F1	0.975	0.970	0.941	0.980	0.965	**0.982**	0.973	0.976	**0.982**
DBLP	20	Mac-F1	0.815	–	0.825	0.845	0.847	0.908	0.894	0.893	**0.913**
		Mic-F1	0.817	–	0.827	0.846	0.849	0.908	0.891	0.907	**0.912**
	40	Mac-F1	0.827	–	0.831	0.856	0.854	0.917	0.907	0.909	**0.919**
		Mic-F1	0.826	–	0.831	0.862	0.861	0.919	0.902	0.913	**0.919**
IMDB	20	Mac-F1	0.421	–	–	0.523	0.515	0.537	0.521	0.521	**0.539**
		Mic-F1	0.434	–	–	0.520	0.517	0.531	0.521	0.531	**0.536**
	40	Mac-F1	0.433	–	–	0.526	0.519	0.551	0.534	0.541	**0.554**
		Mic-F1	0.442	–	–	0.527	0.520	**0.559**	0.535	0.549	0.555

Table 3. Experimental results of node classification

The data of the tasks are shown in Table 3. It is clear that Our model has the best effect on most datasets. The main reason is that Our model collects meta path instances sufficient to cover most of the structural information of nodes through random walk method. Meta path instances enhance the ability of our model to learn the relationship between nodes through end-to-end form. So our model can achieve better performance in node classification.

Figure 3 shows the visualization of the embedded vector of the node learned from our model. We reduce the dimension of the high-dimensional vector representation through the TSNE method. Through the visual picture, we can clearly see that our model divides the nodes into four obvious categories.

6 ANALYSIS

In this section, the impact of some hyper parameter and intermediate structures will be shown to demonstrate the stability of the model.

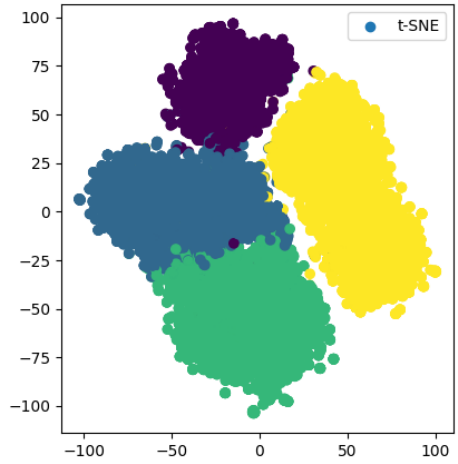


Figure 3. Embedded vector visualization in Academic-I dataset, each dot represents an author, and the color of the dot represents the corresponding field of the author

6.1 Ablation Study

In order to prove that the algorithm we selected in the model has a positive effect, we have done some ablation experiments and designed two different models for comparison.

1. We use the fully connected neural network MLP to embed the meta-path instances (ASIAM-MLP).
2. We use the fully connected neural network MLP to embed the meta-path instances (ASIAM-RNN).

The results of the ablation Study are shown in Figure 4, which can demonstrate:

1. ASIAM-HGNN uses various feature information of nodes which are very helpful to the improvement of experimental effects.
2. ASIAM-HGNN uses the bi-LSTM network in the method of learning the embedding representation of the meta-path instances.

In this way, we can not only preserve the objectively existing logical sequences in heterogeneous graphs, but also learn bi-directional information in meta-path instances through bi-LSTM.

6.2 Hyper-Parameters Sensitivity

Hyperparameters play an important role in our model, and they determine the amount of information we obtain. We design an experiment to analyze the impact

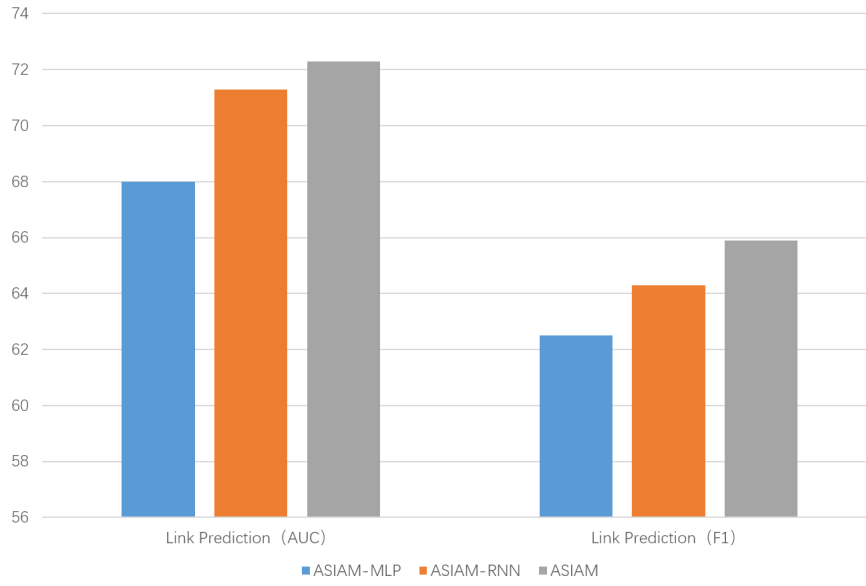


Figure 4. Performances of different models

of changes in the value of k (the number of meta-path instances we choose for each node) and dimensions of node embedding.

According to Figure 5, we can figure out that:

- With the change of k value, the experimental results will not change much, which shows the stability of our model.
- When the k value continues to rise from 3 to 8, the scores of AUC and $F1$ both reach the highest value. When k is equal to 6, there is a slight drop, which may be due to overfitting.

According to Figure 6, we can figure out that:

- When the dimension of the embedded vector increases step by step from 32 to 256, the scores of AUC and F1 indicators continue to improve. When the dimension reaches 128, the score reaches the highest point.

7 CONCLUSIONS

In our article, we propose a neural network model based on a heterogeneous graph structure to solve the previously mentioned problems of meta-path instances selection and low interpretability. We employ a meta-path instance selection method which based on random walk to automatically and inclusively select meta-path instances for each node, this step can automatically select the most associated meta-

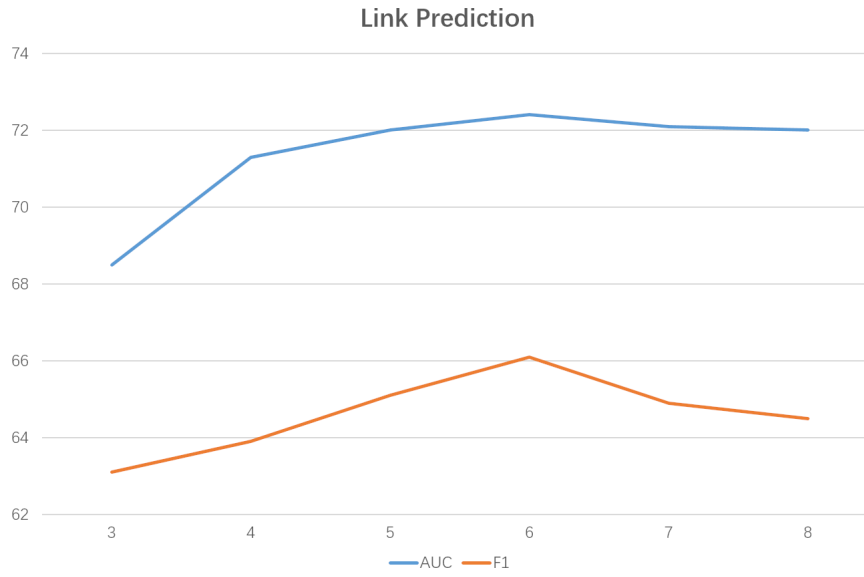


Figure 5. Performances of different value of k

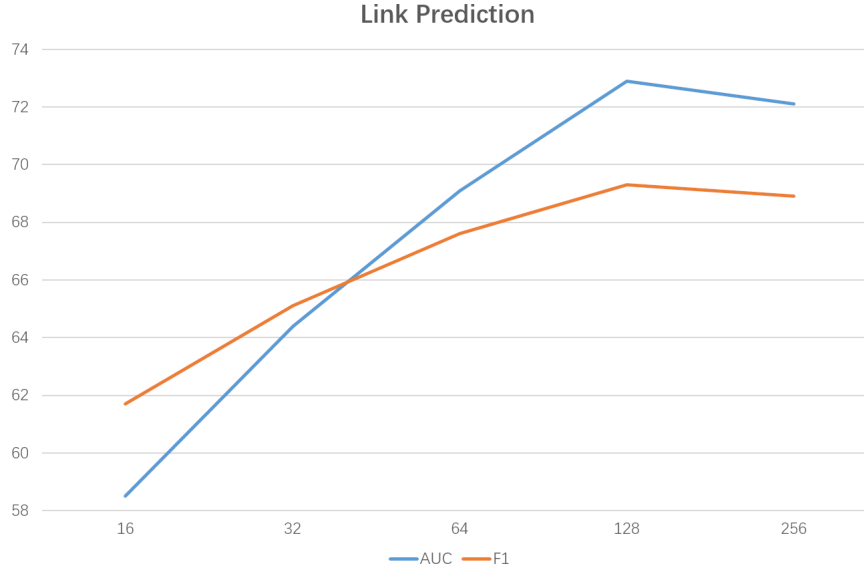


Figure 6. Performances of different value of embedding dimensions

path instances for each node because it makes full use of the objectively existing information in the graph structure. We take bi-LSTM for embedding learning of meta-path instance, which is able to preserves the logical sequence in the graph structure and earn higher interpretability. At the same time, our method aggregates different types of meta paths and single meta path instances of the same type of meta path at the same semantic level, which is conducive to learning the impact of different meta path instances on different nodes. This meta-path instances selection strategy refined to each node can more accurately learn the embedded representation of nodes. Our model has been tested on multiple tasks such as link prediction and node classification. In the task of link prediction, there is a 1.3 % to 2.3 % improvement over the baseline model. In the task of node classification, our model has a 1 % improvement over the baseline model on both the academic dataset and the DBLP dataset, the improvement is 0.5 %–1.8 % on the IMDB dataset. These experimental data confirm that our model has excellent performance in different application scenarios.

Acknowledgements

The paper is supported by the National Nature Science Foundation of China (No. 62-172299) and the Shanghai Science and Technology Committee (No. 22511105500).

REFERENCES

[1] Chen, T.—Sun, Y.: Task-Guided and Path-Augmented Heterogeneous Network Embedding for Author Identification. Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, 2017, pp. 295–304, doi: 10.1145/3018661.3018735.

[2] Zhang, C.—Huang, C.—Yu, L.—Zhang, X.—Chawla, N. V.: Camel: Content-Aware and Meta-Path Augmented Metric Learning for Author Identification. Proceedings of the 2018 World Wide Web Conference, 2018, pp. 709–718, doi: 10.1145/3178876.3186152.

[3] Hu, B.—Shi, C.—Zhao, W. X.—Yu, P. S.: Leveraging Meta-Path Based Context for Top-N Recommendation with a Neural Co-Attention Model. Proceedings of the 24$^{\text{th}}$ ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018, pp. 1531–1540, doi: 10.1145/3219819.3219965.

[4] Ren, X.—Liu, J.—Yu, X.—Khandelwal, U.—Gu, Q.—Wang, L.—Han, J.: Cluscite: Effective Citation Recommendation by Information Network-Based Clustering. Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 821–830, doi: 10.1145/2623330.2623630.

[5] Zhang, Y.—Xiong, Y.—Kong, X.—Li, S.—Mi, J.—Zhu, Y.: Deep Collective Classification in Heterogeneous Information Networks. Proceedings of the 2018 World Wide Web Conference, 2018, pp. 399–408, doi: 10.1145/3178876.3186106.

[6] Perozzi, B.—Al-Rfou, R.—Skiena, S.: Deepwalk: Online Learning of Social Representations. Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 701–710, doi: 10.1145/2623330.2623732.

[7] Dong, Y.—Chawla, N. V.—Swami, A.: Metapath2vec: Scalable Representation Learning for Heterogeneous Networks. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 135–144, doi: 10.1145/3097983.3098036.

[8] Liao, L.—He, X.—Zhang, H.—Chua, T. S.: Attributed Social Network Embedding. IEEE Transactions on Knowledge and Data Engineering, Vol. 30, 2018, No. 12, pp. 2257–2270, doi: 10.1109/TKDE.2018.2819980.

[9] Wang, X.—Ji, H.—Shi, C.—Wang, B.—Ye, Y.—Cui, P.—Yu, P. S.: Heterogeneous Graph Attention Network. The World Wide Web Conference, 2019, pp. 2022–2032, doi: 10.1145/3308558.3313562.

[10] Kipf, T. N.—Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks. Arxiv Preprint Arxiv:1609.02907, 2016, doi: 10.48550/arXiv.1609.02907.

[11] Veličković, P.—Cucurull, G.—Casanova, A.—Romero, A.—Lio, P.—Bengio, Y.: Graph Attention Networks. Arxiv Preprint Arxiv:1710.10903, 2017, doi: 10.48550/arXiv.1710.10903.

[12] Hamilton, W. L.—Ying, R.—Leskovec, J.: Inductive Representation Learning on Large Graphs. Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 1025–1035, doi: 10.48550/arXiv.1706.02216.

[13] Zhang, C.—Song, D.—Huang, C.—Swami, A.—Chawla, N. V.: Heterogeneous Graph Neural Network. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019, pp. 793–803, doi: 10.1145/3292500.3330961.

[14] Zhou, S.—Bu, J.—Wang, X.—Chen, J.—Wang, C.: Hahe: Hierarchical Attentive Heterogeneous Information Network Embedding. Arxiv Preprint Arxiv:1902.01475, 2019, doi: 10.48550/arXiv.1902.01475.

[15] Wang, S.—Chen, Z.—Li, D.—Li, Z.—Tang, L. A.—Ni, J.—Rhee, J.—Chen, H.—Yu, P. S.: Attentional Heterogeneous Graph Neural Network: Application to Program Reidentification. Proceedings of the 2019 SIAM International Conference on Data Mining, SIAM, 2019, pp. 693–701, doi: 10.1137/1.9781611975673.78.

[16] Fu, X.—Zhang, J.—Meng, Z.—King, I.: Magnn: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding. Proceedings of the Web Conference 2020, 2020, pp. 2331–2341, doi: 10.1145/3366423.3380297.

[17] Shang, J.—Qu, M.—Liu, J.—Kaplan, L. M.—Han, J.—Peng, J.: Meta-Path Guided Embedding for Similarity Search in Large-Scale Heterogeneous Information Networks. Arxiv Preprint Arxiv:1610.09769, 2016, doi: 10.48550/arXiv.1610.09769.

[18] Xu, L.—Wei, X.—Cao, J.—Yu, P. S.: Embedding of Embedding (EOE) Joint Embedding for Coupled Heterogeneous Networks. Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, 2017, pp. 741–749, doi: 10.1145/3018661.3018723.

[19] Tang, J.—Qu, M.—Mei, Q.: Pte: Predictive Text Embedding Through Large-Scale Heterogeneous Text Networks. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 1165–1174, doi: 10.1145/2783258.2783307.

[20] Shi, Y.—Zhu, Q.—Guo, F.—Zhang, C.—Han, J.: Easing Embedding Learning by Comprehensive Transcription of Heterogeneous Information Networks. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018, pp. 2190–2199, doi: 10.1145/3219819.3220006.

[21] Tang, J.—Qu, M.—Wang, M.—Zhang, M.—Yan, J.—Mei, Q.: Line: Large-Scale Information Network Embedding. Proceedings of the 24th International Conference on World Wide Web, 2015, pp. 1067–1077, doi: 10.1145/2736277.2741093.

[22] Zhang, C.—Swami, A.—Chawla, N. V.: Shne: Representation Learning for Semantic-Associated Heterogeneous Networks. Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, 2019, pp. 690–698, doi: 10.1145/3289600.3291001.

[23] Shi, C.—Hu, B.—Zhao, W. X.—Philip, S. Y.: Heterogeneous Information Network Embedding for Recommendation. IEEE Transactions on Knowledge and Data Engineering, Vol. 31, 2018, No. 2, pp. 357–370, doi: 10.1109/TKDE.2018.2833443.

[24] Bruna, J.—Zaremba, W.—Szlam, A.—LeCun, Y.: Spectral Networks and Locally Connected Networks on Graphs. Arxiv Preprint Arxiv:1312.6203, 2013, doi: 10.48550/arXiv.1312.6203.

[25] Defferrard, M.—Bresson, X.—Vandergheynst, P.: Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. Advances in Neural Information Processing Systems, Vol. 29, 2016, pp. 3844–3852, doi: 10.48550/arXiv.1606.09375.

[26] Meilian, L.—Danna, Y.: HIN_DRL: A Random Walk Based Dynamic Network Representation Learning Method for Heterogeneous Information Networks. Expert Systems with Applications, Vol. 158, 2020, Art. No. 113427, doi: 10.1016/j.eswa.2020.113427.

[27] Tong, N.—Tang, Y.—Chen, B.—Xiong, L.: Representation Learning Using Attention Network and CNN for Heterogeneous Networks. Expert Systems with Applications, Vol. 185, 2021, Art. No. 115628, doi: 10.1016/j.eswa.2021.115628.

[28] Zhang, Z.—Chen, C.—Chang, Y.—Hu, W.—Zheng, Z.—Zhou, Y.—Sun, L.: HIN2Grid: A Disentangled CNN-Based Framework for Heterogeneous Network Learning. Expert Systems with Applications, Vol. 187, 2022, 115823 pp., doi: 10.1016/j.eswa.2021.115823.

[29] Schlichtkrull, M.—Kipf, T. N.—Bloem, P.—Van Den Berg, R.—Titov, I.—Welling, M.: Modeling Relational Data with Graph Convolutional Networks. The Semantic Web (ESWC 2018), Springer, 2018, pp. 593–607, doi: 10.1007/978-3-319-93417-4_38.

[30] Hu, Q.—Lin, F.—Wang, B.—Li, C.: MBRep: Motif-Based Representation Learning in Heterogeneous Networks. Expert Systems with Applications, Vol. 190, 2022, Art. No. 116031, doi: 10.1016/j.eswa.2021.116031.

[31] Molaei, S.—Farahbakhsh, R.—Salehi, M.—Crespi, N.: Identifying Influential

Nodes in Heterogeneous Networks. Expert Systems with Applications, Vol. 160, 2020, Art. No. 113580, doi: 10.1016/j.eswa.2020.113580.

[32] Hong, H.—Guo, H.—Lin, Y.—Yang, X.—Li, Z.—Ye, J.: An Attention-Based Graph Neural Network for Heterogeneous Structural Learning. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 4132–4139, doi: 10.1609/aaai.v34i04.5833.

[33] Hu, Z.—Dong, Y.—Wang, K.—Sun, Y.: Heterogeneous Graph Transformer. Proceedings of the Web Conference 2020, 2020, pp. 2704–2710, doi: 10.1145/3366423.3380027.

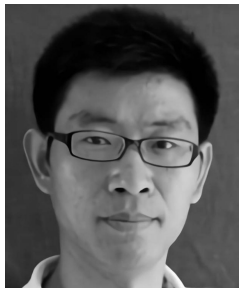
[34] Sun, Y.—Han, J.—Yan, X.—Yu, P. S.—Wu, T.: Pathsim: Meta Path-Based Top-K Similarity Search in Heterogeneous Information Networks. Proceedings of the VLDB Endowment, Vol. 4, 2011, No. 11, pp. 992–1003, doi: 10.14778/3402707.3402736.

[35] Cebrian, J. M.—Jahre, M.—Natvig, L.: Parvec: Vectorizing the PARSEC Benchmark Suite. Computing, Vol. 97, 2015, No. 11, pp. 1077–1100, doi: 10.1007/s00607-015-0444-y.

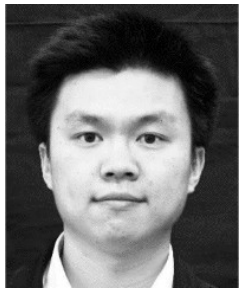
[36] Mikolov, T.—Sutskever, I.—Chen, K.—Corrado, G. S.—Dean, J.: Distributed Representations of Words and Phrases and Their Compositionality. Advances in Neural Information Processing Systems, Vol. 26, 2013, doi: 10.5555/2999792.2999959.

[37] Tang, J.—Zhang, J.—Yao, L.—Li, J.—Zhang, L.—Su, Z.: Arnetminer: Extraction and Mining of Academic Social Networks. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 990–998, doi: 10.1145/1401890.1402008.

Xiaojun Lou is a postgraduate in the School of Mathematics and Computer Science, at Zhejiang Agriculture and Forestry University. His current research interests include graph neural networks, recommender systems, and multimodality.



Guanjun Liu received his Ph.D. degree in computer software and theory from Tongji University, Shanghai, China, in 2011. He was a Post-Doctoral Research Fellow with the Singapore University of Technology and Design, Singapore, from 2011 to 2013, and a Post-Doctoral Research Fellow with the Humboldt University of Berlin, Berlin, Germany, from 2013 to 2014, supported by the Alexander von Humboldt Foundation. He is currently a Professor at the Department of Computer Science, at Tongji University. He has authored over 130 papers and 3 books. He served as a Guest Editor for some journals including the IEEE Transactions on Computational Social Systems, the Mathematical Problems in Engineering, and the Journal of Software (in Chinese), and served as a Program Committee Member for many international conferences. He is now an Associate Editor of the IEEE Transactions on Computational Social Systems. His research interests include Petri net theory, model checking, machine learning, cyber-physical systems, workflow, and credit card fraud detection.



Jian Li received his Ph.D. degree in computer software and theory from Tongji University, Shanghai, China, in 2020. He is currently a Vice Professor at the School of Mathematics and Computer Science, Zhejiang A & F University, Hangzhou, China. His research interests include matrix decomposition, graph neural networks, and their applications.

Computing and Informatics, Vol. 42, 2023, 280–310, doi: 10.31577/cai_2023_2_280

NOVEL APPROACH TO HIDE SENSITIVE ASSOCIATION RULES BY INTRODUCING TRANSACTION AFFINITY

Kshitij Pathak

Department of Computer Science and Engineering
Government Polytechnic College
Mandsaur, MP, India
e-mail: er.k.pathak@gmail.com

Sanjay Silakari

Department of Computer Science and Engineering
University Institute of Technology
RGPV Bhopal, MP, India
e-mail: ssilakari@yahoo.com

Narendra S. Chaudhari

Department of Computer Science and Engineering
Indian Institute of Technology
Indore, MP, India
e-mail: nsc0183@yahoo.com

Abstract. In this paper, a novel approach has been proposed for hiding sensitive association rules based on the affinity between the frequent items of the transaction. The affinity between the items is defined as Jaccard similarity. This work proposes five algorithms to ensure the minimum side-effects resulting after applying sanitization algorithms to hide sensitive knowledge. Transaction affinity has been introduced which is calculated by adding the affinity of frequent items present in the transaction with the victim-item (item to be modified). Transactions are selected either by increasing or decreasing value of affinity for data distortion to hide

association rules. The first two algorithms, MaxaffinityDSR and MinaffinityDSR, hide the sensitive information by selecting the victim item as the right-hand side of the sensitive association rule. The next two algorithms, MaxaffinityDSL and MinaffinityDSL, select the victim item from the left-hand side of the rule whereas the Hybrid approach picks the victim item from either the left-hand side or right-hand side. The performance of proposed algorithms has been evaluated by comparison with state-of-art methods (Algo 1.a and Algo 1.b), MinFIA, MaxFIA and Naive algorithms. The experiments were performed using the dataset generated from IBM synthetic data generator, and implementation has been performed in R language.

Keywords: Association rule hiding, transaction affinity, affinity, AffinityDSR, AffinityDSL

Mathematics Subject Classification 2010: 68T05, 68T30

1 INTRODUCTION

1.1 Terminology and Preliminaries

1. Let $I = \{I_1, I_2, I_3, I_4, \ldots, I_n\}$, a set containing a finite set of literal, known as items and cardinality of the set represented by $|I|$ is n.
2. Any subset $I_s \subseteq I$ is known as an itemset over I. An itemset of size n, represented as n-itemset. Example: Let $I = \{a, b, c, d, e\}$ then a, b, c, d and e are known as items or 1-itemset. ab, bc and cd are examples of 2-itemset. abc, bcd and cde are examples of 3-itemset and so on.
3. The support of itemset I is equal to the number of transaction having itemset I in database D. It is represented by Support(I) or Support(I, D).
4. Database D consisting of m transactions is represented by $D = \{T_1, T_2, T_3, T_4, \ldots, T_m\}$ and $|D|$ is equal to the number of transactions present in the dataset.
5. A transaction set T_s over I is pair $T_s = (\text{tid}, I_s)$ where I_s is the itemset and tid is a unique identifier.
6. $T[m, n]$ is used to identify n^{th} item of m^{th} transaction in the database.
7. The support of the association rule $A \to B$ is calculated by

$$\text{Support}(A \to B) = \text{Support}(A, B) \div |D|. \tag{1}$$

8. The Confidence of the association rule $A \to B$ is calculated by

$$\text{Confidence}(A \to B) = \text{Support}(A, B) \div \text{Support}(A). \tag{2}$$

9. An itemset is said to be frequent if its support is greater than user defined mfreq, minimum frequency threshold and itemsets having frequency lower than mfreq are treated as non-frequent items.

10. Relation between mfreq and user defined threshold MST is MST (%) = mfreq $\div$ $|D| * 100$.

11. An association rule $A \rightarrow B$ is considered to be significant if support and confidence of the rule is greater than or equal to user defined support and Confidence i.e. MST (Minimum Support Threshold) and MCT (Minimum Confidence Threshold) i.e. Support$(A \rightarrow B) \geq$ MST and Confidence$(A \rightarrow B) \geq$ MCT.

12. User marked some association rules to be hided from the set of generated association rules, known as *sensitive* and remaining rules are known as *non-sensitive association rules*.

13. A subset of transactions that can be modified/updated to hide sensitive association rule is known as *candidate transactions or sensitive transactions* whereas modified candidate transactions are known as *victim transactions*.

14. *Victim item* is a term used to represent the itemset being modified by the association rule hiding method.

1.2 Motivating Example

Let us suppose that the officials of BigMDSBazaar company purchase socks from XYZ socks company. XYZ socks company official gives an offer to BigMDSBazaar official that they are ready to reduce the price of their socks if BigMDSBazaar would share their customer data with XYZ socks company. BigMDSBazaar thought that it is a good deal and customer's data can be shared, so, both the parties have accepted the deal. XYZ socks company now started mining the customer's data they receive from BigMDSBazaar. They identified that, in 70 % cases, whoever purchases A type of shoes also going to purchase XYZ socks. After learning this knowledge from the datasets, XYZ socks company made a public offer that the customer purchasing the socks from their company will be given a 20 % cashback on the purchase of A type of shoes. After this particular advertisement, many customers who are purchasing A type of shoes earlier from BigMDSBazaar have moved to XYZ socks company. In this particular scenario, we can see that BigMDSBazaar is losing its customers. At the same time, XYZ socks company increases the price of socks purchased by BigMDSBazaar in the next deal since the sales of socks have decreased at the stores of BigMDSBazaar. BigMDSBazaar is harming its own business by sharing its customer's data. Finally, we can conclude that BigMDSBazaar lost business to the XYZ socks company as a result of exchanging data without sanitizing it. Several other motivating examples have been discussed in [1, 2, 3, 4].

1.3 Association Rule Hiding

Association rules are defined as statements of the form $\{X_1, X_2, \ldots, X_n\} \rightarrow Y$ which means that Y may be present in the transaction if X_1, X_2, …, X_n are all in the transaction. Notice that the use of may imply that the rule is only probable, not identical. Note also, that there can be a set of items, not just a single item. The probability of finding Y in a transaction with all X_1, X_2, …, X_n is called confidence. The threshold (percentage) that a rule holds in all transactions is called support. The level of confidence that a rule must exceed is called interestingness.

In this research work, we focus on knowledge hiding in the database, i.e., association rule hiding. The association rule hiding problem is an extension of the very well-known database inference control problem that has been applied to multilevel and statistical databases. In the database inference control problem, the primary objective is to hide sensitive information that can be inferred from non-sensitive data, whereas in association rule hiding it is inferred that it is not only the data but the hidden information is also a threat to privacy. So, in association rule hiding, before sharing the database various data mining techniques have been applied to identify the sensitive knowledge and then sanitization algorithms are applied to modify the dataset before releasing it to hide the sensitive information/knowledge earlier present in the dataset. The main challenges that are associated with association rule hiding are what strategy should be adopted in modifying the transactions of the database so that sensitive association rule gets hidden whereas non-sensitive association rule can still be mind as earlier as possible. Another critical factor that has to be taken into consideration is how much we are modifying the dataset, i.e., there must be a proper balance between privacy and utility. Association rule hiding or hiding large itemsets approaches can be divided into five major classes.

They are heuristic, border-based, exact, data reconstruction based and cryptographic approaches. The first class, i.e., *heuristic approaches* algorithms, [4, 2, 5, 6, 7, 8] are efficient, fast, and selectively sanitize the candidate transactions and victim items from original datasets to hide the sensitive information. Heuristic approaches are a dominant area of importance and research interest because of their efficiency and scalability. However, such algorithms suffer from various side-effects because the decisions taken are local decisions, which do not necessarily provide the global best solution.

The second class, *border-based approaches*, hide sensitive association rules by modifying the borders in the lattice of frequent and non-frequent itemsets by selecting itemsets of the lattice that control the borderline position which separates frequent and nonfrequent itemsets. Border-based algorithms hide sensitive association rules by tracking the non-sensitive frequent itemsets borders, and greedily data modifications are applied such that there is a minimum effect on the quality of the border to accommodate the hiding of the rules. The work related to the border-based approach is proposed in Main [9], BBA [9], Max-Min [10], positive and negative border-based algorithms [11], and the AARHIL algorithm [12].

The third class, *exact approaches*, treats the hiding problem as a constraint satisfaction problem solved by integer programming. The constraint satisfaction problem is an optimization problem that enables the algorithm to identify optimal solutions by minimum distorting the database and introducing no new side effects. On the contrary, the time complexity of exact approaches is higher when compared to heuristic approaches because of the time taken by the integer programming solver to solve the optimization problem [13].

The fourth class, *data reconstruction-based approaches* [14, 15, 16, 17], perform the hiding process in three phases. In the first phase, association rules are mined, sensitive rules are selected by the data owner and itemset lattice is built. In the second phase, knowledge sanitization is applied on itemset lattice and the database is reconstructed from the modified lattice in the third phase.

The fifth class, *cryptographic approaches*, is used in multi-party computation where data is dispersed in several locations [18]. The owner may want to share their data but does not want the confidential information to be disclosed. The cryptographic approach can be divided into two categories: vertical partitioned distributed data and horizontal partitioned distributed data. Various approaches in this class are discussed in [19, 20, 21, 22, 23, 24].

2 AFFINITY-BASED APPROACH

The affinity between the two items i and j is defined by Aggarwal et al. [25] as

$$A(i,j) = \text{support}(i,j)/(\text{support}(i) + \text{support}(j) - \text{support}(i,j)), \tag{3}$$

where support(.) is the count of the presence of an item in the dataset. This means that affinity is defined as the Jaccard similarity between items.

We propose five sanitization algorithms based on the affinity between the frequent itemsets by introducing the concept of *transaction affinity*. Transaction affinity is calculated by adding the affinity of frequent items present in the transaction with the victim-item (Victim-item is the item selected for modification in the transaction.). Transactions having a high value of transaction affinity signify that the set of items present in the transaction has high similarity with the victim-item. We have conducted experiments to analyze the proposed methods of picking the transaction for modification based on the similarity between victim-item and frequent items present in the transaction.

The first two algorithms, MAXAffinityDSR and MINAffinityDSR, hide the sensitive association rule by selecting transactions on the basis of the similarity of the victim-item (Right-hand side of sensitive association rule) with frequent items present in the transaction. This method hides the rule by decreasing the support of the right-hand side of the rule (victim-item) thereby reducing the confidence or support of the sensitive association rule below a minimum threshold. The third and fourth approach, MAXAffinityDSL and MINAffinityDSL, consider the victim-item as the left-hand side of the rule whereas the fifth one, the hybrid approach, selects

the victim-item by using HybridCode function present in Algorithm 3. Initially, the candidate transaction picked is the one that contains all the items of sensitive association rules, i.e., if $B \rightarrow A$ is to be hidden, then candidate transactions are the ones having BA as the subset of items present in the transaction. In earlier approaches, then candidate transactions are arranged in ascending or descending order either on the basis of length of the transaction as done in [4] or they completely remove the frequent itemset from all transactions which results in increasing the number of side-effects like the border-based approach. The proposed approach uses the concept of transaction affinity to select transactions to be sanitized.

2.1 Hiding Strategy

A sensitive association rule $A \rightarrow B$ can be hidden by following two hiding strategies as per the taxonomy of association rule hiding algorithms:

1. *Support-based or reducing support below user-specified threshold:* We know that support of rule $A \rightarrow B$ is defined as the count of transactions having both A and B divided by a total number of transactions. Support can be reduced by removing either the left-hand side (A) or right-hand side (B) from the transactions in which both A and B are present.
2. *Confidence-based or reducing confidence below user-specified confidence:* Confidence of the rule is defined as support of rule items divided by the support of left-hand side of the rule. To hide the rule by confidence, any item from the right-hand side of rule is to be removed from candidate transaction (Candidate transaction, in this case, are the ones which fully supports the rule) or support of left-hand side of rule is increased by adding items in candidate transaction (Candidate transactions, in this case, are the ones which do not support left-hand and right-hand side of rule).

Example: Consider a sample database D shown in Table 1.

Transaction_Id	Items
1	ABC
2	AB
3	A
4	C

Table 1. Database D

Support Based: In the database, support of rule $A \rightarrow B$ is $2 \div 4$ or 50 % as both A and B are present in two transactions (T_1 and T_2) out of 4 transactions. Let the user-specified support threshold is 30 %. Here, the candidate transactions are T_1 and T_2, as they fully support the rules (i.e., all rule items are present in T_1 and T_2). If we remove any item from the rule from either T_1 or T_2, then the support of the rule will be $1 \div 4$ which is equal to 25 % which makes it falls below a user-specified threshold, and eventually, rules get hided.

Confidence-Based: In the database, the confidence of rule $A \rightarrow B$ is $2 \div 3$ or 66.67 % as both A and B are present in two transactions (T_1 and T_2) out of 4 transactions and A is present in 3 transactions. Let the user-specified confidence threshold is 55 %.

To hide the rule by deleting a subset of the right-hand side of the rule, we need to make 1 modification to hide the rule. Support of AB is reduced to 1 and confidence will be reduced to $1 \div 3$ or 33.33 %.

To hide the rule by increasing support of the left-hand side, we need to make 1 modification to hide the rule. Candidate transaction is 4 since transaction 4 does not support rule items. Item A will be added to transaction 4 which increases support of A to 4. Confidence will be reduced to $2 \div 4$ or 50 % which makes it falls below the user-specified confidence threshold and eventually rule gets hided.

2.2 Preliminary Definitions

Definition 1. The Transaction Id's in support of sensitive rule S_r and will be the candidate for MAXAffinityDSR and MINAffinityDSR approach is defined by:

$$T_{id}(S_r) = T_{id}(L_{S_r}) \cap T_{id}(R_{S_r}), \tag{4}$$

where L_{S_r} and R_{S_r} are the left-hand and right-hand side of the sensitive association rule.

Definition 2. The affinity of each transaction, i.e., transaction affinity in MAXAffinityDSR and MINAffinityDSR algorithm is calculated by

$$\text{affinity}_{sum}(T_{id}) = \sum_{\substack{for\ all\ 1-itemset\ m\ in\ T_{id} \\ and \\ m \subset F \\ and \\ m\ !=\ L_{S_r} \\ and \\ for\ all\ r\ in\ R_{S_r}}} \text{affinity}(m, r), \tag{5}$$

where F is set of frequent itemsets, T_{id} is a transaction, L_{S_r} and R_{S_r} are left-hand and right-hand side of association rule. It is calculated by summation of the affinity between every 1-itemset from the right-hand side of a sensitive association rule, and every frequent 1-itemset present in the transaction ignoring itemsets belongs to the left-hand side of the transaction. Consider a sample database TD shown in Table 2.

Consider a transaction T_1 with items $UWXYZ$, sensitive association rule $U \rightarrow X$, frequent items = $\{U, W, X, Z\}$. Then, $m = \{W, Z\}$, $r = \{X\}$ and transaction affinity for T_1 is affinity(X, W) + affinity(X, Z).

Transaction_Id	Items
T1	UWXYZ
T2	UWXZ
T3	UWX
T4	WXZ

Table 2. Database *TD*

Definition 3. All the candidate transactions identified for approaches are ordered in decreasing order in MAXaffinityDSR and MAXAffinityDSL on the basis of the affinity value evaluated for transactions ($\text{affinity}_{sum}(T_{id})$).

$$\text{SORT}\left(T_{id}, \text{affinity}_{sum}\left(T_{id}\right), \text{decreasing} = \text{TRUE}\right). \tag{6}$$

Definition 4. All the candidate transactions identified for approaches are ordered in increasing order in MINaffinityDSR and MINAffinityDSL on the basis of $\text{affinity}_{sum}(T_{id})$ as described in Equation (5).

$$\text{SORT}\left(T_{id}, \text{affinity}_{sum}\left(T_{id}\right), \text{decreasing} = \text{FALSE}\right). \tag{7}$$

Definition 5. The candidate transaction for AffinityDSL is defined by:

$$T_{id}\left(S_r\right) = T_{id}\left(L_{S_r}\right) \cap T_{id}\left(R_{S_r}\right), \tag{8}$$

where L_{S_r} and R_{S_r} are the left-hand and right-hand side of association rule(S_r).

Definition 6. The transaction affinity in MAXAffinityDSL and MINAffinityDSL is calculated by

$$\text{affinity}_{sum}\left(T_{id}\right) = \sum_{\substack{for\ all\ 1-itemset\ m\ in\ T_{id} \\ and \\ m \subset F \\ and \\ m\ !=\ R_{S_r} \\ and \\ for\ all\ l\ in\ L_{S_r}}} \text{affinity}(m, l), \tag{9}$$

where F is set of frequent itemsets, T_{id} is a transaction, L_{S_r} and R_{S_r} are left-hand and right-hand side of association rule. It is calculated by summation of the affinity between every 1-itemset from the left-hand side of a sensitive association rule, and every frequent 1-itemset present in the transaction ignoring itemsets belongs to the right-hand side of the transaction.

2.3 Algorithm Parameters

In association rule hiding, the decision has to be taken in the following aspects which significantly affect the performance of the algorithm:

1. *Victim-item:* In the proposed work, first, two algorithms, MaxAffinityDSR and MinAffinityDSR, pick the victim item as a subset of the right-hand side of the rule whereas the next two algorithms, MaxAffinityDSL and MinAffinityDSL, pick victim item as a subset of the left-hand side of the rule. AffinityHybrid algorithm picks the victim item at runtime based on a minimum number of modifications required to hide the rule. All five sanitization algorithms remove the item from the database.
2. *Picking the candidate transactions:* In the proposed algorithms, candidate transactions are the ones that fully support the rule.
3. *Selecting transactions for modification from candidate transactions:* The primary decision of the association rule hiding algorithm falls in this step. All the candidate transactions are not modified. A subset of candidate transactions is altered until the rule's support or confidence is below the minimum support threshold and minimum confidence threshold, respectively. In proposed approaches, an affinity between frequent items is calculated and stored in a two-dimensional array in which dimensions are items. Then transaction affinity for each transaction is calculated as per the approach. Now candidate transactions are sorted by transaction affinity. The transaction affinity value is stored in a one-dimensional array.

2.4 Procedure

The procedure for MaxAffinityDSR, MinAffinityDSR, MaxAffinityDSL and MinAffinityDSL is as follows:

1. Generate association rule from the database using the Apriori algorithm.
2. The owner analyzes association rules generated and identifies sensitive association rules.
3. Repeat steps 4 to 10 for every sensitive association rule.
4. Calculate affinity between the items using Equation (3).
5. Pick candidate transactions like the ones which support all the items present in the sensitive association rule defined in Equations (4) and (8).
6. Calculate transaction affinity for all candidate transactions identified in step 5. The transaction affinity is calculated by adding the affinity of the victim item with every frequent 1-itemset present in the transaction. It is defined in Equations (5) and (9).
7. Sort candidate transactions by transaction affinity as defined in Equations (6) and (7).
8. Calculate the minimum number of modifications required to hide the rule.
9. Select the victim item as the right-hand side of the rule for AffinityDSR and the left-hand side of the rule for AffinityDSL respectively, which is going to be deleted from candidate transactions.

10. Pick one by one transaction from the sorted candidate transaction list prepared in step 5 and remove the victim item, till the number of modifications required is achieved.

2.5 MAXAffinityDSR and MINAffinityDSR Algorithm

```
input  : dataset, minsupp, minconf, sensitive association rules
output: Modified database to hide association rules
1  rulesdataset = apriori (dataset, parameter = list(supp = minsupp, conf = minconf,
     minlen = 2));
   // extract rules from dataset with user defined support and confidence threshold
2  inspect (rulesdataset);
   // user decides which rules are sensitive, needs to be hided
3  ruletohide = subset (rulesdataset);
   // selects sensitive rule in ruletohide
4  ruletohidec = |ruletohide |;
5  for o ← 1 to ruletohidec do
6      aff ← affinity(dataset);
       // calculate affinity among the frequent itemset
7      lhs ← selectLhs(ruletohide [o]);
8      rhs ← selectRhs(ruletohide [o]);
9      lhslist = {t | tεtransactions and t fully support LHS of rule};
10     rhslist = {t | tεtransactions and t fully support RHS of rule};
11     transactionDSR ← intersect(lhslist, rhslist);
       // prepare candidate transaction list in transactionDSR by intersecting
          lhslist and rhslist
12     tosorttransactionDSR ← NULL;
13     for i Input: transactionDSR
14      do
15         k ← o;
16         for j Input: items
17          do
18             if j ≠ lhs and j ≠ rhs and j is frequent and i contains j then
19                k = k + aff[rhs, j];
20             end
21             affinity(i) = k;
22             tosorttransactionDSR = concate(tosorttransactionDSR, concate(i, affinity(i)));
               // tosorttrnasactionDSR contain candidate transaction with their
                  respective affinity sum
23         end
24     end
25     sort(tosorttransactionDSR, sortby(affinity(i),i) Decreasing = FALSE);
       // Sorted Candidate Transactions
26     NoOfModificationinDSR
         ← minimum(ceiling((length(transactionDSR) − (TotalNumberOfTransaction ∗ MST
       /100))), ceiling((length(transactionDSR) − (length(lhslist) ∗ MCT /100))));
27     NoofModificationsDoneinDB ← 0;
28     for i ← 1 to NoOfModificationinDSR do
29         NoofModificationsDoneinDB ← NoofModificationsDoneinDB + 1;
30         pick transaction i from tosorttransactionDSR;
31         Dataset [i, rhs] ← 0;
32     end
33 end
```

Algorithm 1. MINAffinityDSR algorithm

The basis of the first two approaches, MAXAffinityDSR and MINAffinityDSR, select those transactions as victim transactions from the candidate transactions having a maximum and minimum value of transaction affinity respectively. In MAXAffinityDSR and MINAffinityDSR, the victim item is on the right-hand side of the sensitive association rule. Suppose there is a transaction t having items A, B, C, D, E. Suppose A is to be removed from transactions assuming it is present on the right-hand side of the sensitive rule $B \rightarrow A$. Let D and E be frequent itemsets and C is non-frequent itemsets. The affinity value of the transaction is identified as the sum of the affinity of A with D and E. Affinity of B is not considered while calculating the affinity of the transaction since the item belongs to the sensitive association rule which needs to be hidden. Affinity calculation with item C is not considered in the affinity value of the transaction since there does not exist any association rule having C on either side of the rule, as it is a non-frequent itemset. In this way all the transactions now have their affinity value then transactions are sorted by their affinity value in decreasing order in MAXAffinityDSR algorithm and in increasing order in case of the MINAffinityDSR algorithm. Then, one by one transactions are picked, and item present on the right-hand side of the rule is deleted from transaction till the confidence or support of the rule falls below the minimum threshold. The motivation for doing that is to reduce the side effects associated with the distorted database. This approach, MINAffinityDSR, is presented in Algorithm 1. In the MAXAffinityDSR algorithm, the only change is a selection of victim transactions from candidate transaction set; here candidate transactions are sorted in decreasing order of their affinity value.

Consider an example shown in Table 3 with user-defined support threshold of 55 % and user-defined confidence threshold is 80 %, following 14 association rules gets generated as shown in Table 4.

Transaction_Id	**Items**
1	ABCD
2	ABCD
3	ABC
4	ABCD
5	C
6	B
7	ABDEF

Table 3. Database D1

The affinity values of the items are calculated by Equation (3) as shown in Table 5.

Let $B \rightarrow A$ be the sensitive association rule that needs to be hided, then candidate transactions to be sanitized are identified as 1, 2, 3, 4, 7 since B and A both are present in this transactions. The affinity of these five transactions is calculated by adding the affinity of frequent items present in the transaction with the items of the right-hand side of a rule by Equation (5). So, the calculation of affinity of

S_No	LHS → RHS	Support	Confidence	Lift
1	$\{C\} \to \{A\}$	0.5714286	0.8	1.12
2	$\{A\} \to \{C\}$	0.5714286	0.8	1.12
3	$\{C\} \to \{B\}$	0.5714286	0.8	0.9333333
4	$\{D\} \to \{A\}$	0.5714286	1	1.4
5	$\{A\} \to \{D\}$	0.5714286	0.8	1.4
6	$\{D\} \to \{B\}$	0.5714286	1	1.1666667
7	$\{A\} \to \{B\}$	0.7142857	1	1.1666667
8	$\{B\} \to \{A\}$	0.7142857	0.8333333	1.1666667
9	$\{A, C\} \to \{B\}$	0.5714286	1	1.1666667
10	$\{B, C\} \to \{A\}$	0.5714286	1	1.4
11	$\{A, B\} \to \{C\}$	0.5714286	0.8	1.12
12	$\{A, D\} \to \{B\}$	0.5714286	1	1.1666667
13	$\{B, D\} \to \{A\}$	0.5714286	1	1.4
14	$\{A, B\} \to \{D\}$	0.5714286	0.8	1.4

Table 4. Association rules for sample databases D1

	A	B	C	D	E	F
A	0	0.8333333	0.6666667	0.8	0.2	0.2
B	0.8333333	0	0.5714286	0.6666667	0.1666667	0.1666667
C	0.6666667	0.5714286	0	0.5	0	0
D	0.8	0.6666667	0.5	0	0.25	0.25
E	0.2	0.1666667	0	0.25	0	1
F	0.2	0.1666667	0	0.25	1	0

Table 5. Affinity matrix

transactions is as follows:

- $\text{Affinity}(1) = \text{affinity}(A, C) + \text{affinity}(A, D) = 0.6666667 + 0.80 = 1.4666667$,
- $\text{Affinity}(2) = \text{affinity}(A, C) + \text{affinity}(A, D) = 0.6666667 + 0.80 = 1.4666667$,
- $\text{Affinity}(3) = \text{affinity}(A, C) = 0.6666667$,
- $\text{Affinity}(4) = \text{affinity}(A, C) + \text{affinity}(A, D) = 0.6666667 + 0.80 = 1.4666667$,
- $\text{Affinity}(7) = \text{affinity}(A, D) = 0.80$.

Note. E and F are not included in finding affinity of the transaction as they are non-frequent items and B is not considered since it belongs to sensitive association rule in the process of being hided.

For hiding the rule $B \to A$, item A, right-hand side of the rule, must be removed from transaction to reduce the support or confidence below the threshold. For reducing support below the threshold, number of modification required is 2, and for reducing confidence below the threshold, number of modification required is 1. Picking the minimum among these two is a sufficient and necessary number of modification required to successfully hide sensitive association rule.

Transaction_Id	Items
1	ABCD
2	ABCD
3	BC
4	ABCD
5	C
6	B
7	ABDEF

Table 6. Released Database D1

Since transaction affinity of Transaction 3 is minimum for MINAffinityDSR, so it is picked as the first transaction to be sanitized, and A is removed. Released database is shown in Table 6.

2.6 MAXAffinityDSL and MINAffinityDSL Algorithm

In this approach, sensitive association rules are hided by reducing the support of left-hand side thereby reducing the support of sensitive association rule. Same process as applied for hiding right-hand side of the rule in MAXAffintyDSR can be implemented with LHS in MAXAffinityDSL and MINAffinityDSL, i.e., the affinity of all the transaction is calculated by summing the affinity of the LHS with every frequent 1-itemset present in the transaction but ignoring the itemsets of the right-hand side of the rule. The insight for selecting the transaction based on affinity sum lies in the fact that more petite will be a side-effect of the modification if the similarity between the victim item and other frequent items is considered while selecting transactions. This approach is presented in Algorithm 2.

2.7 AffinityHybrid Algorithm

The third approach (Algorithm 3 AffinityHybrid) shown in Figure 1 combines the above two approaches to have a mixture of reducing the support of a subset of the left-hand side and right-hand side of the sensitive association rule.

In this method first, it is identified that for hiding sensitive association rule how many modifications are required to hide the rule on lowering the confidence below minimum confidence threshold and how many modifications are necessary to cover up the rule by reducing the support below the minimum support threshold. If the number of modification required is less for reducing confidence below MCT rather than reducing support below MST then victim item belongs to the right-hand side of the rule and removed from the transaction, we call it a hybrid(0) otherwise, items belongs to either left-hand side or right-hand side are removed as per HybridCode function. We call it a hybrid(1).

For hybrid(0), the approach is applied in the same way defined for the AffinityDSR approach.

```
input : Database, minsupp, minconf, sensitive association rules
output: Modified database to hide association rules
rulesdataset = apriori (dataset, parameter = list(supp = minsupp, conf = minconf,
  minlen = 2));
// extract rules from dataset with user defined support and confidence threshold
inspect (rulesdataset);
// user decides which rules are sensitive, needs to be hided
ruletohide = subset (rulesdataset);
// selects sensitive rule in ruletohide
ruletohidec = |ruletohide|;
for o ← 1 to ruletohidec do
    aff ← affinity(dataset);
    // calculate affinity among the frequent itemset
    lhs ← selectLhs(ruletohide [o]);
    rhs ← selectRhs(ruletohide [o]);
    lhslist = {t | tε transactions and t fully support LHS of rule};
    rhslist = {t | tε transactions and t fully support RHS of rule};
    transactionDSL ← intersect(lhslist, rhslist);
    // prepare candidate transaction list in TransactionDSL by intersecting
       lhslist and rhslist
    tosorttransactionDSL ← NULL;
    for i Input: transactionDSL
     do
        k ← o;
        for j Input: items
         do
            if j ≠ lhs and j ≠ rhs and j is frequent and i contains j then
              k = k + aff[lhs, j]
            end
            affinity(i) = k;
            tosorttransactionDSL = concate(tosorttransactionDSL, concate(i, affinity(i)))
               // TosorttrnasactionDSL contain candidate transaction with their
               respective affinity sum
        end
    end
    sort(tosorttransactionDSL, sortby((affinity(i), i)));
    // MINAffinityDSL and MAXAffinityDSL sort transactions in increasing and
       decreasing order respectively.
    NoOfModificationinDSL
      ← ceiling((length(transactionDSL) − (TotalNumberOfTransaction ∗ MST /100)));
    NoofModificationsDoneinDB ← 0;
    for i ← 1 to NoOfModificationinDSL do
        NoofModificationsDoneinDB ← NoofModificationsDoneinDB + 1;
        pick transaction i from tosorttransactionDSL;
        Dataset [i, lhs] ← 0;
    end
end
RulesNew = apriori (dataset, parameter = list(supp = minsupp, conf = minconf, minlen = 2));
inspect (RulesNew);
```

Algorithm 2. MAXAffinityDSL and MINAffinityDSL algorithm

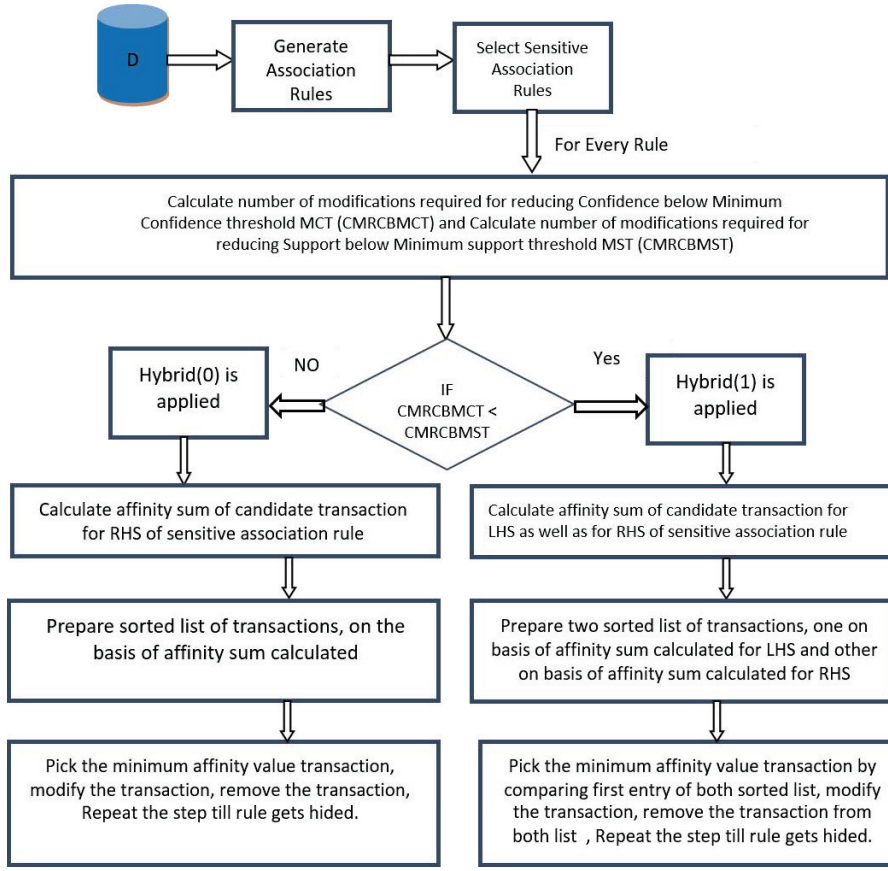


Figure 1. Hybrid Approach

For hybrid(1), the affinity of the transaction is divided into two parts:

1. Affinity of the transaction for left-hand side.
2. Affinity of the transaction for right-hand side.

The affinity of the transaction for the left-hand side is calculated by summing the affinity of the left-hand side with frequent items presenting in the transaction ignoring items belongs to the right-hand side of the transaction. The affinity of transactions for the right-hand side is calculated by summing the affinity of the right-hand side with frequent items present in the transaction ignoring the items belongs to the left-hand side. Then affinity of the transaction for the left-hand side is sorted on the basis of affinity calculated in increasing order. Similarly, affinity for the right-hand side is sorted on the basis of affinity calculated in the increasing order.

```
input  : Database, minsupp, minconf, sensitive association rules
output: Modified database to hide association rules
rulesdataset = apriori (Dataset, parameter = list(supp = minsupp, conf = minconf,
  minlen = 2));
inspect (rulesdataset);
ruletohide = subset (rulesdataset);
ruletohidec = |ruletohide|;
for o ← 1 to ruletohidec do
    aff ← affinity(dataset);
    lhs ← selectLhs(ruletohide [o]);
    rhs ← selectRhs(ruletohide [o]);
    lhslist = {t | tε transactions and t fully support LHS of rule};
    rhslist = {t | tε transactions and t fully support RHS of rule};
    TransactionHybrid ← intersect(lhslist, rhslist);
    // prepare candidate transaction list in TransactionHybrid by intersecting
       lhslist and rhslist
    TosorttransactionHybrid ← NULL;
    for i Input: TransactionHybrid
     do
        kl ← o;
        kr ← o;
        for all frequent items j Input: items
         do
            if j ≠ lhs and j ≠ rhs and i contains j then
                kl = kl + aff[lhs, j];
                kr = kr + aff[rhs, j];
            end
            TosorttransactionHybrid = concate(TosorttransactionHybrid, concate(i, kl, kr))
               // TosorttransactionHybrid contain candidate transaction with their
               respective affinity sum with lhs and affinity sum with rhs
        end
    end
    SortedLHSHybrid ← sort(TosorttransactionHybrid over (i, kl));
    NewMatrixOrderedLHS ← as(SortedLHSHybrid, "Matrix");
    // Sorted candidate transaction by affinity sum with LHS
    SortedRHSHybrid ← sort(TosorttransactionHybrid over (i, kr));
    NewMatrixOrderedRHS ← as(SortedRHSHybrid, "Matrix");
    // Sorted candidate transaction by affinity sum with RHS
    CMRCBMST ← ⌈(|TransactionHybrid| − (|Dataset| ∗ MST/100))⌉;
    CMRCBMCT ← ⌈(|TransactionHybrid| − (|lhslist| ∗ MCT/100))⌉;
    NoOfModificationinHybrid ← minimum(CMRCBMST, CMRCBMCT);
    if CMRCBMST ≤ CMRCBMCT then
        NoHybrid(NoOfModificationinHybrid, Dataset, SortedRHSHybrid, rhs)
    end
    else
        Call HybridCode(NoOfModificationinHybrid, Dataset, NewMatrixOrderedLHS,
          NewMatrixOrderedRHS, lhs, rhs)
    end
end
call CheckPerformance(Dataset, rulesdataset);
```

Algorithm 3. AffinityHybrid algorithm

```
1  kl ← 1;
   // Index for sorted transaction list having transaction ID and
      Transaction affinity with victim-item as LHS
2  kr ← 1;
   // Index for sorted transaction list having transaction ID and
      Transaction affinity with victim-item as RHS
3  NoofModificationsDoneinDB ← 0;
4  NoofModificationsDoneinLHS ← 0;
5  NoofModificationsDoneinRHS ← 0;
6  while NoofModificationsDoneinDB < NoOfModificationinHybrid do
       // Transaction is selected by comparing the two sorted list
          prepared
7      if NewMatrixOrderedLHS [kl, 2] < NewMatrixOrderedRHS [kr, 2] then
8          Dataset [(NewMatrixOrderedLHS [kl, 1]),Lhs ] ← 0;
           // transaction picked from sorted list prepared by having
              victim-item as LHS
9          x ← which(NewMatrixOrderedRHS [, 1] == NewMatrixOrderedLHS
            [kl, 1]);
10         NewMatrixOrderedRHS ← NewMatrixOrderedRHS [−x, ];
           // Transaction removed from sorted list prepared by
              having victim-item as RHS, as it is already sanitized
11         NoofModificationsDoneinLHS ← NoofModificationsDoneinLHS + 1;
12         NoofModificationsDoneinDB ← NoofModificationsDoneinDB + 1;
13         kl ← kl + 1;
14     end
15     else
16         Dataset [(NewMatrixOrderedRHS [kr, 1]), Rhs ] ← 0;
           // transaction picked from sorted list prepared by having
              victim-item as RHS
17         x ← which(NewMatrixOrderedLHS [, 1] == NewMatrixOrderedRHS
            [kr, 1]);
18         NewMatrixOrderedLHS ← NewMatrixOrderedLHS [−x, ];
           // Transaction removed from sorted list prepared by
              having victim-item as LHS, as it is already sanitized
19         NoofModificationsDoneinRHS ← NoofModificationsDoneinRHS + 1;
20         NoofModificationsDoneinDB ← NoofModificationsDoneinDB + 1;
21         kr ← kr + 1;
22     end
23 end
```

Algorithm 4. HybridCode algorithm

```
for i ← 1 to NoOfModificationinHybrid do
    NoofModificationsDoneinDB ← NoofModificationsDoneinDB + 1;
    Dataset [m in SortedRHSHybrid, Rhs ] ← 0;
end
Return(Dataset);
```

Algorithm 5. NoHybrid algorithm

```
RulesNew = Apriori (dataset, parameter = list(supp = minsupp,
  conf = minconf, minlen = 2));
Inspect (RulesNew);
GhostRules ← SetDiff(RulesNew, rulesdataset);
LostRules ← SetDiff(rulesdataset, RulesNew);
GhostRulesCount ← Length(SetDiff(RulesNew, rulesdataset));
LostRulesCount ← Length(SetDiff(rulesdataset,
  RulesNew)) − ruletohidec;
```

Algorithm 6. Performance After Hybrid algorithm

Let x be the number of modification required for hybrid(1). Then one by one transaction is selected by comparing the two sorted lists prepared, and transaction having the least affinity is modified with the condition that if transaction has been picked from sorted affinity list of the transaction for the left-hand side then the left-hand side of the rule gets removed from transaction otherwise if the transaction is picked from sorted affinity of the transaction for the right-hand side, then the right-hand side of the rule gets removed. Once the transaction picked from the one list, it cannot be picked again from the second list since if the same transaction is used for removing the left-hand side and the right-hand side – this does not add any benefit to the approach.

Let X and Y be present on the left-hand side and the right-hand side of the rule. For reducing the support below MST either X or Y is sufficient to remove the transaction to reduce the support of the overall rule. The rationale for a hybrid approach is that changing only one side of the rule would result in a large reduction in its support, which will have more unintended consequences that can be managed by taking into account both the left-hand and right-hand sides of the rule. Additionally, the side-effect is diminished because the choice of transaction is fully chosen based on how it would affect another frequently used itemset.

3 COMPUTATIONAL EXPERIMENTS AND RESULTS

Approaches present in the literature for hiding the rule fall into two broad categories viz

1. hiding a large itemset,

2. hiding an association rule directly.

It is more complicated to hide rules in comparison to hiding itemsets. This paper presents five approaches that hide sensitive association rule by directly working on hiding rules as it gives the database owner more control. Latest work that has been done on association rule hiding hides large itemsets to preserve the privacy in the database like the border based approach. Greedy approaches [26] (2013) hides a sensitive association rule by increasing the number of transactions that will greatly affect the database size as well as a lot of computation needed to add the items to the added transaction. So, we evaluate our work against algorithms that fall under the category of heuristic algorithms. The algorithms used for comparison are Algo 1.a [4], Algo 1.b [4], MinFIA [2], MaxFIA [2] and Naive [2]. To evaluate performance, we ran two experiments. In the first setup, experiments were performed with dataset generated by IBM Synthetic data generator where database size is ranging from 10 K to 100 K. In the second setup, experiments were performed with real-world datasets downloaded from UCI repository and fimi.

3.1 Experiment Setup 1

We have performed experiments on a computer with a core-i7 processor, 8 GB RAM running on Windows 10 Operating System. The datasets used in the assessment trials are generated using IBM synthetic data generator [27]. The database size employed in the dataset range from 10 K to 100 K with the average transaction length, $|\text{ATL}| = 5$, and a total number of items is 50. The minimum support threshold picked is 4 % and the minimum confidence threshold picked is 20 %. In the series of experiments, database size ranging from 10 K to 100 K is generated ten times and arbitrary five rules are selected for hiding. All the graphs were plotted to represent the average of 10 iterations of experiments. The language used for implementation is *R* [28]. To evaluate the performance of the algorithms following side-effects are considered:

1. Rule Hiding Failure (RHF);
2. Rule Falsely Generated or Ghost Rules (GR) (Also known as Artifactual Patterns (AF));
3. Rules Falsely Hidden or Lost Rules (LR);
4. Dissimilarity measure.

The rule hiding failure side-effect counts the number of sensitive association rules; the algorithm fails to hide. Rule falsely generated (Ghost rules) side-effect counts the number of rules that were not available with the original dataset, but after the modifications performed by the algorithm, the rule appears. The rules falsely hidden (Lost rules) side-effect counts the number of nonsensitive rules hided because of the data distortion process. Comparisons were made with Algo1.a, Algo1.b, MinFIA, MaxFIA and naive algorithm against various database sizes ranging from 10 K to 100 K.

Hiding failure is 0 in all algorithms except Algo 1.a. So Algo 1.a is not considered in analyzing other side-effects.

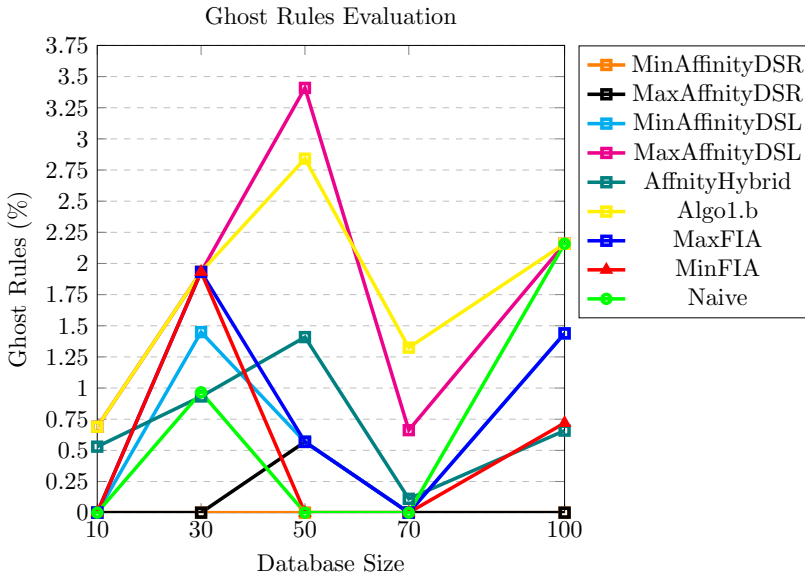


Figure 2. Performance of Algorithms with respect to Ghost Rules

Figures 2 and 3 account for the ghost rule and lost rule side-effect evaluation of algorithms, respectively. Table 7 represents data for the ghost rule and lost rule side-effect in a tabular form, where each average value is accompanied by a value of the standard error.

It is depicted in Figure 2 that MinAffinityDSR algorithm is free from the ghost rule side-effect. It never generates ghost rules in all our experiment trials. MaxAffnityDSR also performs well in the ghost rule side-effect. It suffered from this side-effect only once.

It is clear from Figure 3 that MinAffinityDSR, MaxAffinityDSL and AffinityHybrid algorithm performs best with lost rules side-effects. Less rules lost means more is the utility of the database.

Dissimilarity measure is based upon the count of the frequency of the items before the sanitization algorithm and after the sanitization algorithm, i.e., to measure the frequencies of the items in the original database and the released database. Dissimilarity measure evaluation is shown in Figure 4. Table 8 represents data for Dissimilarity measure evaluation in a tabular form, where each average value is accompanied by a value of the standard error. It is clear from the graph that AffinityHybrid outperforms all algorithms used in experiments for comparison. MinAffinityDSR, MaxAffinityDSR and MaxAffinityDSL also has good results and Naive algorithm performance was the weakest with respect to dissimilarity measure.

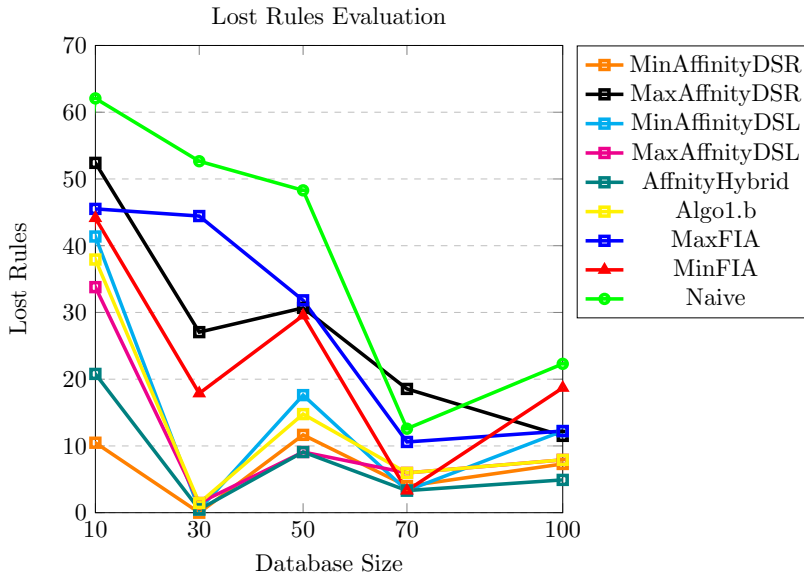


Figure 3. Performance of Algorithms with respect to Lost Rules

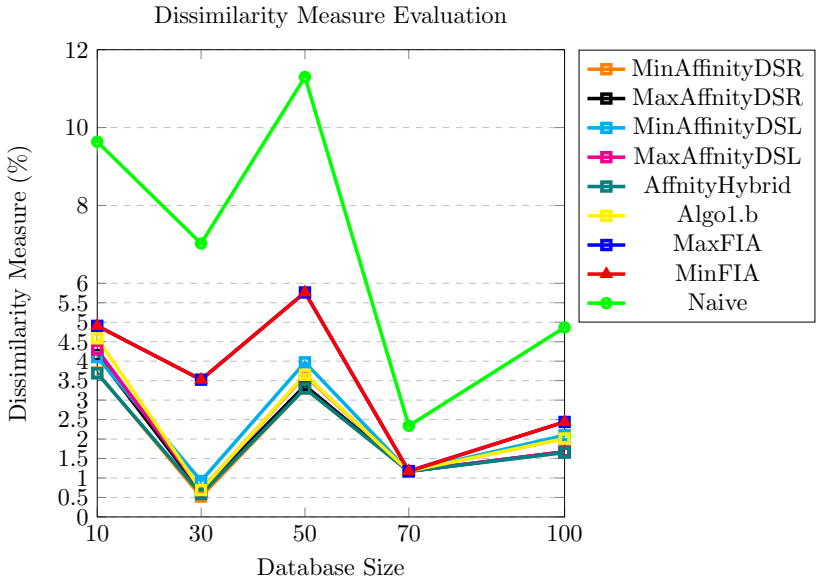


Figure 4. Performance of Algorithms with Dissimilarity Measure

Algorithm	Database Size	Ghost Rules (GR)	Standard Error (GR)	Lost Rules (LR)	Standard Error (LR)
MinAffinityDSR	10	0	0	10.48	2.82
	30	0	0	0	0.2
	50	0	0	11.66	3.2
	70	0	0	3.97	0.67
	100	0	0	7.3	1.56
MaxAffinityDSR	10	0	0	52.41	4.81
	30	0	0	27.05	3.22
	50	0.57	0.01	30.68	1.56
	70	0	0	18.54	3.39
	100	0	0	11.51	1.22
MinAffinityDSL	10	0	0	41.38	4.29
	30	1.45	0.11	0.48	0.55
	50	0.57	0.12	17.61	4.99
	70	0	0	3.31	2.23
	100	1.44	0.49	12.23	4.81
MaxAffinityDSL	10	0.69	0.66	33.79	1.64
	30	1.93	1.21	1.45	2.59
	50	3.41	0.23	9.09	3.58
	70	0.66	0.47	5.96	2.47
	100	2.16	1.09	7.91	3.29
Affinityhybrid	10	0.53	0.04	20.79	1.01
	30	0.93	0.37	0.48	1.2
	50	1.41	0.01	9.09	0.25
	70	0.11	0.01	3.31	1.23
	100	0.66	0.78	4.91	0.45
Algo 1.b	10	0.69	0.16	37.93	0.68
	30	1.93	0.19	1.45	1.26
	50	2.84	1.01	14.77	2.32
	70	1.32	1.07	5.96	1.67
	100	2.16	1.22	7.91	0.44
MaxFIA	10	0	0	45.52	4.25
	30	1.93	1.19	44.44	2.25
	50	0.57	0.09	31.82	3.29
	70	0	0	10.6	4.17
	100	1.44	0	12.23	2.38
MinFIA	10	0	0	44.14	1.82
	30	1.93	0.08	17.87	0.17
	50	0	0	29.55	3.68
	70	0	0	3.31	4.25
	100	0.72	0.49	18.71	1.19
Naive	10	0	0	62.07	2.28
	30	0.97	0.11	52.66	4.28
	50	0	0	48.3	4.66
	70	0	0	12.58	2.38
	100	2.16	0.19	22.3	2.87

Table 7. Performance of Algorithms – Ghost Rules and Lost Rules (With Standard Error)

3.2 Experiment Setup 2

We have performed performance evaluation experiments on a PC with a core-i7 processor, 8 GB RAM running on Windows 10 Operating System and the language used for implementation is R Language.

We tested the proposed algorithm on three real representative databases. One is a mushroom [29] (descriptions of hypothetical samples corresponding to 23 species

Algorithm	Database Size	Dissimilarity	Standard Error
MinAffinityDSR	10	3.70819848975189	1.52
	30	0.52254906665387	2.31
	50	3.30037822918644	0.65
	70	1.17524852996062	0.58
	100	2.01695689527802	0.29
MaxAffinityDSR	10	4.15110779188449	2.28
	30	0.64602428139546	3.47
	50	3.37266527447172	1.69
	70	1.17055066774891	2.59
	100	1.67285940210462	3.57
MinAffinityDSL	10	4.10339390921915	1.22
	30	0.92254906665387	0.52
	50	3.96994611329351	0.47
	70	1.17524852996062	1.66
	100	2.09695689527802	2.59
MaxAffinityDSL	10	4.29839847315575	3.69
	30	0.68561351312443	2.48
	50	3.63516217123998	3.58
	70	1.17524852996062	2.46
	100	1.67285940210462	3.46
Affinityhybrid	10	3.68819848975189	2.59
	30	0.5882	3.57
	50	3.30037822918644	2.58
	70	1.17055066774891	2.69
	100	1.65	4.24
Algo 1.b	10	4.58675628578541	0.66
	30	0.687413023657565	2.25
	50	3.64830527038276	3.28
	70	1.17524852996062	2.69
	100	2.01414387940674	2.47
MaxFIA	10	4.90726910629823	2.56
	30	3.52464129756706	3.58
	50	5.76507440454459	3.98
	70	1.17524852996062	2.58
	100	2.43897100267592	3.69
MinFIA	10	4.90726910629823	3.57
	30	3.52464129756706	2.49
	50	5.76507440454459	3.74
	70	1.17524852996062	4.25
	100	2.43897100267592	1.12
Naive	10	9.63612978176085	2.28
	30	7.02169010029272	3.48
	50	11.2986842297414	1.69
	70	2.33953538142726	2.49
	100	4.86772718401089	1.66

Table 8. Performance of Algorithms – Dissimilarity (With Standard Error)

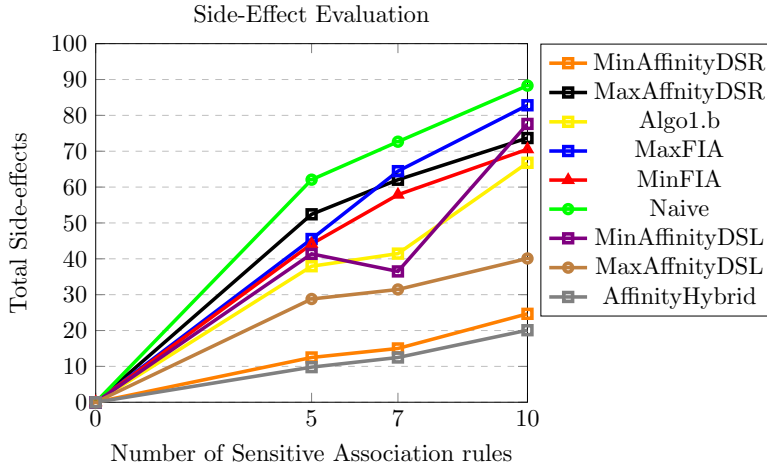


Figure 5. Performance on real datasets – Mushroom, BMS-WebView-1, BMS-WebView-2

of gilled mushrooms in the Agaricus and Lepiota Family (pp. 500–525). Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended), which was prepared by Roberto Bayardo and is publicly available through the FIMI repository website located at `http://fimi.ua.ac.be/data/` (Frequent Itemset Mining Dataset Repository). The other datasets were BMS-WebView-1 (downloaded from [30]) from Blue Martini Software Inc. that were used for the KDD Cup of 2000. This dataset contains 59,601 sequences of clickstream data from an e-commerce. It contains 497 distinct items. The average length of sequences is 2.42 items with a standard deviation of 3.22. In this dataset, there are some long sequences. For example, 318 sequences contain more than 20 items. Another dataset used is BMS-WebView-2 (downloaded from [30]) which is a second dataset used in the KDD-CUP 2000 competition. It contains 77 512 sequences of click-stream data. It contains 3340 distinct items. The average length of sequences is 4.62 items with a standard deviation of 6.07 items. The thresholds of minimum support were appropriately set to ensure an adequate amount of frequent itemsets. Comparisons were made with state-of-art approaches (Algo 1.a and Algo 1.b), MinFIA, MaxFIA and Naive, and the results are summarized in Figure 5. The graph represents the average of total side-effects generated on all three datasets when number of sensitive association rule is varied from 0 to 10. The result obtained is similar to the case when experiments are performed with datasets generated from IBM Synthetic data generator. AffinityHybrid algorithm gives the best solution. It can also be concluded that while hiding the rule by reducing the support of right-hand side of rule, transactions must be selected in increasing value of affinity, i.e., MinAffinityDSR is preferred. If the rule is to be hided by decreasing the support of the left-hand side of the rule, transactions must be selected in decreasing value of affinity, i.e., MaxAffinityDSL is pre-

ferred.

4 ANALYSIS AND DISCUSSION

4.1 Accuracy

In association rule hiding techniques, the accuracy of the heuristic algorithms depends on the victim item as well as the victim transaction. In [4] approaches selects the victim transaction by count of itemsets present in the transaction. Picking the transaction in this fashion can significantly increase side-effects. Let there are four transactions all having precisely four itemsets, then as per [4], all four will be considered equally as the candidate transaction to be modified.

Proposed strategy while selecting a potential transaction, takes into account the effects of modifying it. This is done by the concept of affinity of transaction introduced in the paper. Also in [4], approaches either select victim item from the left-hand side of the rule or right-hand side of the rule.

In proposed hybrid approach, victim item, as well as victim transaction both are selected on the basis of transaction affinity calculated exclusively for the left-hand side as well as the right-hand side of the sensitive association rule.

4.2 Effect of MST and MCT

For hiding sensitive association rule, either support of the rule should be below the minimum support threshold (MST), or confidence should be below minimum confidence threshold (MCT). The number of modification increases as MST and MCT selected by the data owner decreases.

It is identified by experimental results that proposed approaches performed better not only with the higher value of MST and MCT but also with low range values. Although the side-effect may get an increase, as the too low value of MST and MCT is considered by data owner, while the optimal sanitization in association rule hiding belongs to the class of NP-Complete problems.

4.3 Number of Modifications

1. The number of modifications to be done in MAXAffinityDSR and MINAffinityDSR can be identified by:

$$NM_{DSR} = \left\lceil MIN(|\mathbb{T}_{id}(S_r)| - \frac{|\mathbb{T}_{id}| * MST}{100}, |\mathbb{T}_{id}(S_r)| - \frac{|\mathbb{T}_{id}(L_{S_r})| * MCT}{100}) \right\rceil . \tag{10}$$

   $\mathbb{T}_{id}(S_r)$ or $(\sum X \cup Y)$ and $\mathbb{T}_{id}(L_{S_r})$ contains the set of all transactions containing X ∪ Y and X respectively. $|\mathbb{T}_{id}|$ is the total number of transactions. To hide X → Y, removing items in Y from the transactions will decrease $\text{support}_{X \cup Y}$ i.e., it

will reduce the number of transactions supporting the rule by deleting elements from transactions present on the right-hand side of the rule. Let NM_{DSR} (θ), an integer, be number of modified transactions when the rule $X \rightarrow Y$ is hidden. This make either support is reduced below MST as defined in Equation 11 or confidence reduced below MCT as defined in Equation 12 which is sufficient to hide the sensitive association rule.

$$\frac{|\mathfrak{T}_{id}(S_r)| - \theta}{|\mathfrak{T}_{id}|} < \frac{MST}{100}, \tag{11}$$

$$\frac{|\mathfrak{T}_{id}(S_r)| - \theta}{|\mathfrak{T}_{id}(L_{S_r})|} < \frac{MCT}{100}. \tag{12}$$

Hence, the number of modifications required is minimum value of θ to satisfy either Equations (11) or (12).

2. The number of modifications in MAXAffinityDSL and MINAffinityDSL is identified by:

$$NM_{DSL} = \left\lceil |\mathfrak{T}_{id}(S_r)| - \frac{|\mathfrak{T}_{id}| * MST}{100} \right\rceil. \tag{13}$$

$\mathfrak{T}_{id}(S_r)$ or $(\sum X \cup Y)$ contains the set of all transactions containing $X \cup Y$. To hide $X \rightarrow Y$, removing items in X from the transactions will decrease the $\text{support}_{X \cup Y}$ i.e., it will reduce the number of transactions supporting the rule by deleting elements from transactions present on left-hand side of rule. Let NM_{DSL} (θ), an integer, be number of modified transactions when rule $X \rightarrow Y$ is hidden. This makes support to reduce below MST as defined in Equation (14) which eventually hide the sensitive association rule.

$$\frac{|\mathfrak{T}_{id}(S_r)| - \theta}{|\mathfrak{T}_{id}|} < \frac{MST}{100}. \tag{14}$$

3. The third approach AffinityHybrid combines the above two approaches to have a mixture of reducing the support of a subset of the left-hand side and right-hand side of the sensitive association rule. Therefore, number of modification can be identified by:

$$NM_H = \left\lceil MIN\left(|\mathfrak{T}_{id}(S_r)| - \frac{|\mathfrak{T}_{id}| * MST}{100}, |\mathfrak{T}_{id}(S_r)| - \frac{|\mathfrak{T}_{id}(L_{S_r})| * MCT}{100} \right) \right\rceil. \tag{15}$$

4.4 Results Summary

- MinAffinityDSR algorithm never generates ghost rules.

- Algo 1.b perform worst in case of ghost rules side-effects.
- MaxAffinityDSR and MinAffinityDSR generate good results with dissimilarity measure.
- Ghost rule percentage is low in comparison to lost rule percentage in case of all algorithms as the maximum percentage of ghost rule side effect is under 3.5. This shows that lost rule plays a primary concern in evaluating the performance of the algorithm.
- MaxAffinityDSL and MinAffinityDSL performance were better in comparison to Algo 1.a, Algo 1.b, MinFIA, MaxFIA and naive algorithm regarding lost rule side-effect. Also, both perform better with dissimilarity measure.
- AffinityHybrid algorithm outperforms all the algorithms used for comparison with the lost rule, ghost rule and dissimilarity measure.

5 CONCLUSION

This work proposes five new algorithms based on modifying transactions by considering the side effect, by calculating affinity sum of victim items with other frequent items present in the transaction. The experimental result clearly shows the fruitfulness of the approach. Experiments suggest that proposed approach not only outperforms Algo 1.a, Algo 1.b, MINFia, MaxFIA and NAive algorithm regarding dissimilarity measure but at the same time, side-effects have been reduced. A drawback of the proposed algorithm is its running time. Algorithm performs a bit slower in comparison to other algorithms when experiments performed with large databases. The reason for a slow speed is that the affinity calculation time increases with database size. Among the five algorithms presented in the paper, AffinityHybrid algorithm gives the best solution. It can also be concluded that while hiding the rule by reducing the support of the right-hand side of the rule, transactions must be selected in the increasing value of affinity, i.e., MinAffinityDSR is preferred. If the rule is to be hided by the decreasing the support of the left-hand side of the rule, transactions must be selected in decreasing value of affinity, i.e., MaxAffinityDSL is preferred. If there is no restriction on selecting the victim item from the rule, the AffinityHybrid algorithm is preferred as it is the top performer among the set of algorithms discussed in the paper.

REFERENCES

[1] Clifton, C.—Marks, D.: Security and Privacy Implications of Data Mining. ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, Citeseer, 1996, pp. 15–19.

[2] Oliveira, S. R. M.—Zaiane, O. R.: Privacy Preserving Frequent Itemset Mining. In: Clifton, C., Estivill-Castro, V. (Eds.): IEEE ICDM Workshop on Privacy, Security and Data Mining. ACS, Maebashi City, Japan, CRPIT, Vol. 14, 2002, pp. 43–54.

[3] Sun, X.—Yu, P.: A Border-Based Approach for Hiding Sensitive Frequent Itemsets. Fifth IEEE International Conference on Data Mining (ICDM'05), 2005, 8 pp., doi: 10.1109/ICDM.2005.2.

[4] Verykios, V.—Elmagarmid, A.—Bertino, E.—Saygin, Y.—Dasseni, E.: Association Rule Hiding. IEEE Transactions on Knowledge and Data Engineering, Vol. 16, 2004, No. 4, pp. 434–447, doi: 10.1109/TKDE.2004.1269668.

[5] Oliveira, S.—Zaiane, O.: Protecting Sensitive Knowledge by Data Sanitization. Third IEEE International Conference on Data Mining, 2003, pp. 613–616, doi: 10.1109/ICDM.2003.1250990.

[6] Pontikakis, E.—Tsitsonis, A.—Verykios, V.: An Experimental Study of Distortion-Based Techniques for Association Rule Hiding. Research Directions in Data and Applications Security XVIII, 2004, pp. 325–339, doi: 10.1007/1-4020-8128-6_22.

[7] Jain, D.—Khatri, P.—Soni, R.—Chaurasia, B. K.: Hiding Sensitive Association Rules Without Altering the Support of Sensitive Item(s). Advances in Computer Science and Information Technology. Networks and Communications, 2012, pp. 500–509, doi: 10.1007/978-3-642-27299-8_52.

[8] Lin, Y.—Wang, E.—Lee, G.: A Novel Method for Protecting Sensitive Knowledge in Association Rules Mining. Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC 2005), IEEE Computer Society, Los Alamitos, CA, USA, Vol. 1, 2005, pp. 511–516, doi: 10.1109/COMPSAC.2005.27.

[9] Sun, X.—Yu, P. S.: Hiding Sensitive Frequent Itemsets by a Border-Based Approach. Journal of Computing Science and Engineering, Vol. 1, 2007, No. 1, pp. 74–94, doi: 10.5626/jcse.2007.1.1.074.

[10] Moustakides, G. V.—Verykios, V. S.: A Maxmin Approach for Hiding Frequent Itemsets. Data and Knowledge Engineering, Vol. 65, 2008, No. 1, pp. 75–89, doi: 10.1016/j.datak.2007.06.012 (Including Special Section: Privacy Aspects of Data Mining Workshop (2006) – Five Invited and Extended Papers).

[11] Gkoulalas-Divanis, A.—Verykios, V.: A Hybrid Approach to Frequent Itemset Hiding. 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007), Vol. 1, 2007, pp. 297–304, doi: 10.1109/ICTAI.2007.68.

[12] Quoc Le, H.—Arch-Int, S.—Arch-Int, N.: Association Rule Hiding Based on Intersection Lattice. Mathematical Problems in Engineering, Vol. 2013, 2013, doi: 10.1155/2013/210405.

[13] Gkoulalas-Divanis, A.—Verykios, V. S.: An Integer Programming Approach for Frequent Itemset Hiding. Proceedings of the 15th ACM International Conference on Information and Knowledge Management – CIKM'06, ACM Press, 2006, doi: 10.1145/1183614.1183721.

[14] Guo, Y.: Reconstruction-Based Association Rule Hiding. Proceedings of SIGMOD 2007 Ph.D. Workshop on Innovative Database Research, Vol. 2007, 2007, pp. 51–56.

[15] Chen, X.—Orlowska, M.—Li, X.: A New Framework of Privacy Preserving Data

Sharing. Proceedings of the 4th IEEE ICDM Workshop: Privacy and Security Aspects of Data Mining. IEEE Computer Society, Citeseer, 2004, pp. 47–56.

[16] Wang, Y.—Wu, X.: Approximate Inverse Frequent Itemset Mining: Privacy, Complexity, and Approximation. Fifth IEEE International Conference on Data Mining (ICDM '05), 2005, 8 pp., doi: 10.1109/ICDM.2005.27.

[17] Guo, Y.—Tong, Y.—Tang, S.—Yang, D.: A FP-Tree-Based Method for Inverse Frequent Set Mining. In: Bell, D. A., Hong, J. (Eds.): Flexible and Efficient Information Handling. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 152–163, doi: 10.1007/11788911_13.

[18] Ahmed, G.—Abd_Ellatif, L.—Sharaf, A.: Association Rules Hiding for Privacy Preserving Data Mining: A Survey. International Journal of Computer Applications, Vol. 150, 2016, No. 12, pp. 34–43, doi: 10.5120/ijca2016911664.

[19] Vaidya, J.—Clifton, C.: Privacy Preserving Association Rule Mining in Vertically Partitioned Data. Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, New York, NY, USA, KDD '02, 2002, pp. 639–644, doi: 10.1145/775047.775142.

[20] Vaidya, J.—Clifton, C.: Secure Set Intersection Cardinality with Application to Association Rule Mining. J. Comput. Secur., Vol. 13, 2005, No. 4, pp. 593–622, doi: https://dl.acm.org/doi/10.5555/1239367.1239368.

[21] Zhong, S.: Privacy-Preserving Algorithms for Distributed Mining of Frequent Itemsets. Information Sciences, Vol. 177, 2007, No. 2, pp. 490–503, doi: 10.1016/j.ins.2006.08.010.

[22] El-Sisi, A.: Fast Cryptographic Privacy Preserving Association Rules Mining on Distributed Homogenous Database. Int. Arab J. Inf. Technol., Vol. 7, 2010, No. 2, pp. 152–160.

[23] Kaosar, M. G.—Paulet, R.—Yi, X.: Fully Homomorphic Encryption Based Two-Party Association Rule Mining. Data and Knowledge Engineering, Vol. 76–78, 2012, pp. 1–15, doi: 10.1016/j.datak.2012.03.003.

[24] Alborzi, S.—Raji, F.—Saraee, M.: Privacy Preserving Mining of Association Rules on Horizontally Distributed Databases. International Conference on Software and Computer Applications ICSCA 2012, IACSIT Press, Singapore, 2012, pp. 158–164, http://usir.salford.ac.uk/id/eprint/42930/.

[25] Aggarwal, C.—Procopiuc, C.—Yu, P.: Finding Localized Associations in Market Basket Data. IEEE Transactions on Knowledge and Data Engineering, Vol. 14, 2002, No. 1, pp. 51–62, doi: 10.1109/69.979972.

[26] Lin, C. W.—Hong, T. P.—Chang, C. C.—Wang, S. L.: A Greedy-Based Approach for Hiding Sensitive Itemsets by Transaction Insertion. J. Inf. Hiding Multim. Signal Process., Vol. 4, 2013, No. 4, pp. 201–214.

[27] Agrawal, R.—Srikant, R.: Fast Algorithms for Mining Association Rules in Large Databases. In: Bocca, J. B., Jarke, M., Zaniolo, C. (Eds.): VLDB '94, Proceedings of 20th International Conference on Very Large Data Bases, September 12–15, 1994, Santiago De Chile, Chile. Morgan Kaufmann, 1994, pp. 487–499, http://www.vldb.org/conf/1994/P487.PDF.

[28] R Core Team: R: A Language and Environment for Statistical Computing. R Foun-

dation for Statistical Computing, Vienna, Austria, 2017, https://www.R-project.org/.

[29] Mushroom. 1987, doi: 10.24432/C5959T.

[30] Fournier-Viger, P.—Lin, J. C. W.—Gomariz, A.—Gueniche, T.—Soltani, A.—Deng, Z.—Lam, H. T.: The SPMF Open-Source Data Mining Library Version 2. In: Berendt, B., Bringmann, B., Fromont, É., Garriga, G., Miettinen, P., Tatti, N., Tresp, V. (Eds.): Machine Learning and Knowledge Discovery in Databases. Springer International Publishing, Cham, 2016, pp. 36–40, doi: 10.1007/978-3-319-46131-1_8.

Kshitij Pathak is a Lecturer in the Department of Computer Science and Engineering at Government Polytechnic College Mandsaur, MP, India. He has been awarded a Ph.D. degree in computer science and engineering. He has published 42 papers in International and National journals and conferences. His research interest includes data mining, information security and artificial intelligence. He is a life member of the Computer Society of India and the Institution of Engineers India.

Sanjay Silakari is Professor in the Department of Computer Science and Engineering at the University Institute of Technology, RGPV Bhopal, MP, India. He has more than two decades of teaching and administrative experience and has guided several students in their doctoral and master's studies. He has several research publications to his credit in different reputed national and international conferences and journals. His areas of interest include network security, web engineering, web personalization and search engines, operating systems, computer networks and e-commerce. He is a life member of ISTE, CSI, and IAENG and a member of IEEE and ACM. He is the author of the book Basic Computer Engineering.

Narendra S. Chaudhari is Professor (HAG) in the Computer Science and Engineering Department at Indian Institute of Technology (IIT), Indore (M.P.), India. He has done significant research work on game AI, novel neural network models and security of the wireless mobile communication. He has been referee and reviewer for a number of premier conferences and Journals including IEEE Transaction, Neurocomputing, etc. Also, he is fellow and recipient of Eminent Engineer Award (Computer Engineering) of the Institution of Engineers, India (IE-India), as well as fellow of the Institution of Electronics and Telecommunication Engineers (IETE) (India), senior member of Computer Society of India, senior member of IEEE (USA), member of Indian Mathematical Society (IMS), Cryptology Research Society of India (CRSI) and many other professional societies.

Computing and Informatics, Vol. 42, 2023, 311–339, doi: 10.31577/cai_2023_2_311

A PROPOSED SCHEDULING ALGORITHM FOR IOT APPLICATIONS IN A MERGED ENVIRONMENT OF EDGE, FOG, AND CLOUD

Xuan Thi Tran

University of Information and Communication Technology
Thai Nguyen University
Z115, Quyet Thang, Thai Nguyen, Vietnam
e-mail: ttxuan@ictu.edu.vn

Abstract. With the rapid increase of Internet of Things (IoT) devices and applications, the ordinary cloud computing paradigm soon becomes outdated. Fog computing paradigm extends services provided by a cloud to the edge of network in order to satisfy requirements of IoT applications such as low latency, locality awareness, low network traffic, mobility support, and so forth. Task scheduling in a Cloud-Fog environment plays a great role to assure diverse computational demands are met. However, the quest for an optimal solution for task scheduling in the such environment is exceedingly hard due to diversity of IoT applications, heterogeneity of computational resources, and multiple criteria. This study approaches the task scheduling problem with aims at improving service quality and load balancing in a merged computing system of Edge-Fog-Cloud. We propose a Multi-Objective Scheduling Algorithm (MOSA) that takes into account the job characteristics and utilization of different computational resources. The proposed solution is evaluated in comparison to other existing policies named LB, WRR, and MPSO. Numerical results show that the proposed algorithm improves the average response time while maintaining load balancing in comparison to three existing policies. Obtained results with the use of real workloads validate the outcomes.

Keywords: Cloud-fog computing, job service demand, load balance, job scheduling

Mathematics Subject Classification 2010: 68-M20

1 INTRODUCTION

With the rapid increase of Internet of Things (IoT) and its applications, the ordinary cloud computing paradigm faces a challenge to satisfy applications that require frequent data access, low latency, real-time interaction, high-speed communication, and so forth. Edge computing was introduced to support task computation on source devices of generated data. Generally, computational capacity is limited on edge nodes. Fog computing appears as an effective complement of Cloud center to cope with those issues by extending cloud services to the edge network [1, 2, 3, 4, 5].

Task scheduling in such distributed environment plays a great role to assure that diverse computational demands are met. However, the quest for an efficient and effective solution in merged computing environments is exceedingly hard due to diversity of IoT applications, heterogeneity of computational resources, and multiple criteria [3, 6, 7, 8]. In the literature, task scheduling has been studied with various desired factors of effective processing [9, 10, 11, 12], load balancing [13, 14] and/or power efficiency [15, 16, 17].

Energy cost contributes a significant factor to overall operation cost of large-scale computational systems. The use of dynamic power management (DPM) techniques has been crucial to achieve energy efficiency as addressed in [15, 16, 18] and references therein. Switching off technique (that is, switching idle servers off and only turning them back on when they are needed) has been addressed as an effective method of power/energy consumption management of Cloud systems [13, 15, 16, 17, 19, 20].The application of switching off technique in a merged, heterogeneous computing environment of Edge, Fog, Cloud has been studied in [13]. Authors proposed the use of load thresholds of computing resources in a subsystem to determine the number of active servers in the subsystem.

Due to the diversity of applications and the resource heterogeneity, we argue that characteristics of both user jobs and resources play a role in the system cost and performance. Therefore, scheduling policy should take into account job characteristics and resource capacity in order to improve the performance. In addition, we follow the resource utilization based approach given in [13] for load balance and energy efficiency. The contributions of this work are highlighted in what follows.

- This study argues that job service demand and resource processing capacity play a role for efficient computation;
- Load threshold based policy helps to avoid load stress at power-sensitive machines;
- Numerical results (both theoretical and traced workloads) address that our proposed algorithm improves both performance and load balance of a Cloud-Fog computing system, in comparison with three other existing algorithms named Weighted Round Robin, Load-Balance, and Modified Particle Swarm Optimization (MPSO) based heuristic.

- The energy cost of Cloud center is reduced with Load-Balance and our proposed algorithm at low load, while our proposal yields lower energy cost of Raspberry Pi cluster than Load-Balance does.

The paper is organized as follows. Section 2 gives a review on the literature works. Section 3 describes the system model and the proposed scheduling solution. Section 4 presents obtained numerical results and discussions. Finally, Section 5 concludes the paper.

2 RELATED WORK

This section is a brief review on the related works of task scheduling in Cloud-Fog computing environments. In [9], authors proposed a batch-mode task scheduling algorithm based on the relationship between a fog node set and a task set. Compared with existing batch-mode scheduling algorithms (MCT, MET, MIN-MIN), the proposal yields a shorter total completion time of tasks. Rafique et al. [15] focused on balancing task execution time and energy consumption at Fog layer computing resources; the proposed NBIHA algorithm resulted in resource utilization, average response time, and energy consumption but an increase in task execution time. A time-cost based scheduling algorithm was introduced in [11], wherein authors applied a set of modifications of Genetic Algorithm in their proposal. They showed that the time-cost based algorithm achieves a better trade-off between time and cost execution.

Xiang et al. [21] proposed a solution for mode selection and resource allocation with the aim to maximize the energy efficiency of the fog-RAN system. The proposed algorithm that is based on particle swarm optimization leads to energy savings – delay trade-off. Oueis et al. [22] introduced three variants of the algorithm that clusters small cells into computational clusters to process the users' requests, they and indicated that the power-centric solution results in a low energy consumption per user. In [23], a workload allocation policy was proposed to solve the trade-off problem between job execution delay and energy consumption.

Agarwal et al. [24] proposed an algorithm that allows efficiently distributing the workload over the fog and the cloud domains according to the available resources. Simulation results showed that the proposed algorithm is more efficient when compared to other existing strategies. Huedo et al. [25] focused on processing latency-critical application in Edge Cloud computing and proposed a platform model of Edge computing. Workload redistribution in the fog stratum was studied in [14]. The proposed framework focused on balancing between communication load and computation latency, taking into account the task redundancy and mobility.

In [26], authors considered a hierarchical architecture of cloud and fog and proposed a task scheduling policy, wherein real-time tasks are to be processed in the fog layer, while computational-intensive tasks are to be executed by cloud servers. Their study covered both time and fog energy consumption. However, the results

showed only the fitness value of the proposed algorithm, which does not represent comprehensively the system performance.

In scalable computing systems, load balancing refers to efficient use of computational resources for task execution. D. Tychalas et al. [13] proposed a load balancing solution wherein all available computing resources have been utilized. Authors showed that their proposed scheduling algorithm can improve the resource utilization and reduce energy costs at the cloud center in low load in comparison to a weighted round robin policy. Recently, artificial intelligence (AI) has attracted a great attention in the research of the task scheduling and resource management problem as shown in [27, 28, 29].

In this study, we partly take the load based scheduling approach in [13] to achieve load balance. Moreover, we also investigate the impact of job characteristics in terms of job service demand and task size model on the system performance.

3 SYSTEM MODELING AND PROPOSAL

3.1 Cloud Fog Computing Overview

Stand-alone cloud centers have become outdated for extremely heterogeneous IoT applications, of which a portion requires high-speed communication and quick response time. Edge computing paradigm represents the use of IoT devices for data storage and computation to avoid data transfer and retrieve near real-time processing. Normally edge devices have limited storage and computation capacity. Fog computing was born to enhance cloud services nearby IoT devices. Fog computing is not a replacement but a complement of cloud by extending cloud services to the edge of the network. To give a comprehensive look on the hierarchical, distributed environment of Cloud-Fog, we can consider a multi-tiered Cloud-Fog architecture (shown in Figure 1), including the following layers:

Edge layer: In the edge, end-user smart devices connect to the (IoT) gateways which provide various services of computation, local storage, data routing, security, and so forth.

Fog layer: Lying in the middle of the architecture, Fog layer represents a bridge between user devices in edge networks and cloud center. In fog layer, data and resource management are decided by the fog broker. Due to the increase of IoT applications and data, a solution of locating a small-scale version of cloud data centers geographically nearby users (so-called Cloudlet) becomes feasible to improve job response.

Cloud layer: The cloud represents the most available storage and computing capacity to provide big data storage, real-time and batch processing for generated data from IoT devices.

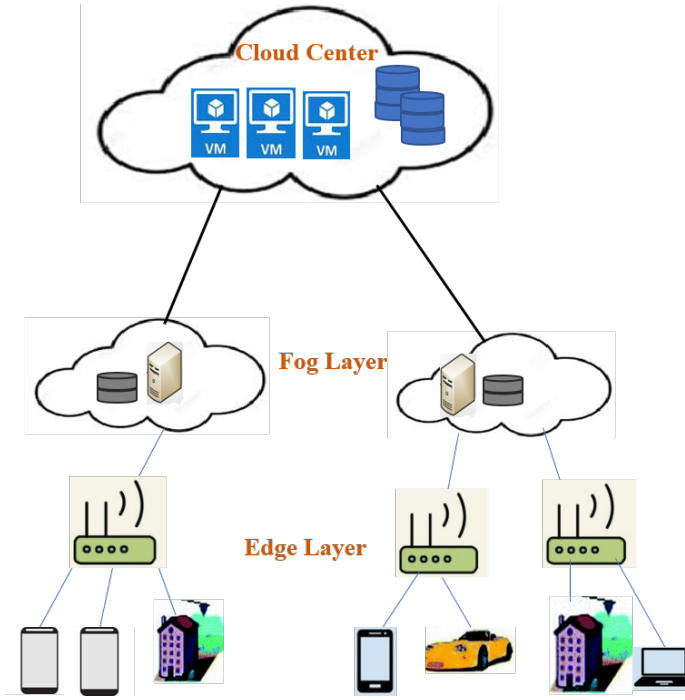


Figure 1. The Cloud-Fog architecture model

3.2 System Modeling

In the big picture of an IoT-enable Cloud computing environment, any device composed of processing capacity and storage in the network can be referred as a fog node. Therefore, there is a wide range of fog node types from end-user smart devices, low-performance gateways, powerful cloudlet servers, to virtual or physical machines at cloud center.

The considered computing system makes use of all available computational resources from four subsystems:

1. end-user smart devices,
2. low-performance network gateways using Raspberry Pi devices,
3. powerful cloudlet servers, and
4. VM pool at cloud center (as shown in Figure 2).

Let $M(i)$ $(i = 1, 2, 3, 4)$ be the number of nodes in subsystem i. We assume that nodes of subsystem i $(i = 1, 2, 3, 4)$ are homogeneous and have a service rate μ_i. Load Dispatcher is responsible to distribute incoming workloads to computational resources.

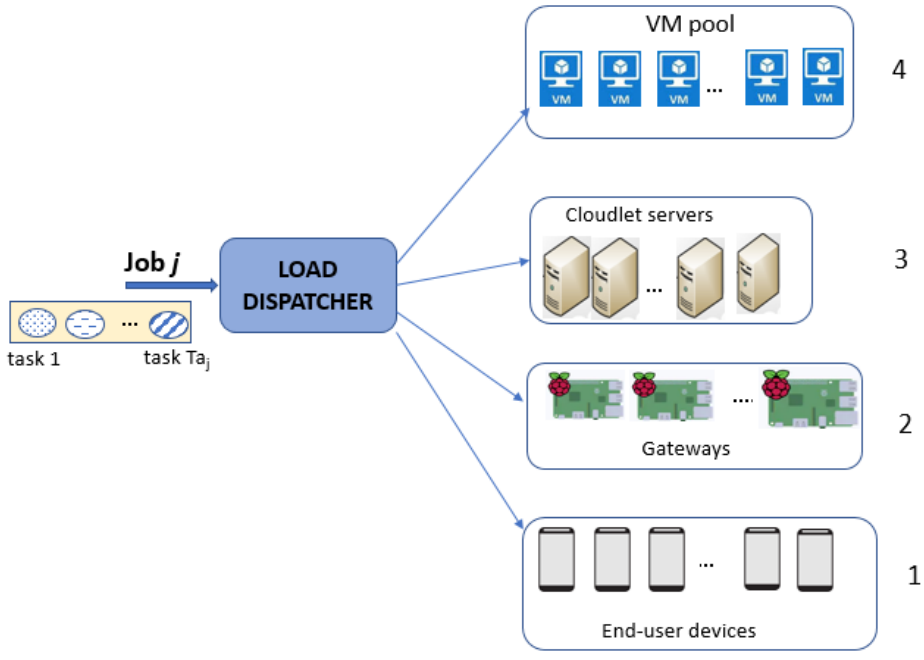


Figure 2. The considered Cloud-Fog computing system model

We consider Bag-of-Tasks (BoT) application model as the input workload. A BoT job consists of a set of independent tasks (i.e., tasks do not require communication with each other during their execution and can be executed in an arbitrary order) [30, 31]. This application model represents a such wide range of practices as data mining, heavily searches, computer imaging sweeping parameters, bioinformatics, and fractal calculations which occur in cloud computing environments. Thanks to their simplicity, BoT applications are appropriate to run over widely distributed, large-scale computing systems. As a result, efficient scheduling for the BoT applications has received a great attention of researchers [31, 13, 32, 33, 34]. We assume that incoming jobs have the following characteristics:

- a job task can be executed on any fog node;
- a job task is non-preemptible (i.e., task is uninterruptible while being processed);
- jobs are compute-intensive and have service time demand known by the scheduler;
- jobs are independent of each other.

3.3 Scheduling Problem and the Proposal of MOSA

Task scheduling problem in a single-machine system or homogeneous computing cluster simply refers to dispatching tasks in an appropriate order of execution. On the other hand, scheduling problem in a heterogeneous system composed of various computing machine types pays more attention to resource allocation for jobs/tasks. Task allocation refers to the selection of a computing machine to which a task is routed. Allocating an appropriate resource has a major impact on both user's satisfaction of services and operation cost paid by a service provider.

This study focuses on the resource allocation policy. If not stated otherwise, we assume that jobs arriving the system are to be served with First Come – First Served (FCFS) policy. Each task of a job is routed to node based on a resource allocation policy applied by the scheduler. A job task in the system will be served immediately if there is an available computing node. If all nodes are busy, a task will be routed to a node with the shortest queue and wait in the selected node's queue.

In [13], authors proposed a Load-Balance allocation policy that uses subsystem loads as thresholds for deciding resource selection. Their goal was to reduce the operation cost of cloud and Raspberry Pi cluster by keeping a low number of active servers in those subsystems so that idle servers can be switched off. Their proposal was compared with the best-effort Round-Robin policy. It is worth to note that switching off technique is inefficient to be applied when Round-Robin policy is used. The reason is Round-Robin selects computing servers with roughly equal probability, which causes the idle period of a server not significantly long enough to power it off.

In this study, we take the same approach of Load-Balance given in [13] to balance loads of all subsystems (cloud center, cloudlet, poor-resource devices, and edge devices). To enhance switching off technique for a system, the resource allocation policy attempts to reduce the number of servers needed for task processing (i.e., the more free servers are available and can be switched off, the lower the energy consumption of the system). Since switching on a sleeping server takes time, it may add more cost of execution delay and energy consumption (if the sleep period is shorter than switching on delay). To avoid the added cost, we only switch off free servers when the subsystem load meets the minimum threshold (θ_3) and switch on a server if the load reaches the maximum threshold (θ_4).

In addition, Cloud-Fog computing is a highly heterogeneous environment. Hence, user jobs' characteristics and resource heterogeneity should be taken into account to decide an appropriate task-resource mapping.

We propose a Multi-Objective scheduling algorithm (MOSA) taking into account job service time demands, resource processing capacity, and the loads of subsystems to improve quality of service as well as system operation cost with the following rules:

- End-user devices and poor-resource Raspberry Pis are preferred if their loads are less than the lower load threshold θ_1;

- Utilization of Cloudlet subsystem should be kept under the upper load threshold θ_2;
- Tasks with acute service time demand (that is, less than the service demand threshold $\beta(t)$) should be executed in resources closer to them (end-devices or Cloudlet);
- Cloud center is chosen if the above rules are not satisfied.

Let a job be identified by (j, Ta_j, sd_j), where j is job identification, Ta_j is the number of tasks, and sd_j is the service time demand of job j. To estimate the service demand of job, we use a threshold called statistical mean service demand in what follows. Let N(t) and $\beta(t)$ denote the number of historical incoming jobs and the average service time demand of those at the considered time t. The statistical mean service demand is calculated as:

$$\beta(t) = \frac{1}{N(t)} \sum_{j=1}^{N(t)} sd_j. \tag{1}$$

Algorithm 1 presents pseudo-code of the proposed resource allocation solution. Algorithm 2 describes the switching off policy for cloud and RaspberryPi subsystems.

4 RESULTS

4.1 Experimental Design

The evaluation is conducted using a simulation software written in C. We make a long-term run for each simulation that stops after five million completions. We also apply the statistical module [35] developed by Politecnico di Torino to collect and evaluate statistics during simulation. The results are obtained with the confidence level of 95 % and the accuracy (i.e., the ratio of the half-width of the confidence interval to the mean of collected observations) of 0.05.

The proposed MOSA is evaluated in comparson to other existing scheduling policies, Load-Balance (LB) and Weighted Round Robin (WRR) given in [13] and MPSO heuristic [36]. The system constructed for simulation runs is composed of 64 end-user devices, 64 powerful Cloudlet servers, 32 Raspberry Pis, and 128 virtual machines (VMs) located at cloud center.

We assume that the inter-arrive time and the execution time of jobs are exponentially distributed with means of $1/\lambda$ and $1/\mu$, respectively. An end-user device, a Raspberry Pi, a Cloudlet node, and a VM are assumed to have ability of executing tasks at service rate $\mu_1 = 2.0$ (tasks/s), $\mu_2 = 0.5$ (tasks/s), $\mu_3 = 1.0$ (tasks/s), $\mu_4 = 1.0$ (tasks/s), respectively.

We consider that the number of tasks of BoT jobs follows a uniform distribution in range of $[1, 8]$. Being frequently observed in practice, Power-of-two and Square job models [37] are also used as input workloads for evaluation. Workload models and their parameters are given in Table 1.

Algorithm 1 Pseudo-code of resource allocation in MOSA

for each new arriving job j **do**
  CALCULATE Load[End-Devices], Load[Cloudlet], Load[RasberryPi], Load[Cloud]
  if Load[End-Devices] $\leq \theta_1$ OR ($sd_j \leq \beta(t)$ AND Load[End-Devices] $\leq \theta_2$) **then**
    chosen_subsystem $\leftarrow$ End-Devices
  else if Load[Cloudlet] $\leq \theta_2$ **then**
    chosen_subsystem $\leftarrow$ Cloudlet
  else if Load[RaspberryPi] $\leq \theta_1$ OR ($sd_j \leq \beta(t)$ AND Load[RaspberryPi] $\leq \theta_2$) **then**
    chosen_subsystem $\leftarrow$ RaspberryPi
  else
    chosen_subsystem $\leftarrow$ Cloud
  end if
  GOTO ALLOCATION
end for
ALLOCATION: {Shortest queue based task scheduling}
for each task in task set Ta_j of job j **do**
  if found *free_server* in *chosen_subsystem* **then**
    ROUTE task to *free_server*
    GOTO ALGORITHM 2
  else
    Calculate server.queue in *chosen_subsystem*
    ROUTE task to the server with the shortest queue
  end if
end for

Algorithm 2 Pseudo-code of switching-off policy

if *chosen_subsystem* is Cloud or RaspberryPi **then**
  $s \leftarrow$ *chosen_subsystem*
  CALCULATE Load[s]
  if Load[s] $\leq \theta_3$ AND *number_active_servers*[s] > 1 **then**
    Switch off a free server in the subsystem
    Decrement *number_active_servers*[s]
  else if Load[s] $\geq \theta_4$ AND *number_active_servers*[s] is less than total number of servers in the system s **then**
    Switch on a sleep server in the subsystem s
    Increment *number_active_servers*[s]
  end if
end if

Workload Model	Description	Task Size	Average Task Size
Uniform	Task size (i.e., the number of tasks) is an integer that follows the uniform distribution within the range of $[1, 8]$	$[1, 2, \ldots, 8]$	4.5
Power-of-two	Task size is an integer that is calculated by 2^k, $k = 1, 2, 3$	$[2, 4, 8]$	≈ 4.67
Square	Task size is an integer that is calculated by k^2, $k = 1, 2, 3$	$[1, 3, 9]$	≈ 4.67

Table 1. Workload models and their parameters

The average service rate of the entire system is calculated as:

$$\mu = \left(\sum_{i=1}^{4} M(i) \times \mu_i \right) / T_{avg}. \tag{2}$$

Therefore the considered system has the average service rate of $(64 \times 2.0 + 64 \times 1.0 + 32 \times 0.5 + 128 \times 01.0)/4.5 = 336/4.5 \approx 74.67(jobs/s)$. We run the simulations with various arrival rates of 16.8 (jobs/s), 22.68 (jobs/s), 28.56 (jobs/s), 34.44 (jobs/s), and 40.32 (jobs/s).

We choose the load thresholds that $\theta_1 = 0.3$, $\theta_2 = 0.7$ for allocation decision and $\theta_3 = 0.5$, $\theta_4 = 0.7$ for switching off policy.

System performance metrics are as follows.

- Average response time of job (RT): The response time of a job is defined as the time period between job arrival and its departure. Let N be the total number of completed jobs during simulation time and rt_j be the response time of job j. The average response time is calculated as:

$$RT = \frac{1}{N} \sum_{j=1}^{N} rt_j. \tag{3}$$

- Average waiting time of task (WT): Let wt_t be the wait time in queue of task t before its execution and T_N be the total number of tasks of N completed jobs.

$$WT = \frac{1}{T_N} \sum_{t=1}^{T_N} wt_t. \tag{4}$$

- Average service time of tasks (ST): Let st_t be the service time of task t. The average service time of T_N tasks (the total number of tasks of N completed jobs)

is calculated as:

$$ST = \frac{1}{T_N} \sum_{t=1}^{T_N} st_t. \tag{5}$$

- Resource utilization ($U(i)$): defined as the average percentage of time that each server of subsystem i is in the busy state over the simulation time and calculated as

$$U(i) = \frac{(\sum_{m=1}^{M(i)} busy\_time_m)/M(i)}{simulation\_time} * 100\,\%. \tag{6}$$

- Average busy servers: the average number of servers processing tasks during the simulation time.

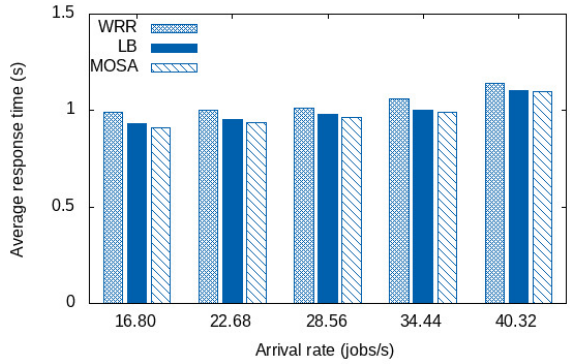
To estimate the effectiveness of switching off technique, metrics of number of busy servers and the resource utilization are used. The energy cost is directly proportional to the busy time of servers and the server power consumption. Therefore, utilization of a subsystem that addresses the percentage of time that a server is in busy state can be interpreted as the cost of subsystem. Moreover, the number busy servers can determine the energy cost of the total system. The system notations are listed in Table 2.

Notation	Description
$M(i)$	Number of servers in Subsystem i
μ_i	Service rate of a server in Subsystem i
μ	System service rate
λ	System arrival rate
sd_j	service demand of job j
Ta_j	Number of tasks of job j
T_{avg}	Average task number per job
$\beta(t)$	Service demand threshold at time t
θ_1	Lower Load threshold
θ_2	Upper Load threshold
θ_3	Minimum Load threshold for switching off
θ_4	Maximum Load threshold for swithcing on
RT	Average response time per job
WT	Average waiting time per task
ST	Average service time per task
$U(i)$	Average resource utilization of subsystem i

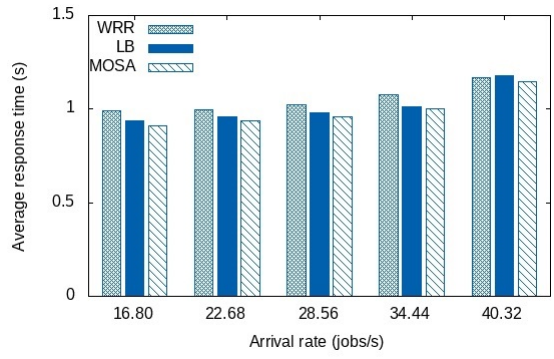
Table 2. System Notations

4.2 Numerical Results with Theoretical Loads

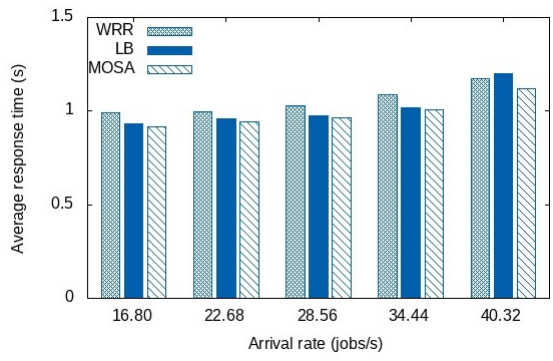
Figure 3 plots the average response time per job where three types of workload model are used. Figure 3 a) (with a uniform distribution workload model) shows that



a) Uniform task size model

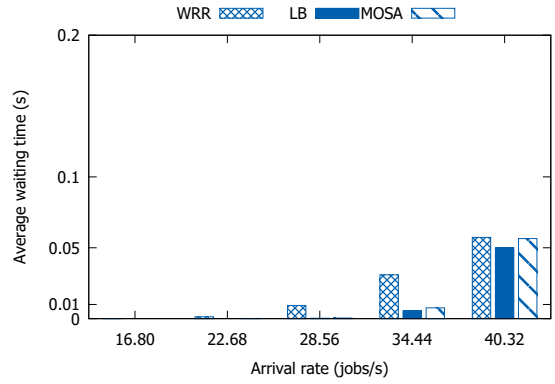


b) Power-of-two task size model

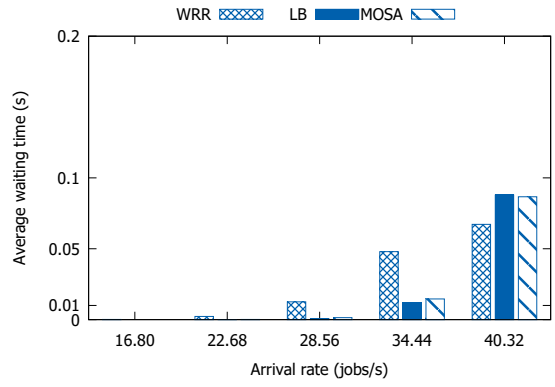


c) Square task size model

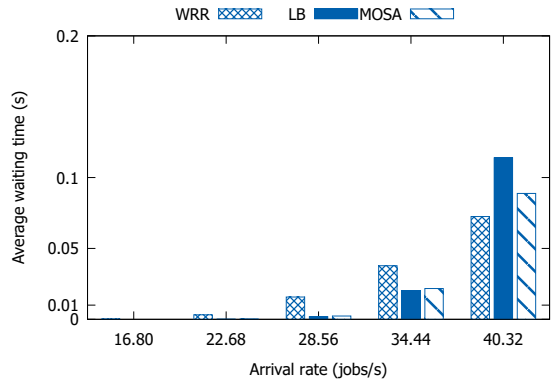
Figure 3. Average Response time (s) (a) Uniform model, b) Power-of-two model, c) Square model)



a) Uniform task size model



b) Power-of-two task size model



c) Square task size model

Figure 4. Average Waiting time of tasks (s) (a) Uniform model, b) Power-of-two model, c) Square model)

the proposed MOSA outperforms the other policies, regardless the used workload model. Particularly, MOSA improves the performance by 10 % and ≈ 3 % compared to Weighted Round Robin (WRR) and Load-Balance (LB) at low load, respectively. When the intensity is high, MOSA performs 6 % better than WRR and as well as the LB policy. Figures 3 b) and 3 c) (wherein Power-of-two and Square workload size models are used) point out that LB outperforms WRR at low load, but causes a slight increase in the response time when the load intensity is high. For the instance of Square workload size model, the average response time is increased by 3 % with the use of LB algorithm but decreased by approximately 6 % with MOSA, in compared to that achieved by WRR policy. In summary, the proposed MOSA attains better performance regardless the workload model and intensity.

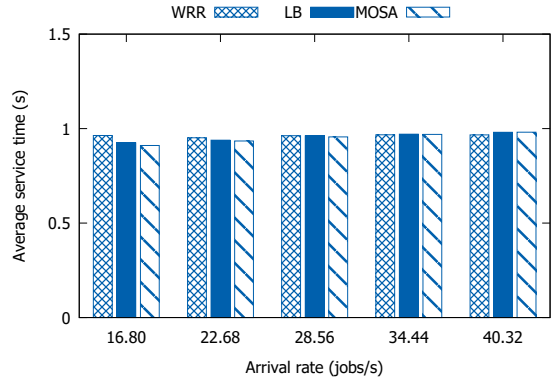
The average waiting time and average service time of tasks are plotted in Figures 4 and 5, respectively. It can be observed that at low to medium workload intensity, tasks have to wait for shorter time with LB and MOSA in compared to WRR. At high load, the waiting time with LB and MOSA is higher. It can be explained that tasks need to wait for servers to be switched on. Figure 5 shows that MOSA results in a slightly faster service at low load and there is no difference among algorithms at high load.

Figure 6 presents the resource utilization of subsystems where three scheduling policies are applied. We can observe that when the workload intensity increases, Weighted Round Robin (WRR) policy puts load stress on end-user devices (Figure 6 a)) while letting powerful servers in Cloudlet and Cloud center under-utilized (see Figures 6 b) and 6 d)). That can degrade the performance of end-user devices and cause a discomfort to users. Load-Balance (LB) policy leads to high utilization of low-performance Raspberry Pis when load is high (see Figure 6 c)). With the MOSA policy, loads of end-user devices and of Raspberries Pi are kept under the desired thresholds, as well as a balanced utilization between Cloudlet and Cloud center is achieved. It can be interpreted that the proposed MOSA results in better resource utilization compared to Weighted Round Robin (WRR) and Load-Balance policy.

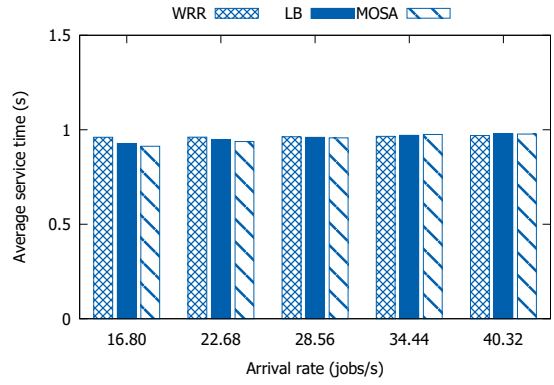
The resource utilization also depicts the energy cost of subsystems. In Figure 6 c), the portion of busy time of Raspberry Pi machines where MOSA is applied is slightly higher compared to LB at low load but less than LB when the intensity increases. Figure 6 d) shows that our proposed MOSA yields 2 % reduction in busy time of cloud servers at low load in comparison to LB. There is no such difference between these two algorithms at high load. It means that MOSA is able to maintain energy cost reduction as LB did. It is worth noting that WRR is not to be compared since switching off is not effective as aforementioned in Section 3.3.

Obtained results of subsystem utilization with the presence of Power-of-two and Square workload model are given in Figures 7 and 8. We observe stable outcomes and system behavior regardless the types of workloads.

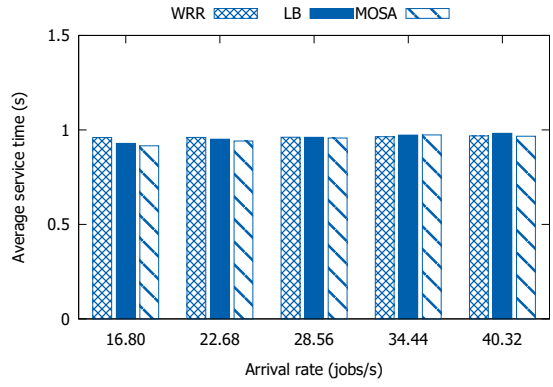
Figure 9 depicts the average number of busy servers over the run time while three job models are applied as input workload alternatively. It shows that the proposed MOSA and LB use less servers for task processing than WRR at low



a) Uniform task size model



b) Power-of-two task size model



c) Square task size model

Figure 5. Average Service time of tasks (s) (a) Uniform model, b) Power-of-two model, c) Square model)

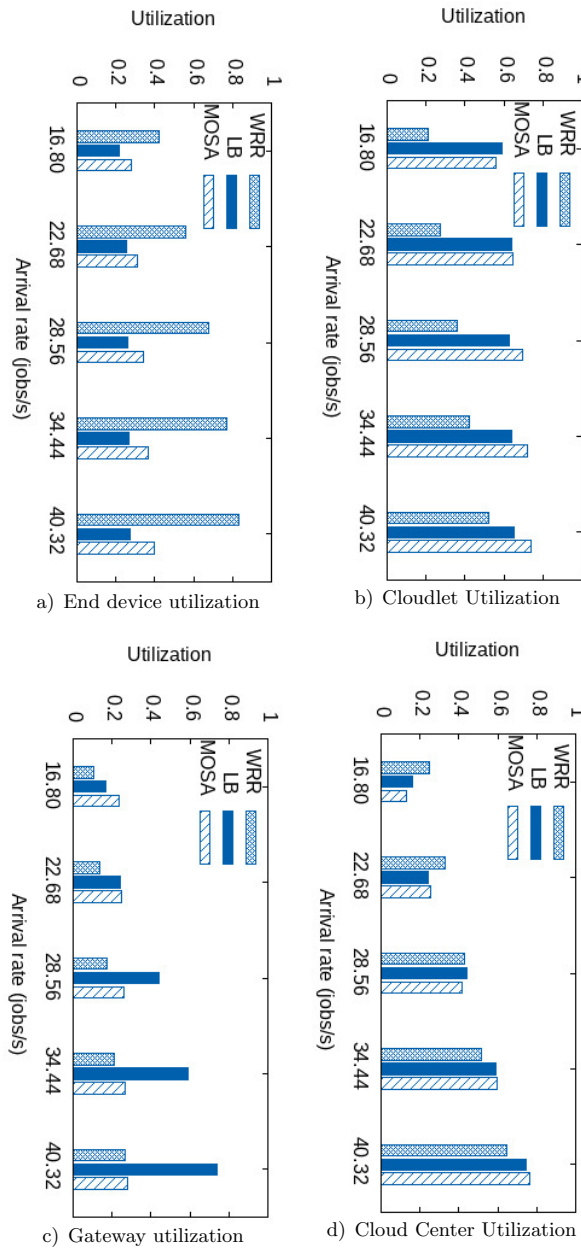


Figure 6. Utilization of subsystems with Uniform workload model (a) End-device, b) Cloudlet, c) Raspberries, d) Cloud center)

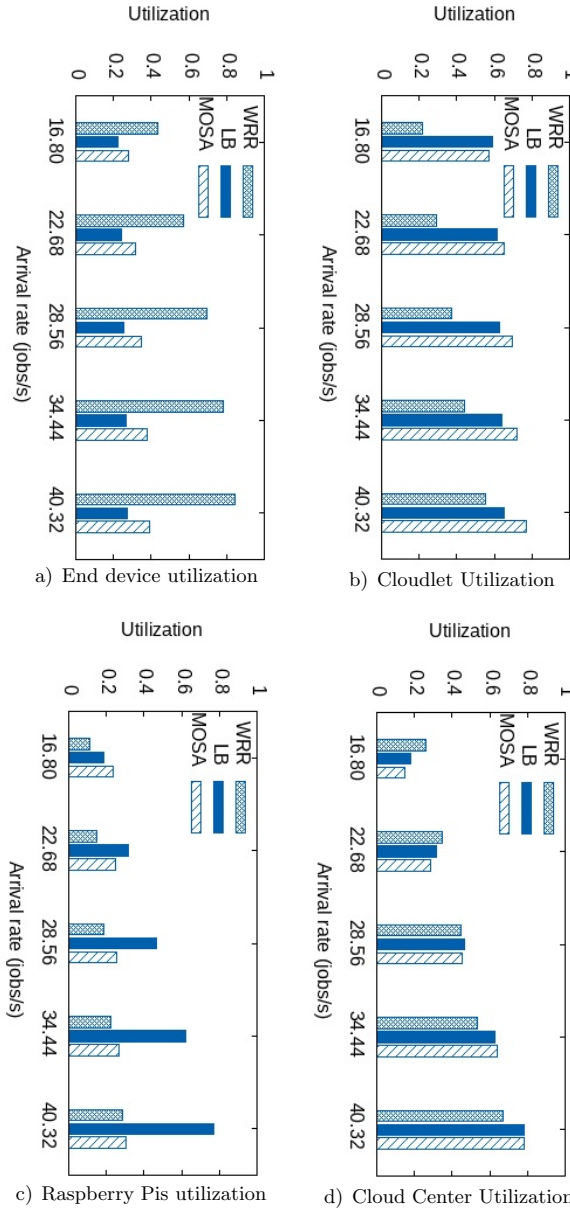


Figure 7. Utilization of subsystems with Power-of-two workload model (a) End-device, b) Cloudlet, c) Raspberries, d) Cloud center)

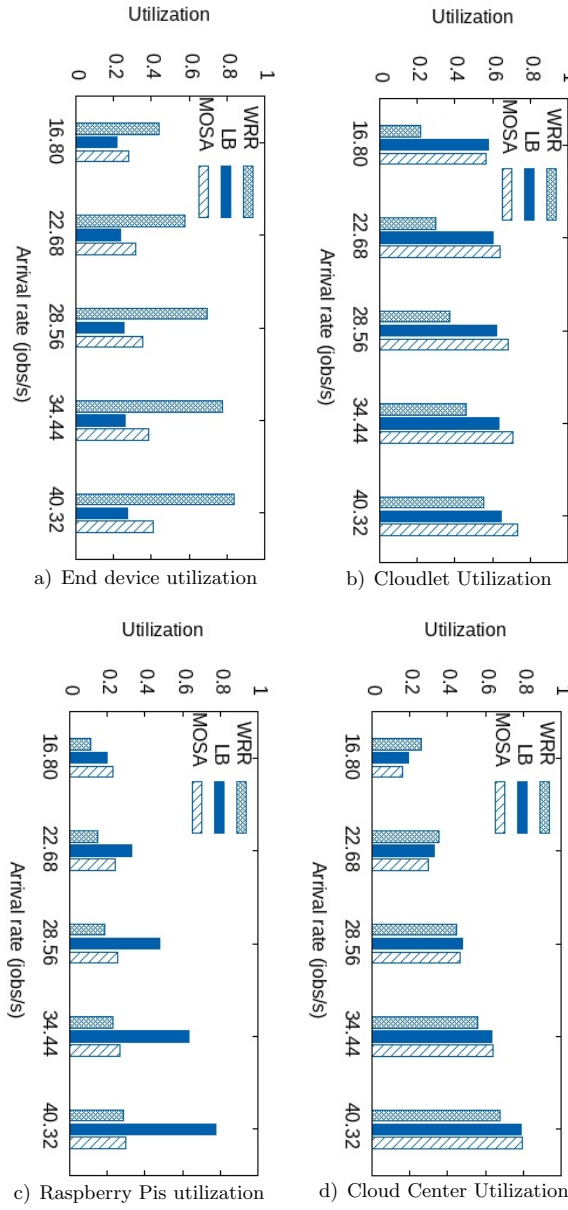
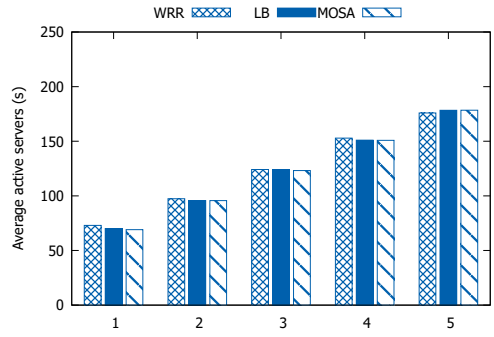
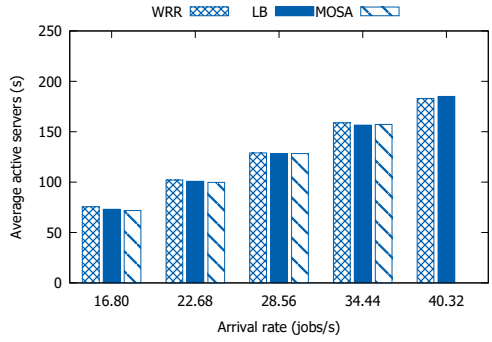


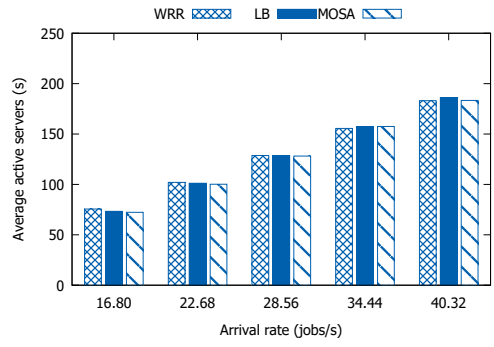
Figure 8. Utilization of subsystems with Square workload model (a) End-device, b) Cloudlet, c) Raspberries, d) Cloud center)



a) Uniform task size model



b) Power-of-two task size model



c) Square task size model

Figure 9. Average active servers (s) (a) Uniform model, b) Power-of-two model, c) Square model)

load and there is no difference observed at high workload intensity. That means the overall energy cost of the Cloud-Fog system can be reduced with MOSA and LB at low load. Similar behaviors are obtained with different workload models.

4.3 Comparison Between the Proposed MOSA and MPSO

This section presents a comparison between the proposed algorithm and the MPSO-based scheduling heuristic. The MPSO (Modified Particle Swarm Optimization) was studied in [36] with the aim to balance the job makespan and total operation cost of the system. The principle of MPSO based heuristic in the comparison can be drawn as follows:

1. Calculate the fitness values of particles as a function of execution time and the utilization of nodes;
2. Find the personal best position for a task in each subsystem according to fitness values;
3. Update the global best position of which the lowest fitness value is obtained.

Figures 10, 11 and 12 plot the job average response time, task service time, and task waiting time, respectively. It can be observed in Figure 10 that the MOSA policy outperforms the others (WRR, LB and MPSO) at low load. At high workload intensity, there is only an obscure difference in the performance with LB, MPSO and MOSA whilst WRR performs worst. Figure 11 indicates that MPSO provides the fastest execution service among the solutions. However, MPSO also causes the longest waiting time of tasks for the service as shown in Figure 12. Therefore, we observe a performance degradation with MPSO, showed in Figure 10.

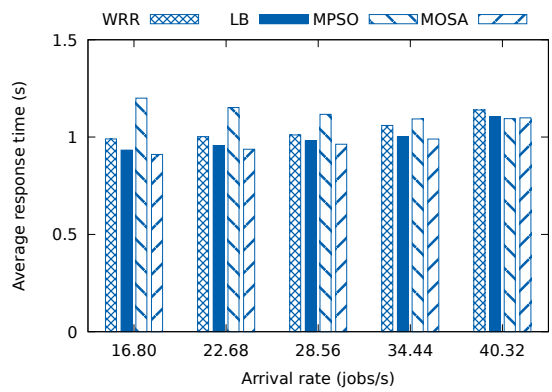


Figure 10. Average response time per job (uniform model)

The average active servers versus load intensity with the application of different algorithms are illustrated in Figure 13. It shows that MPSO leads to lowest number

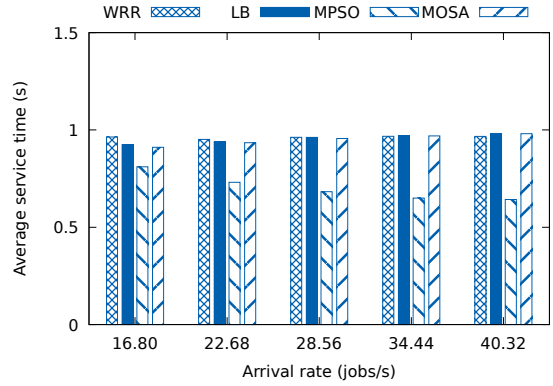


Figure 11. Average service time per job (uniform model)

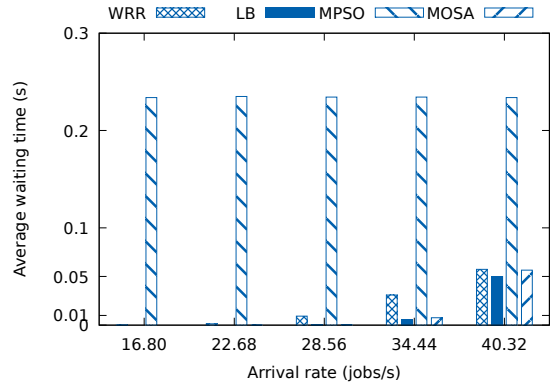


Figure 12. Average waiting time per job (uniform model)

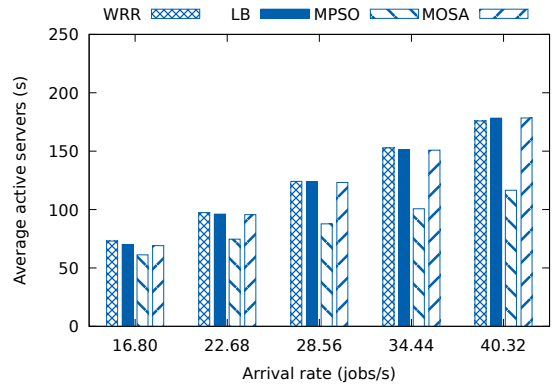
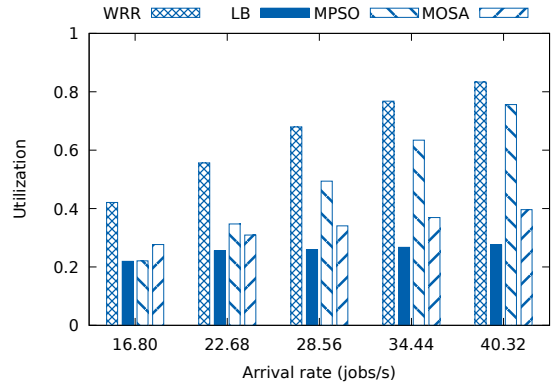


Figure 13. Average active servers (uniform model)

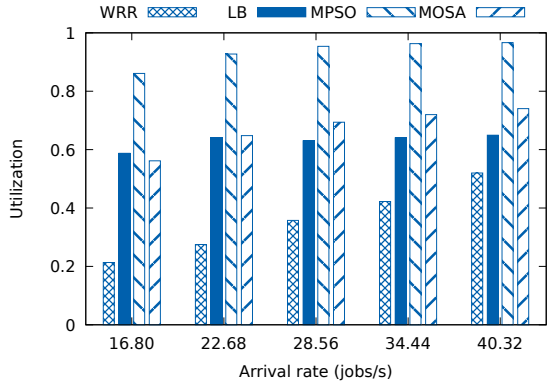
of busy server among solutions, which may result in lower operation energy cost of the system.

Figure 14 depicts the utilization ratio of the four subsystems. It is shown that MPSO causes an intense amount of workloads processed in end devices and cloudlet severs which are closer to users. On the other hand, cloud servers and network gateways are mostly underutilized with the MPSO algorithm. This phenomenon can be explained by the powerful capacity of local processing. A heavy execution load on end devices may cause service discomfort of users, while other resources are not utilized properly.

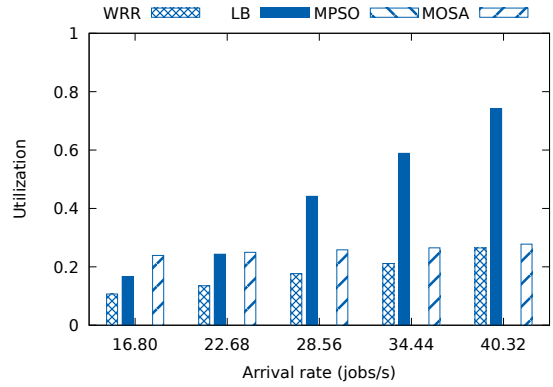
In summary, MPSO yields a balance for execution time and total operation cost from operator's perspective. However, the overall service quality defined by the job response time and a balanced resource utilization which consider user's experience is achieved by the proposed solution MOSA.



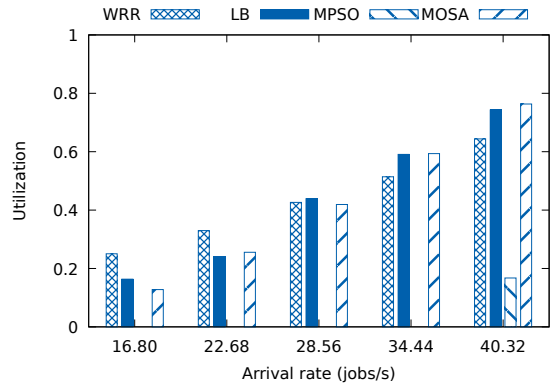
a) End device utilization



b) Cloudlet Utilization



c) Raspberry Pis utilization



d) Cloud Center Utilization

Figure 14. Utilization of subsystems with Square workload model (a) End-device, b) Cloudlet, c) Raspberry Pi, d) Cloud center)

4.4 Experimental Results with Traced Workloads

In order to examine the proposed algorithm thoroughly, we also experiment the simulation with various real workloads which were captured from practical executions. Table 3 presents the statistical metrics of traced workloads in terms of mean and variance. It is observed that these do not follow exponential distribution.

Figure 15 depicts the average response time of job when three traced workloads are used for comparing the proposed MOSA with three other scheduling algorithms. It shows that with INCC workload trace the proposed MOSA outperforms WRR and LB with a reduction of over 15 % in the average response time, while resulting in roughly the same performance to MPSO. With other workload traces, the

Workload	Number of Samples	Inter-Arrival Time		Service Time	
		Mean	Var	Mean	Var
INCC trace	1 037 093	15.236	102 062.096	546.591	1 789 574.190
NVS trace	2 188 683	62.861	671 414.041	404.909	446 969.583
UPR trace	1 352 714	129.436	473 524.089	850.539	1 509 552.918

Table 3. Traced workloads

proposed MOSA outperforms other solutions with a reduction of 30 %–40 % in average response time. There is no clear difference observed between WRR, LB and MPSO.

Figure 16 shows that the utilization of subsystems with the use of INCC workload. The outputs verify that the proposal yields better resource utilization by maintaining the resource load under desired thresholds. It can also be observed that WRR and LB policies send a larger percentage of workload to remote cloud center and low-performance gateways (Raspberry Pi). That means the idle period of servers in those subsystems become shorter which lead to the higher energy cost in comparison to our solution. On the other hand, the MPSO overloads end user devices that causes dissatisfaction to users.

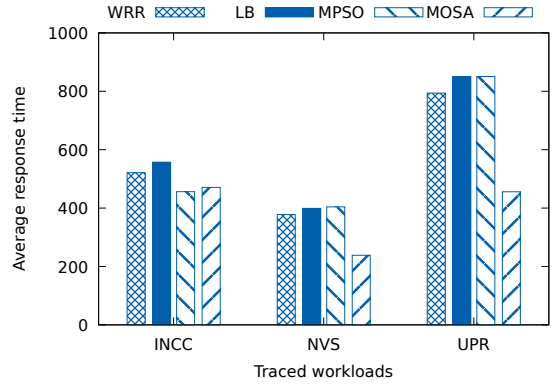


Figure 15. Average response time vs. traced workloads

5 CONCLUSIONS

Fog computing has become significant to develop 5G/6G network and allow more IoT applications to connect to the system. As a result, a merged computing environment for IoT applications is spread from edge node to remote cloud center. In this study, we investigate the impact of characteristics of jobs and resources on the system performance of a merged computing environment of Edge, Fog and Cloud. With the aim at improving system performance and energy efficiency of this

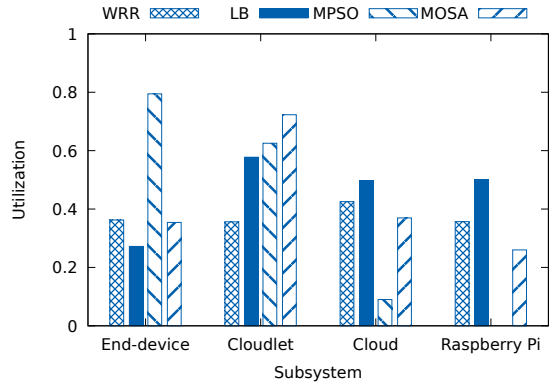


Figure 16. Utilization of subsystems (INCC workload)

such heterogeneous environment, we propose a multi-objective scheduling algorithm-MOSA using various characteristics of user workloads and computational resources. The proposed algorithm MOSA uses job service demand and resource processing capacity to find an appropriate task-resource mapping. Numerical results show that MOSA leads to shorter average response time per job than existing WRR, LB, and MPSO policies.

For load balancing and energy efficiency, the proposed MOSA uses load threshold based approach. Obtained results indicate that MOSA yields a slight improvement in load balancing than LB in some scenarios and similar outcomes in other cases. Compared with MPSO that balances execution time and total system cost, MOSA achieves similar or better performance while yielding better resource utilization and guaranteeing user's service experience. Experimental outputs with the use of real workloads validate the foregoing conclusions wherein MOSA yields a reduction of 15 %–40 % in the average response time per job and balances resource utilization among subsystems.

The proposed MOSA is based on job service demand as well as hard thresholds of resource utilization to make decision, while different service level agreements (SLAs) also depend on other proprieties of jobs and computing resources. Therefore, we must investigate more complex scenarios wherein other characteristics such as types of jobs, physical location of job request should be taken into account to achieve various SLA requirements in future work.

REFERENCES

[1] Bonomi, F.—Milito, R.—Zhu, J.—Addepalli, S.: Fog Computing and Its Role in the Internet of Things. Proceedings of the First Edition of the MCC Workshop

on Mobile Cloud Computing, Association for Computing Machinery, New York, NY, USA, MCC '12, 2012, pp. 13–16, doi: 10.1145/2342509.2342513.

[2] Sarkar, S.—Misra, S.: Theoretical Modelling of Fog Computing: A Green Computing Paradigm to Support IoT Applications. IET Networks, Vol. 5, 2016, No. 2, pp. 23–29, doi: 10.1049/iet-net.2015.0034.

[3] Atlam, H. F.—Walters, R. J.—Wills, G. B.: Fog Computing and the Internet of Things: A Review. Big Data and Cognitive Computing, Vol. 2, 2018, No. 2, doi: 10.3390/bdcc2020010.

[4] Yi, S.—Li, C.—Li, Q.: A Survey of Fog Computing: Concepts, Applications and Issues. Proceedings of the 2015 Workshop on Mobile Big Data, Association for Computing Machinery, New York, NY, USA, Mobidata '15, 2015, pp. 37–42, doi: 10.1145/2757384.2757397.

[5] Dastjerdi, A.—Gupta, H.—Calheiros, R.—Ghosh, S.—Buyya, R.: Chapter 4 – Fog Computing: Principles, Architectures, And applications. In: Buyya, R., Vahid Dastjerdi, A. (Eds.): Internet of Things. Morgan Kaufmann, 2016, pp. 61–75, doi: 10.1016/B978-0-12-805395-9.00004-6.

[6] Das, R.—Inuwa, M. M.: A Review on Fog Computing: Issues, Characteristics, Challenges, and Potential Applications. Telematics and Informatics Reports, Vol. 10, 2023, Art. No. 100049, doi: 10.1016/j.teler.2023.100049.

[7] Margariti, S. V.—Dimakopoulos, V. V.—Tsoumanis, G.: Modeling and Simulation Tools for Fog Computing – A Comprehensive Survey from a Cost Perspective. Future Internet, Vol. 12, 2020, No. 5, doi: 10.3390/fi12050089.

[8] Shakarami, A.—Ghobaei-Arani, M.—Masdari, M.—Hosseinzadeh, M.: A Survey on the Computation Offloading Approaches in Mobile Edge/Cloud Computing Environment: A Stochastic-Based Perspective. Journal of Grid Computing, Vol. 18, 2020, pp. 639–671, doi: 10.1007/s10723-020-09530-2.

[9] Liu, L.—Qi, D.—Zhou, N.—Wu, Y.—Lin, F.: A Task Scheduling Algorithm Based on Classification Mining in Fog Computing Environment. Wirel. Commun. Mob. Comput., Vol. 2018, 2018, doi: 10.1155/2018/2102348.

[10] Deng, R.—Lu, R.—Lai, C.—Luan, T. H.—Liang, H.: Optimal Workload Allocation in Fog-Cloud Computing Toward Balanced Delay and Power Consumption. IEEE Internet of Things Journal, Vol. 3, 2016, No. 6, pp. 1171–1181, doi: 10.1109/JIOT.2016.2565516.

[11] Nguyen, B. M.—Thi Thanh Binh, H.—The Anh, T.—Bao Son, D.: Evolutionary Algorithms to Optimize Task Scheduling Problem for the IoT Based Bag-of-Tasks Application in Cloud-Fog Computing Environment. Applied Sciences, Vol. 9, 2019, No. 9, doi: 10.3390/app9091730.

[12] Aladwani, T.: Scheduling IoT Healthcare Tasks in Fog Computing Based on Their Importance. Procedia Computer Science, Vol. 163, 2019, pp. 560–569, doi: 10.1016/j.procs.2019.12.138 (16^{th} Learning and Technology Conference 2019 Artificial Intelligence and Machine Learning: Embedding the Intelligence).

[13] Tychalas, D.—Karatza, H.: A Scheduling Algorithm for a Fog Computing System with Bag-of-Tasks Jobs: Simulation and Performance Evaluation. Simulation Modelling Practice and Theory, Vol. 98, 2020, Art. No. 101982, doi:

10.1016/j.simpat.2019.101982.

[14] Li, S.—Maddah-Ali, M. A.—Avestimehr, A. S.: Coding for Distributed Fog Computing. IEEE Communications Magazine, Vol. 55, 2017, No. 4, pp. 34–40, doi: 10.1109/MCOM.2017.1600894.

[15] Rafique, H.—Shah, M. A.—Islam, S. U.—Maqsood, T.—Khan, S.—Maple, C.: A Novel Bio-Inspired Hybrid Algorithm (NBIHA) for Efficient Resource Management in Fog Computing. IEEE Access, Vol. 7, 2019, pp. 115760–115773, doi: 10.1109/ACCESS.2019.2924958.

[16] Nan, Y.—Li, W.—Bao, W.—Delicato, F. C.—Pires, P. F.—Dou, Y.—Zomaya, A. Y.: Adaptive Energy-Aware Computation Offloading for Cloud of Things Systems. IEEE Access, Vol. 5, 2017, pp. 23947–23957, doi: 10.1109/ACCESS.2017.2766165.

[17] Kabirzadeh, S.—Rahbari, D.—Nickray, M.: A Hyper Heuristic Algorithm for Scheduling of Fog Networks. 2017 21st Conference of Open Innovations Association (FRUCT), 2017, pp. 148–155, doi: 10.23919/FRUCT.2017.8250177.

[18] Katal, A.—Dahiya, S.—Choudhury, T.: Energy Efficiency in Cloud Computing Data Centers: A Survey on Software Technologies. Cluster Computing, 2022, doi: 10.1007/s10586-022-03713-0.

[19] Wadhwa, H.—Aron, R.: TRAM: Technique for Resource Allocation and Management in Fog Computing Environment. Journal of Supercomputing, Vol. 78, 2022, pp. 667–690, doi: 10.1007/s11227-021-03885-3.

[20] Qiu, Y.—Jiang, C.—Wang, Y.—Ou, D.—Li, Y.—Wan, J.: Energy Aware Virtual Machine Scheduling in Data Centers. Energies, Vol. 12, 2019, No. 4, doi: 10.3390/en12040646.

[21] Xiang, H.—Peng, M.—Sun, Y.—Yan, S.: Mode Selection and Resource Allocation in Sliced Fog Radio Access Networks: A Reinforcement Learning Approach. IEEE Transactions on Vehicular Technology, Vol. 69, 2020, No. 4, pp. 4271–4284, doi: 10.1109/TVT.2020.2972999.

[22] Oueis, J.—Strinati, E. C.—Sardellitti, S.—Barbarossa, S.: Small Cell Clustering for Efficient Distributed Fog Computing: A Multi-User Case. 2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall), 2015, pp. 1–5, doi: 10.1109/VTCFall.2015.7391144.

[23] Abbasi M., Mohammadi P. E., K. M.: Workload Allocation in IoT-Fog-Cloud Architecture Using a Multi-Objective Genetic Algorithm. Journal of Grid Computing, Vol. 18, 2020, pp. 43–56, doi: 10.1007/s10723-020-09507-1.

[24] S. Agarwal, S. Yadav, A. K. Y.: An Efficient Architecture and Algorithm for Resource Provisioning in Fog Computing. International Journal of Information Engineering and Electronic Business (IJIEEB), Vol. 8, 2016, No. 1, pp. 48–61, doi: 10.5815/ijieeb.2016.01.06.

[25] Huedo, E.—Montero, R.—Moreno-Vozmediano, R.: Opportunistic Deployment of Distributed Edge Clouds for Latency-Critical Applications. Journal of Grid Computing, Vol. 19, 2021, No. 2, doi: 10.1007/s10723-021-09545-3.

[26] Movahedi, Z.—Defude, B.—Hosseininia, A. m.: An Efficient Population-Based Multi-Objective Task Scheduling Approach in Fog Computing Systems. J. Cloud

Comput., Vol. 10, 2021, No. 1, doi: 10.1186/s13677-021-00264-4.

[27] Hassanat, A. B. A.: Furthest-Pair-Based Decision Trees: Experimental Results on Big Data Classification. Information, Vol. 9, 2018, No. 11, doi: 10.3390/info9110284.

[28] Xu, F.—Yang, F.—Zhao, C.—Wu, S.: Deep Reinforcement Learning Based Joint Edge Resource Management in Maritime Network. China Communications, Vol. 17, 2020, No. 5, pp. 211–222, doi: 10.23919/JCC.2020.05.016.

[29] Zhang, W.—Yang, D.—Peng, H.—Wu, W.—Quan, W.—Zhang, H.—Shen, X.: Deep Reinforcement Learning Based Resource Management for DNN Inference in Industrial IoT. IEEE Transactions on Vehicular Technology, Vol. 70, 2021, No. 8, pp. 7605–7618, doi: 10.1109/TVT.2021.3068255.

[30] Tran, N. M.—Wolters, L.: Towards a Profound Analysis of Bags-of-Tasks in Parallel Systems and Their Performance Impact. Association for Computing Machinery, New York, NY, USA, 2011, doi: 10.1145/1996130.1996148.

[31] Iosup, A.—Sonmez, O.—Anoep, S.—Epema, D.: The Performance of Bags-of-Tasks in Large-Scale Distributed Systems. Association for Computing Machinery, New York, NY, USA, 2008, doi: 10.1145/1383422.1383435.

[32] Moschakis, I. A.—Karatza, H. D.: A Meta-Heuristic Optimization Approach to the Scheduling of Bag-of-Tasks Applications on Heterogeneous Clouds with Multi-Level Arrivals and Critical Jobs. Simulation Modelling Practice and Theory, Vol. 57, 2015, pp. 1–25, doi: 10.1016/j.simpat.2015.04.009.

[33] Oprescu, A. M.—Kielmann, T.: Bag-of-Tasks Scheduling Under Budget Constraints. Proceedings of the 2010 IEEE Second International Conference on Cloud Computing Technology and Science, IEEE Computer Society, USA, CLOUDCOM '10, 2010, pp. 351–359, doi: 10.1109/CloudCom.2010.32.

[34] Weng, C.—Lu, X.: Heuristic Scheduling for Bag-of-Tasks Applications in Combination with QoS in the Computational Grid. Future Generation Computer Systems, Vol. 21, 2005, No. 2, pp. 271–280, doi: 10.1016/j.future.2003.10.004 (Advanced Grid Technologies).

[35] di Torino, P.: A Statistical Module. `https://www.telematica.polito.it/oldsite/class/statistics.ps.gz` [accessed 2020-10-15].

[36] Izakian, H.—Tork Ladani, B.—Zamanifar, K.—Abraham, A.: A Novel Particle Swarm Optimization Approach for Grid Job Scheduling. In: Prasad, S. K., Routray, S., Khurana, R., Sahni, S. (Eds.): Information Systems, Technology and Management. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 100–109.

[37] Aida, K.: Effect of Job Size Characteristics on Job Scheduling Performance. In: Feitelson, D. G., Rudolph, L. (Eds.): Job Scheduling Strategies for Parallel Processing. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000, pp. 1–17, doi: 10.1007/3-540-39997-6_1.

Xuan Thi Tran works as Researcher and Lecturer at the University of Information and Communication Technology, Thai Nguyen University, Vietnam. She earned her M.Sc. and Ph.D. degree at the Budapest University of Technology and Economics, Hungary in 2014 and 2020, respectively. Her research focuses on performance evaluation, distributed computing, job scheduling, and big data processing. Her research interests are cloud computing, 5G/6G mobile network, and machine learning.

Computing and Informatics, Vol. 42, 2023, 340–363, doi: 10.31577/cai_2023_2_340

EVOLUTION-BY-COEVOLUTION OF NEURAL NETWORKS FOR AUDIO CLASSIFICATION

Włodzimierz Funika, Paweł Koperek, Tomasz Wiewióra

Institute of Computer Science
Faculty of Computer Science Electronics and Telecommunication
AGH University of Krakow, al. Mickiewicza 30, 30-059 Kraków, Poland
e-mail: funika@agh.edu.pl, {pkoperek, tomasz.wiewiora95}@gmail.com

Abstract. Neural networks are increasingly used in recognition problems, including static and moving images, sounds, etc. Unfortunately, the selection of optimal neural network architecture for a specific recognition problem is a difficult task, which often has an experimental nature. In this paper we present the use of evolutionary algorithms to obtain optimal architectures of neural networks used for audio sample classification. We extend the Pytorch DNN Evolution tool implementing co-evolutionary algorithms which create groups of neural networks that solve a given problem with a certain accuracy, with the support for problems in which training data consists of audio samples. In this paper we use the co-evolutionary approach to solve a sample sound classification problem. We describe how the sound data was prepared for processing with the use of the Mel Frequency Cepstral Coefficients (MFCC). Next we present the results of experiments conducted with the AudioMnist dataset. The obtained neural network architectures, whose classification accuracy is comparable to the classification accuracy attained by the AlexNet neural network, and their implications are discussed.

Keywords: Neural networks, evolutionary algorithms, sound recognition

Mathematics Subject Classification 2010: 68-T05, 68-T10

1 INTRODUCTION

Speech is the basic medium of interpersonal communication. However, it is more and more often used as a communication channel between a human and a ma-

chine thanks to recent developments in the area of speech recognition. Nowadays, we observe a very rapid development of many tools based on speech recognition technology. There are multiple examples of such systems, e.g. virtual assistants ([1]: Google Assistant, Apple Siri, Amazon Alexa, Microsoft Cortana), car control systems [2], robot control systems [3]. The number of applications and systems which employ a speech-based interface is constantly growing. Speech recognition and audio classification are already used in search engines, car navigation and translators.

Recently, using the neural networks became a very popular approach to creating audio classification systems. One of the main challenges in this context is the time consuming process of designing the neural network architecture. It requires a lot of domain knowledge and a large number of experiments. Incorrect decisions may lead to suboptimal classification performance and render the newly created systems incapable of serving its basic purpose. In order to automate and streamline the model discovery process an evolutionary algorithm may be used. In this paper we demonstrate how the Pytorch DNN Evolution framework [4] can be used to accelerate the process of creating neural network architectures which solve the audio classification problem. The network architectures are obtained in subsequent iterations of the genetic algorithm, which over time solves the given classification problem. To validate our approach we present the results of a series of experiments conducted with the AudioMNIST dataset [5] used as a sample input dataset.

The neural network model is only one of the components required. Audio preprocessing is another crucial element of building a system which communicates with human users successfully. However, the sound signal analysis is also applicable to many other fields, e.g. medicine, bio-acoustics or seismology. In medicine, mainly in otorhinolaryngology, a spectrogram could be used in a voice examination. It separates the sound signal into bands with different frequencies. Such a result is used by a phoniatrist to detect a subtle early changes in the voice. These changes may be the initial stage in the development of vocal chords nodules [6]. Another field in which sound signal analysis is used is bio-acoustics – it studies the impact and role of sound in the lives of animals. In this field, tools are mainly used to extract individual sound characteristics, which can be used to distinguish between species of animals or even their specific individuals. An example of the use of sound recognition techniques in bio-acoustics is a bat echolocation research [7]. Sound analysis is also often used in seismology. There are methods for extracting features from sound samples. Systems for the classification of micro-seismic signals are being developed in order to minimize risks in the mining industry [8]. These systems aim to early detect events which might cause dangerous vibrations in mines. Since audio pre-processing can be used in such a variety of contexts, there are numerous available techniques. This means that audio pre-processing requires careful examination and adjusting to a specific problem. In the system presented in the current paper, the audio signal is transformed to the MFCC coefficients [9]. In our research we have attempted to empirically determine an optimal number of coefficients which need to be used in the context of the analyzed dataset.

The paper is structured as follows. In Section 2 we discuss related research and the background of our work. In Section 3 we describe the modifications to the Pytorch DNN Evolution framework. In Section 4 we present a description of the conducted experiments. In Section 5 the results of the experiments are discussed. Finally, in Section 6 we conclude our research based on the conducted experiments.

2 RELATED WORK

In this section the background for our research is presented.

2.1 Using Neural Networks in Sound Classification

Neural networks are a very flexible method for approximating very complex functions. This makes them very useful in many domains, including speech recognition. An example of the use of neural networks in those areas can be the problem of classifying the sound recordings of numbers in English [10]. The authors presented a method of sound samples classification based on spectrograms. The architecture used in the experiment was based on the architecture of AlexNet [11]. The architecture contained five convolutional layers. Two types of experiments were conducted on the AlexNet network. The first one was a classification of the recordings of digits, so there were ten possible classes. The second one was classification of the recordings according to the gender of the person who has been recorded.

The problem of sound classification can also be solved with architectures, which previously worked well with the problems of image classification [12]. The authors adapted the existing network architecture *VGG19* [13]. The *VGG19* architecture is most often used to classify images. In the case of image classification tasks, it is common only to retrain the fully connected layers. However, in the case of audio classification, the authors decided to retrain the last convolutional block along with the fully connected layers.

The above approach is based on manual adaptation of the network architecture to a new type of problem. However, this does not guarantee the creation of an optimal architecture that will solve the classification problem embedded in another domain. Subsequent changes introduced in such a network architecture and their subsequent verification are time-consuming. Therefore, we would like to propose an approach to automating this process with the use of the coevolutionary algorithm outlined below.

2.2 Convolutional Neural Networks

There are many types of neural networks which are used in the sound recognition problems. Convolutional Neural Network *(CNN)* [14] is one of the most widely used ones. *CNN* typically consists of four types of layers [15]: convolutional layer, pooling layer, fully-connected layer (dense layer), and a softmax layer.

All neurons in the convolution layer take as input a cross-section of the output from the previous layer. Each neuron multiplies the local input data by the weight matrix. The weight matrix or the local filter is replicated over the entire input space to detect a specific type of pattern. All neurons share the same weights to create a feature map of objects. The entire convolutional layer consists of many feature maps of objects that have been generated using differently placed filters. This procedure is used to isolate many types of local patterns that may occur in any location. For speech recognition, the input space may be a two-dimensional plane where the dimensions of the data are frequencies and time [16]. Following the convolution layer typically a pooling layer is used. Such a layer similarly as the convolutional layer takes input from the local region of the previous convolutional layer to generate a single output from that region. The common operator of these layers used in *CNN* is *max-pooling*. It outputs the maximum value in each sub-region. This operation reduces the computational complexity and makes the network resistant to slight changes in the position of local patterns. The next layer after at least one convolutional and one pooling block is a fully connected layer, also called *dense layer*. The main task of this layer is the final classification of the object. This layer identifies the input object and assigns it to the specific class.

In literature one can also find other types of neural network types, e.g. recurrent [17], LSTM [18] or Transformer [19]. They have various applications, however in this paper we focus just on those which are relevant to our research.

2.3 Evolving of Deep Neural Network Architectures

The use of Deep Learning provided state-of-the-art results in many domains like image recognition [11, 13] or machine translation [20, 21]. Unfortunately, it involves to carefully design the neural network architecture, which is often a very complicated task. It relies heavily on the experience and knowledge of the researcher, what makes it difficult for beginners to modify or create new models fitting their particular use-case. A variety of Neural Architecture Search (NAS) algorithms were developed [22, 23] to tackle this issue. Among the employed methods there are examples of the Reinforcement Learning [24, 25], gradient optimization [26] or evolutionary algorithms (EA) [27, 28]. In the current research we focus on the last category, due to the high flexibility of the algorithms from that category.

Evolutionary algorithms share one common weakness: in their basic form they require large amounts of resources in order to evaluate the architectures they create. In order to mitigate this problem different strategies are being employed [22]:

- *Reducing the search space* [29, 27]. In the basic approach, the search space of architectures has the representation of all necessary components of an architecture (e.g. layers, their sizes, connections between layers etc.). This means that the algorithm needs to take many smaller steps in order to arrive at the optimal solution. To reduce the number of such steps, less granular concepts are

introduced as search space building blocks, e.g. *cells* or *blocks* (which consist of multiple neural layers themselves).

- *Reusing neural architecture* [28]. Instead of creating every model from scratch, the search procedure uses the existing artificially designed architectures as a starting point, then it and transforms them to improve the performance.
- *Incomplete training* [30]. The individual architecture evaluation is speeded up by changing the mechanism which ranks the architectures between each other. Instead of conducting a lengthy training of a full dataset, e.g. early stopping can be used to limit the amount of computations required to obtain the result of a comparison. Another variant of this approach is to share some or all weights with an already trained model.

The co-evolutionary approach employed in the current paper [31] uses subsets of the original training dataset to reduce the amount of time required to evaluate an individual. Low level architecture building blocks (neural network layers) are used to define the search space. All individual neural network models are created and trained from scratch. This allows to categorize the method as using *incomplete training* technique while not employing *neural architecture reuse* nor *reducing the search space*.

2.4 Coevolution of Neural Networks and Fitness Predictors

Coevolutionary algorithms are a type of evolutionary algorithms in which the training process involves two or more individuals species. In the coevolution algorithm, the assessment of the quality of a given individual in a population depends on individuals from a different population. Coevolutionary algorithms can be divided into two main types: *competitive* and *cooperative* ones. In the competitive approach, the assessment of an individual is obtained through competing with the individuals of the other population. On the other hand, the cooperative algorithms allow individuals from one population to enhance the fitness of individuals of the second one. This means they promote cooperation of individuals with each other. During the training process, this results in rewarding the individuals for solving problems together and punishing for independence.

The evaluation of the individuals in the context of the evolutionary algorithms can be one of two types: objective or subjective fitness. The first one is aimed at defining a function that is used to evaluate the assessment of a given individual which does not require taking into account other individuals (e.g. an error rate in classification of images). In the second approach the assessment depends also on other individuals, including individuals of a different species and thus can be classified as using subjective fitness. To evaluate neural networks (first species), training set subsets (second species) are being used. Such an approach can be used to avoid using the full, very often large, training dataset to evaluate the neural networks what helps to significantly reduce the time required for such an evaluation.

An example of implementation of this idea is the Pytorch DNN Evolution [31] project which attempts to discover an optimal architecture of a neural network used for the purpose of image classification. It conducts the process of co-evolution of two populations in which it recombines and mutates individuals to obtain the most fit individual. The first population consists of neural networks architectures, which are being trained to classify images. The second one is a population of subsets of the original training examples dataset, which can be used for quick assessment of the neural network model. In other words it is a population of so called *fitness predictors*. Each population has its own definition of *fitness*, which is a measure of *quality* of an individual and is used to select which individuals are going to survive to the next iteration of evolution. It is worth nothing that hyperparameters of the neural network training process are provided as the input to the process and are not modified by the evolution.

It is important to note that the *fitness* definition in the context of evolution of neural networks depends on the problem which is being solved. For a sample task of predicting the next element in a time-series it might be, e.g., the accuracy of predictions made by the neural network. In Pytorch DNN Evolution each population uses its own fitness definition. When trained with the use of dataset S, the neural network's n fitness $f_{NN}(n, S)$ can be defined as the accuracy of image recognition on the given set examples. This can be formalized as Formula (1).

$$f_{NN}(n, S) = 100 \cdot \frac{\sum_{i=1}^{|S|} \text{IsCorrect}(\text{Recognize}(n, x_i), \text{Target}(x_i)}{S}, \tag{1}$$

where:

- S – the training dataset, it can be e.g. the full dataset or a fitness predictor,
- $|S|$ – is the size of the dataset S,
- x_i – i^{th} element of dataset S,
- $\text{Recognize}(n, x_i)$ – is the class recognized by the neural network under evaluation,
- $\text{Target}(x_i)$ – is the expected class of the sample as specified in the dataset,
- $\text{IsCorrect}(x, y)$ – is a function which can be defined as follows:

$$\text{IsCorrect}(x, y) = \begin{cases} 1, & \text{if } x = y, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

As the evolution progresses, we expect the fitness to increase, which translates to an improvement to the recognition accuracy. In the second population of fitness predictors, the objective of the evolutionary process is to find a subset of the training dataset which allows to compare the fitness of two neural networks. One way to achieve such a goal is to identify the samples of the training dataset, which render similar results to training over the complete dataset. Such samples might have features which are e.g. very common across the dataset or might be very difficult to

accurately recognize. The fitness of a fitness predictor p (the $f_{FP}(p)$) is therefore also defined with the use of the recognition accuracy, however in this case it is not maximized (Formula (3)).

$$f_{FP}(p) = 100 * |f_{NN}(T, p) - f_{NN}(T, \text{FullDataset})|, \tag{3}$$

where:

- FullDataset – the full training dataset,
- p – fitness predictor under evaluation, subset of FullDataset,
- f_{NN} – the neural network fitness as defined in Formula (1),
- T – the neural network used as a trainer for the fitness predictor population.

The co-evolution algorithm which implements these ideas is expressed more formally in the form of a pseudocode, presented in Listing 39.

The result of the coevolution algorithm is a set of neural network architectures. In the paper [31] the described algorithm is applied to the image recognition problem based on the *MNIST* [32] dataset.

2.5 Sound Representation

Sound can be represented in many ways [33]. Depending on the needs, various representations of the sound samples allow to emphasize a specific aspect of the data that is the most interesting in a given context. The most basic method to represent audio data is the waveform which describes changes in sound amplitude over time. Such a representation is very easy to interpret by humans. Another method used to represent audio data is the spectrogram. It shows the distribution of the amplitude spectrum of the sound signal at a given time. Thus, it informs us about the distribution of the intensity of the sound components depending on the frequency of these components. One of the most popular representations is the Mel Frequency Cepstral Coefficients (*MFCC*) [9]. It is based on the *mel* scale. This scale determines the subjective perception of the sound level by human due to the frequency measurement scale measured in hertz (Hz). The units of this scale are called *mel*s. *MFCC* is often used to prepare sound data as input for neural networks. It was used in the preprocessing described in [34]. Each recording has been split into audio chunks and transformed into the *MFCC* representation.

3 EVOLUTIONARY SYSTEM MODIFICATIONS

The main functionality of the Pytorch DNN Evolution framework [4] is the automatic discovery of neural network architectures for solving supervised learning problems. It has been designed to support only datasets consisting of images, e.g. the *MNIST* [32] and *CIFAR10* [35] collections, which are examples of the image classification problems. However, the architecture discovery method implemented

```
def EvaluateIndividual(dnn, fp):
  phenotype = TranslateGenotypeToDNN(dnn)
  dataset_sample = ExtractSamplesFromTrainingDataset(fp)
  phenotype.train(dataset_dample)
  return phenotype.evaluate_test_dataset()

def EvolutionIteration(parents, trainer):
  children = []
  parents_size = len(parents)
  for i in range(parents_size):
    swap(parents, i, random(parents_size))
  for i in range(0, parents_size, 2):
    crossed_over = CrossingOver(parents[i], parents[i+1])
    mutated = [Mutate(crossed_over[0]), Mutate(crossed_over[1])]
    children.extend(mutated)

  for i in len(children):
    children[i].fitness = EvaluateIndividual(children[i], trainer)

  new_population = TournamentPopulations(parents, children)
    best_individual = SelectBestFitnessIndividual(new_population)
    return new_population, best_individual

def CoEvolution(N_fp, N_dnn, N_epochs):
    population_fp = InitializeRandomFPPopulation(N_fp)
    population_dnn = InitializeRandomDNNPopulation(N_dnn)

    best_fp = RandomInt(N_fp)
    best_dnn = RandomInt(N_dnn)

    for i in range(N_epochs):
      for j in range(N_dnn):
        population_dnn, best_dnn = EvolutionIteration(
          population_dnn, best_fp)
        best_dnn.fitness = EvaluateIndividual(
          best_dnn, FullTrainingDataset)
        population_fp, best_fp = EvolutionIteration(
          population_fp, best_dnn)
```

Listing 1. The pseudocode of the co-evolution algorithm implemented by *pytorch-dnn-evolution* package

in the discussed framework, presented in Figure 1, is not constrained to that class of problems. It can be applied to other domains, which can be expressed as supervised learning problems, i.e. it is possible to create a dataset for which sample network outputs can be assigned. In the current paper we present an attempt to use the the Pytorch DNN Evolution for sound recognition domain. In the sections that follow we demonstrate its effectiveness by applying it to a sample dataset.

The Pytorch DNN Evolution is designed to work in a distributed environment and consists of two major components: the *evolution driver* and *workers*. The first component is responsible for execution of the evolutionary algorithm. It maintains two populations: generates the genotype of the individuals constituting the initial population, crosses-over and mutates them according to set probability parameters, triggers evaluation of individual's fitness when necessary. The responsibility of the worker is to perform evaluation of an individual, what can be translated to the

following steps: translating the genotype to a trainable neural network, preparing the input dataset for training, conducting the training and finally measuring and reporting the fitness value, e.g. by testing the classification accuracy with the use of a separate test dataset. It is worth noting that to perform all of its tasks, the evolution driver is not required to translate the individual's genotype to another form. Since the genotype is represented as an array of numbers, applying crossing-over and mutation is a straightforward operation. Thanks to that, the evolution driver component can be applied to a wide range of problems and does not need to be changed, e.g., to introduce support for other types of neural network layers or a new domain. On the contrary, it is just the worker component which needs to be altered. Such an architecture allows the researcher to focus on the details of a specific domain and allows to simply reuse the core co-evolutionary algorithm without changes.

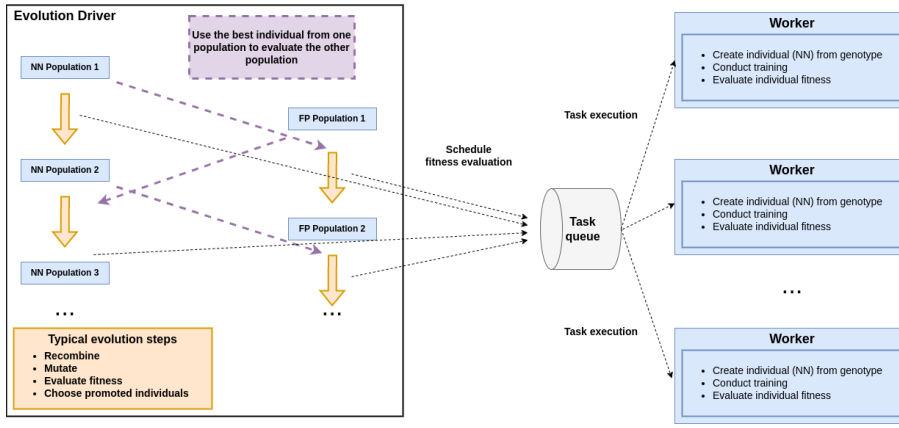


Figure 1. The architecture of the pytorch-dnn-evolution

In our work, to allow the application of the co-evolutionary approach to the domain of sound classification, we have extended the worker component. The worker is one of the crucial parts of the evolutionary system, as it conducts the evaluation of the individual genotypes. The modifications included:

- Interpretation of the individual genotype has been extended with support for other neural network layer types e.g. a convolutional layer. The network architecture is is not limited to creating simple, fully connected layers anymore. Supporting the new layer types includes also dynamically introducing the additional components which transform the format of samples between layers.
- Adjusting the training logic. Introducing the support for different layer types required also changing how the training and evaluation of networks is conducted.
- Introducing support for new types of datasets. This involves extending the preprocessing procedures to ensure that the data samples are presented to the neural

network as Mel Frequency Cepstral Coefficients and the datasets can be used to conduct co-evolution. Operations on audio data such as reading data from a file and retrieving *MFCC* were implemented using the *torchaudio library* [36]. In order to allow evaluation of the discussed approach, we have chosen to integrate the AudioMnist dataset [5], which has been already widely used in that research area [37, 38].

The modified version of the framework is presented in Figure 2.

During the first attempts to train neural networks on a set of audio data, we found a considerable time overhead was introduced by the pre-processing of sound data (processing the relevant part of the dataset into MFCCs). This would have a significant impact on the overall operating time of the genetic algorithm, since each neural network training process requires pre-processing of data before it can commence. Therefore, we have implemented the caching of the MFCCs and labels associated with these data. In this approach, the worker converts the entire audio data set to MFCCs only once before the first training. Coefficients are stored in the file with the relevant labels. During the training of the subsequent neural networks, the worker uses the data saved in this file. This optimization allowed to significantly reduce the experiment time.

4 DATASET PREPARATION

To conduct our experiments we have used the AudioMnist dataset. This collection contains 30 000 recordings of reading numbers from 0 to 9 in English. The recordings were prepared by 60 various speakers. Each recording is additionally enhanced with metadata about the speaker, such as: accent, age, gender. 48 men and 12 women participated in the recordings. The dataset could be used as a model benchmark for various audio data classification tasks. The MNIST or the CIFAR10 datasets perform a similar function for the classification of images.

The analysis of Mel Cepstral Coefficients was used for the data preprocessing for neural networks. The *MFCC* is based on the mel scale that reflects the subjective perception of sound, which is often used in the audio analysis. Generating MFCCs requires choosing an appropriate number of coefficients which are taken into account when analyzing a sound sample. The result of *MFCC* analysis is a three-dimensional representation of the recording. The dimensions of this data type are time, the number of Mel Cepstral Coefficient and its value [9]. Figure 3 presents the results of such an analysis for 10 sound samples of English words from *zero* to *nine*. The horizontal axis represents time, the left vertical axis – the number of coefficients, the color denotes the MFCC value.

As depicted in Figure 3, the values of Mel Cepstral Coefficients oscillate closer and closer to zero as the frequency increases. This means that they are less and less useful for the neural network training. Unfortunately, at the same time, processing them requires using a model with more parameters, what leads to an increase in the training time. To optimize the training time, only a subset including between 12

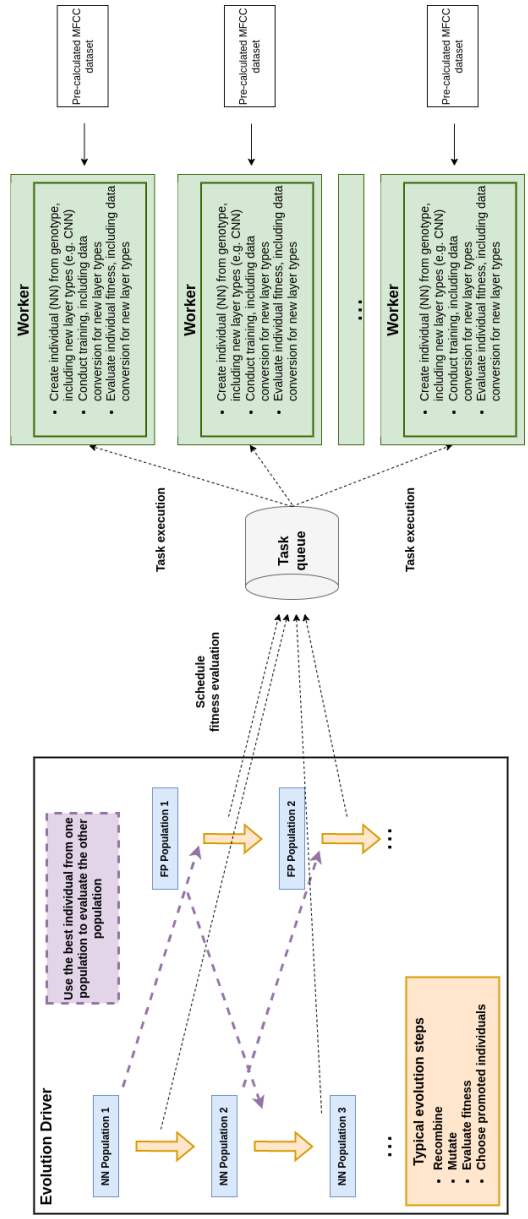


Figure 2. The extended architecture of the pytorch-dnn-evolution which includes changes described in Section 3

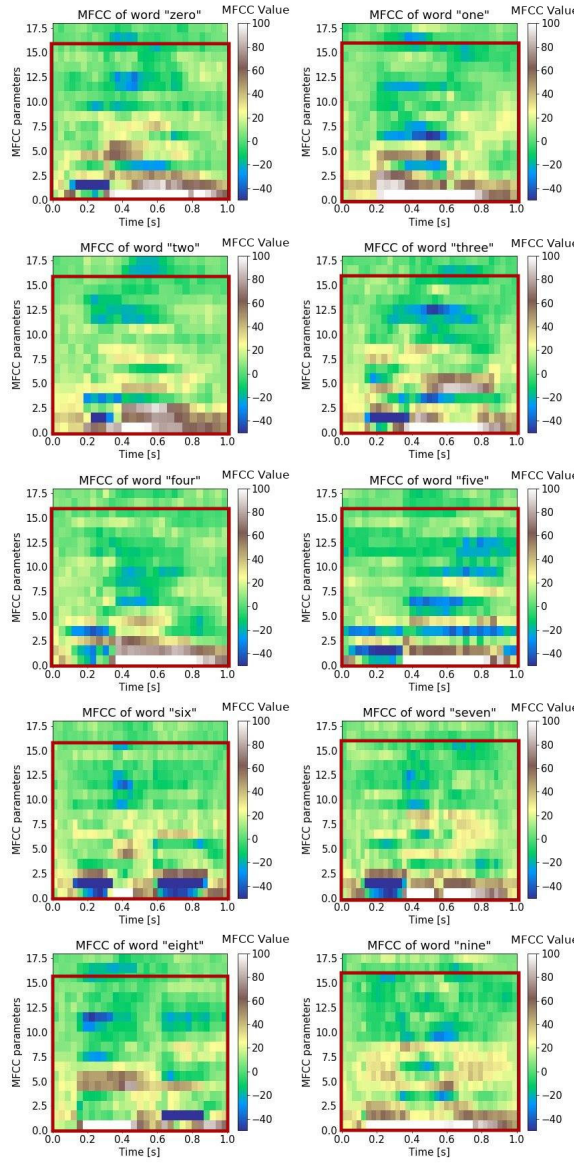


Figure 3. Visualization of the first 18 *MFCC* mel coefficients of the audio sample of words from 'zero' to 'nine'. The coefficients that were taken into account for neural network training were marked with a red rectangle.

to 18 first *MFCC* coefficients is taken into account for the purpose of training. The optimal number of coefficients to be taken into account has been determined empirically by running the neural network training experiments. In those experiments the neural network consisted of three convolutional layers. The dataset was divided into a training set (25 000 samples) and a test set (5 000 samples). The charts in Figures 4 and 5 present the results: while Figure 4 shows the relationship between the number of Mel Cepstral Coefficients and the classification accuracy, Figure 5 shows the relationship between the number of mel cepstral coefficients and the training time of the neural network.

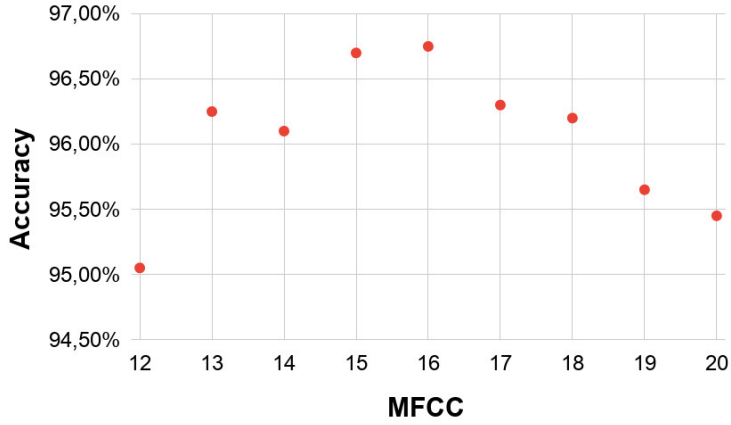


Figure 4. Classification accuracy for the first 12 to 20 *MFCC* coefficients

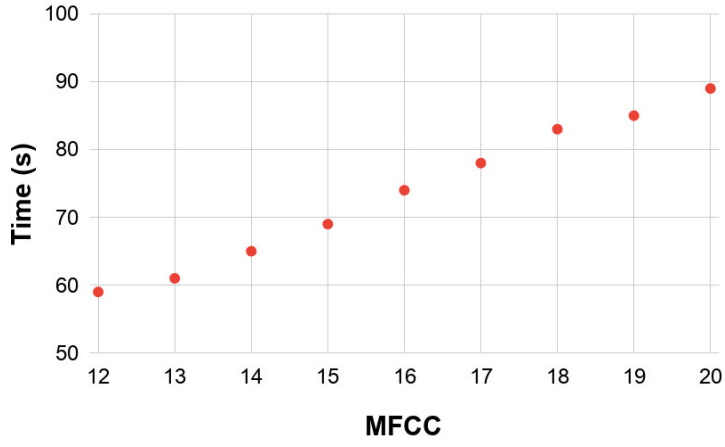


Figure 5. Training time of the neural network for the first 12 to 20 *MFCC* coefficients

The conducted experiments show that for the AudioMnist dataset, the optimal number of *MFCC* coefficients which should be taken into account is 16. Increasing the number of the coefficients has a negative effect on the training time of the neural network. Using more coefficients requires operating larger matrices and therefore requires performing more computations. We also noticed that the classification accuracy decreases when using more than 16 coefficients, what indicated that using more data would not lead to improvements in the context of classification.

5 EXPERIMENTS RESULTS

Two experiments were conducted using the modified Pytorch DNN Evolution framework. In the first one, we tried to estimate the appropriate individual size from the population of the training datasets. This experiment was aimed at evaluation of the minimum size of the training dataset for which the coevolution algorithm would still work correctly. The goal of the second experiment was to validate the correctness of the coevolutionary process and generate an architecture, which would render results comparable to one designed manually by a human researcher. The coevolution was applied to the creation of neural networks which attempted to classify sound samples.

All the test runs were performed using the same configuration. The only exception was the size of population of training datasets used in the Experiment 1, which was required due to the nature of that experiment. In all the runs 100 iterations of coevolution were performed. The parameters of the genetic algorithm are presented in Table 1.

Parameter	**Population of Neural Network**	**Population of Training Dataset**
Crossover probability	0.75	0.75
Mutation probability	0.1	0.1
Individual size	8	1 000
Population size	8	4

Table 1. Parameters of the genetic algorithm used during the coevolution

In the case of the population of neural networks, the size of the individual corresponds to the size of the generated networks. However, in the case of the population of training datasets, the size of the individual is the size of the training dataset.

5.1 Experiment 1

The first experiment using the Pytorch DNN Evolution tool was conducted to find the optimal size of the training dataset used during the evolutionary process. Such

a dataset on the one hand should be as small as possible to allow for fast individual evaluation (neural network training) and on the other hand it should contain enough samples which would allow the evolution to make progress. To find the optimal size, we conducted subsequent subexperiments in which the size of the set of training data was gradually being reduced. For each training dataset size we have conducted five runs. For a given dataset size we have recorded the maximum accuracy achieved by a neural network and the average accuracy of the best neural networks obtained through evolution but trained on a full training dataset.

Table 2 presents the accuracy for different sizes of the training dataset.

Size of the Training Dataset	**The Maximum Accuracy of Neural Networks Trained on the Subset**	**The Average Accuracy of Neural Networks Trained on the Full Dataset**
5 000	94.70 %	97.28 %
4 000	93.28 %	96.98 %
3 000	**93** %	**97.05** %
2 000	92 %	96.66 %
1 000	90 %	94.14 %
800	81 %	93.26 %
600	79 %	92.30 %
400	79 %	92.18 %
200	77 %	92.72 %
100	67 %	90.72 %

Table 2. Accuracy of training for AudioMnist subsets of different size

Based on those results, we drawn the following conclusions:

- While reducing the size of the training dataset we were obtaining lower classification accuracy in the neural networks population.
- The neural networks trained on several hundred recordings would not achieve very good accuracy in the classification. It should be noted that despite that, the effect of coevolution is still noticeable. We could observe a growing trend in the accuracy of neural network's classification for only 800 samples. The progress made by evolution in this case is presented in Figure 6.
- We have observed gains in the accuracy of the neural network when using training set sizes above 1 000 samples.
- The optimal size of the training dataset is around 3 000 samples. The accuracy of classification of the whole dataset of networks obtained through co-evolution did not grow significantly (for 4 000 it dropped to 96.98 %, for 5 000 it grew to 97.28 %) when we increased the size of training dataset further. However, the training was becoming considerably slower (Figure 7), (about 7 seconds for 4 000

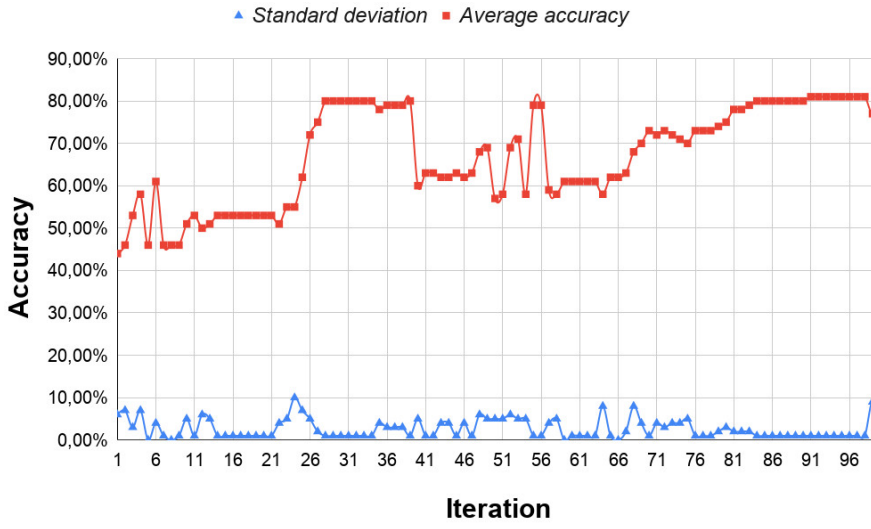


Figure 6. Accuracy of classification achieved by the population of neural networks. The size of the subset is 800. Maximum achieved accuracy is 81 %.

and about 13 seconds for 5 000) per each network training. We have decided to use the subset size of 3 000 samples in the experiments that followed.

5.2 Experiment 2

In the second experiment the goal of the coevolution process was to find a neural network architecture capable of solving the problem of digit classification. The size of the training subset was set to 3 000 recordings, as per result of Experiment 1. The graphs given in Figures 8 and 9 show the progress of the coevolution algorithm:

- Figure 8 shows the accuracy of the neural networks in the classification of digit recordings over successive iterations of the coevolutionary algorithm. The increasingly higher fitness values obtained in the subsequent iterations prove that the individuals cope better and better with speech recognition. This confirms that the evolution is able to make progress in the expected direction.
- Figure 9 shows the average accuracy of the neural network trained on individuals from the population of training datasets (fitness predictors). We can observe that on the contrary to the neural networks improving their accuracy over the course of evolution, the average accuracy of the training over the population of fitness predictors is decreasing. This suggests that in order to approximate the results of the training with a full dataset, the evolution chooses the samples, for which the classification accuracy is lowest, in other words they are hard to classify by the neural networks.

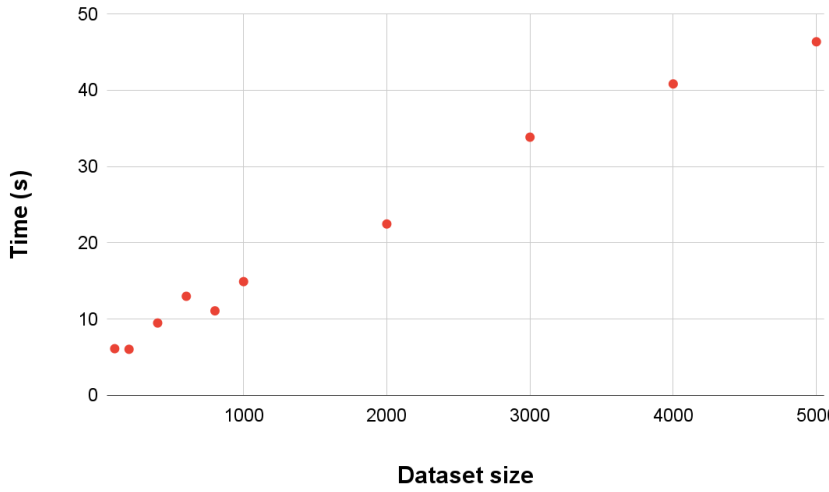


Figure 7. Training time of the neural networks for different dataset sizes

Over the course of the evolution, the maximum accuracy that was achieved was equal to 93 %. However, it should be noted that the network model in the Pytorch DNN Evolution framework was trained only on a certain subset of training data selected (the fitness predictor). Under those conditions, the classification accuracy of the neural network model is expected to be lower than in the case of training the same network model on the entire dataset. Therefore, the neural network model was trained again, however by using the entire training dataset. This allowed achieving the classification accuracy of 97.05 %.

This result can be compared, e.g. with the AlexNet model (designed by a human researcher) used in [10] which is a convolutional neural network as well. That model has also been trained to classify the AudioMnist with the use of the stochastic gradient descent and reached $95.82\,\% \pm 1.49\,\%$. In this context the result obtained by the automatically generated model could be considered satisfactory. Figure 10 presents the neural network architecture generated by Pytorch DNN Evolution. It consists of four subsequent CNN layers followed by a fully connected layer, which produces the final result (a vector of probabilities that the input sample belongs to each of the ten classes). Such a structure resembles AlexNet in which convolutional layers are also followed by the fully connected ones.

6 CONCLUSIONS

In this paper we have demonstrated how the Pytorch DNN Evolution tool can be used to automatically create a neural network architecture for the classification of

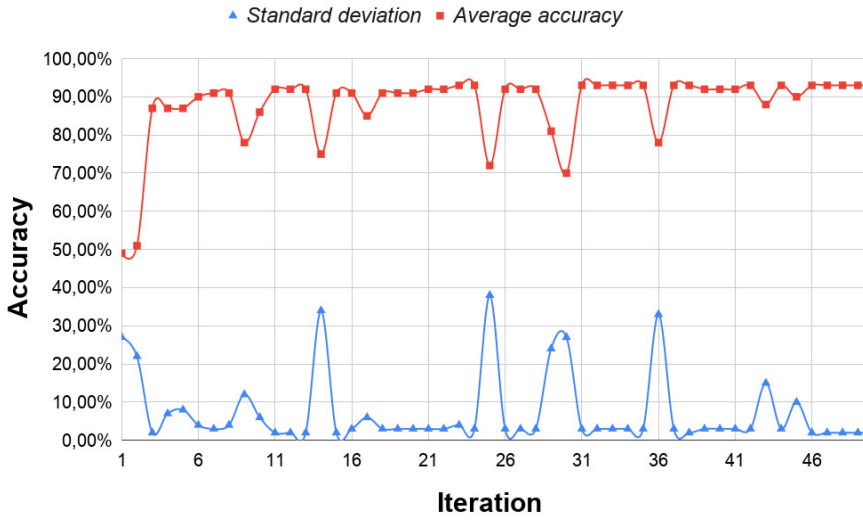


Figure 8. Classification accuracy achieved by the population of neural networks

digits in speech recordings. This approach allowed us to avoid creating the neural network architecture ourselves. Since the architecture was created with the use of an automated procedure (the coevolution process), we only had to define the elements which the network would consist of and the amount of resources we wanted to dedicate to searching for an appropriate architecture. First, in Experiment 1, we have examined what is the optimal size of the training dataset (size of an individual in the training set population). We also showed that reducing the size of individuals in the population of training subsets resulted in decreasing the neural network training time. As a consequence, the pace of the genetic algorithm accelerated. At the same time, the experiment has demonstrated that even though the size of fitness predictors was reduced, the classification accuracy did not significantly decrease. This allowed the evolutionary process to make progress towards the optimal architecture, while reducing the amount of resources required. One needs to remember, though, that trading off the accuracy for resource consumption may affect the evolution and lead it in a wrong direction. In order to avoid rendering the presented approach ineffective, it is beneficial to empirically confirm that reducing the size of fitness predictors does not lead to significant negative changes in the fitness metric values, e.g. in our case reduction of the classification accuracy. If possible the optimal size should be determined through rigorous experimentation with a wide variety of fitness predictor sizes.

In Experiment 2, the neural network obtained in the process of co-evolution achieved a classification accuracy of 97.05 %. This value is comparable to the best results achieved on the AudioMnist dataset (recognition accuracy of $95.82\,\% \pm 1.49\,\%$), described in [10]. The results obtained during the tests of the Pytorch DNN Evo-

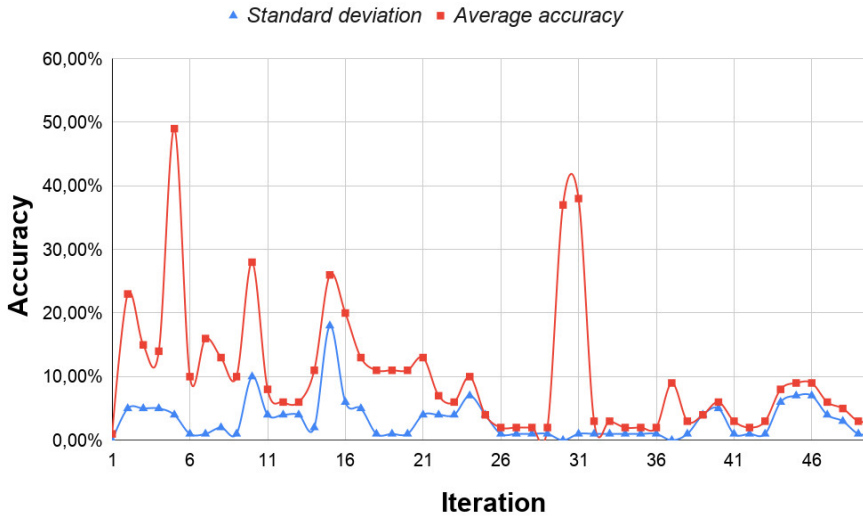


Figure 9. Classification accuracy achieved by the population of training datasets. The accuracy declines as expected: The evolution of the population of training datasets generates individuals which are harder and harder to classify correctly by the neural networks.

lution framework confirmed that coevolution can be used to search for the optimal neural network architecture, used to solve the problem of sound classification.

The positive results of the experiments prove that the data representation based on the mel cepstral coefficients is more memory-efficient than the spectrogram. The mel cepstral coefficients are approximately 10 times smaller than a spectrogram representation of the same waveform. Data complexity reduction is especially important when training neural networks as it allows to reduce the training time.

In the future, the Pytorch DNN Evolution framework can further be extended with support for other datasets. This would allow to verify whether the coevolution algorithm works also for other types of problems that may use other data types. Currently, Pytorch DNN Evolution offers the creation of neural networks by using two types of layers: convolutional layer and dense layer. We believe that extending it with new types of layers, e.g. pooling layers, recursive layers, or dropout layers, would help to apply it to more domains.

Data Availability

The datasets used, generated and analysed during the current study are available in publicly accessible repositories [5] or can be provided from the corresponding author on a reasonable request.

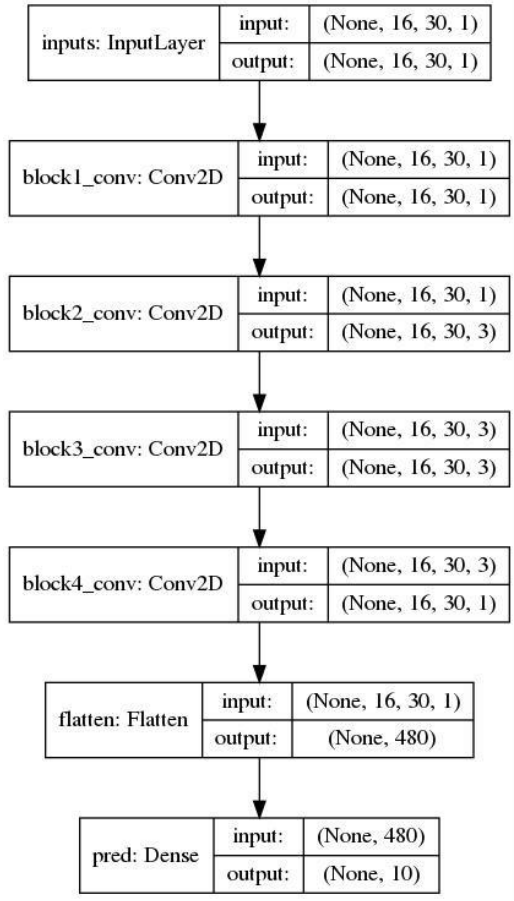


Figure 10. Neural network architecture generated by Pytorch DNN Evolution framework. The left side of the rectangle denotes the name and type the NN layer: *InputLayer* – first layer which does not transform data, *Conv2D* – convolution of input data, *Flatten* – change the format of data from a multidimensional vector to an array, *Dense* – fully connected layer. The right side specifies the format of data at the input and output to and from a given layer.

Acknowledgements

The research presented in this paper was supported by the funds assigned to AGH University of Krakow by the Polish Ministry of Education and Science. Our thanks go also to the PL-Grid infrastructure resources of the ACC CYFRONET AGH, where experiments have been carried out.

REFERENCES

[1] López, G.—Quesada, L.—Guerrero, L. A.: Alexa Vs. Siri Vs. Cortana Vs. Google Assistant: A Comparison of Speech-Based Natural User Interfaces. In: Nunes, I. L. (Ed.): Advances in Human Factors and Systems Interaction. Springer International Publishing, Cham, 2018, pp. 241–250.

[2] Tombeng, M. T.—Najoan, R.—Karel, N.: Smart Car: Digital Controlling System Using Android Smartwatch Voice Recognition. 2018 6th International Conference on Cyber and IT Service Management (CITSM), 2018, pp. 1–5, doi: 10.1109/CITSM.2018.8674359.

[3] Gundogdu, K.—Bayrakdar, S.—Yucedag, I.: Developing and Modeling of Voice Control System for Prosthetic Robot Arm in Medical Systems. Journal of King Saud University - Computer and Information Sciences, Vol. 30, 2018, No. 2, pp. 198–205, doi: https://doi.org/10.1016/j.jksuci.2017.04.005.

[4] Pytorch DNN Evolution Framework. 2018, `https://gitlab.com/pkoperek/pytorch-dnn-evolution` (Accessed: 2022-03-02).

[5] AudioMNIST Dataset. `https://github.com/soerenab/AudioMNIST` (Accessed: 2021-01-07).

[6] Niebudek-Bogusz, E.—Woznicka, E.—Korczak, I.—Śliwińska Kowalska, M.: The Applicability of Formant Voice Analysis in Diagnostics of Functional Voice Disorders. Otorynolaryngologia, Vol. 8, 2009, pp. 184–192.

[7] Mirzaei, G.—Majid, M. W.—Ross, J.—Jamali, M. M.—Gorsevski, P. V.—Frizado, J. P.—Bingman, V. P.: The BIO-Acoustic Feature Extraction and Classification of Bat Echolocation Calls. 2012 IEEE International Conference on Electro/Information Technology, 2012, pp. 1–4, doi: 10.1109/EIT.2012.6220700.

[8] Li, Z.—Peng, P.—He, Z.—Wang, L.: Automatic Classification of Microseismic Signals Based on MFCC and GMM-HMM in Underground Mines. Shock and Vibration, Vol. 2019, 2019, doi: 10.1155/2019/5803184.

[9] Bridle, J. S.—Brown, M. D.: An Experimental Automatic Word-Recognition System. JSRU Report, Vol. 1003, 1974, No. 5.

[10] Becker, S.—Ackermann, M.—Lapuschkin, S.—Müller, K.—Samek, W.: Interpreting and Explaining Deep Neural Networks for Classification of Audio Signals. CoRR, 2018, arXiv: 1807.03418.

[11] Krizhevsky, A.—Sutskever, I.—Hinton, G. E.: Imagenet Classification with Deep Convolutional Neural Networks. Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, Curran Associates Inc., Red Hook, NY, USA, NIPS '12, 2012, pp. 1097–1105.

[12] Zhang, B.—Leitner, J.—Thornton, S.: Audio Recognition Using Mel Spectrograms and Convolution Neural Networks. Technical Report, 2019.

[13] Simonyan, K.—Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. 2014, arXiv: 1409.1556.

[14] Lecun, Y.—Bottou, L.—Bengio, Y.—Haffner, P.: Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE, Vol. 86, 1998, No. 11, pp. 2278–2324, doi: 10.1109/5.726791.

[15] Guiming, D.—Xia, W.—Guangyan, W.—Yan, Z.—Dan, L.: Speech Recognition Based on Convolutional Neural Networks. 2016 IEEE International Conference on Signal and Image Processing (ICSIP), 2016, pp. 708–711, doi: 10.1109/SIPROCESS.2016.7888355.

[16] Huang, J. T.—Li, J.—Gong, Y.: An Analysis of Convolutional Neural Networks for Speech Recognition. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 4989–4993, doi: 10.1109/ICASSP.2015.7178920.

[17] Rumelhart, D. E.—Hinton, G. E.—Williams, R. J.: Learning Representations by Back-Propagating Errors. Nature, Vol. 323, 1986, No. 6088, pp. 533–536, doi: 10.1038/323533a0.

[18] Hochreiter, S.—Schmidhuber, J.: Long Short-Term Memory. Neural Computation, Vol. 9, 1997, No. 8, pp. 1735–1780.

[19] Vaswani, A.—Shazeer, N.—Parmar, N.—Uszkoreit, J.—Jones, L.—Gomez, A. N.—Kaiser, L.—Polosukhin, I.: Attention Is All You Need. CoRR, 2017, arXiv: 1706.03762.

[20] Chen, M. X.—Firat, O.—Bapna, A.—Johnson, M.—Macherey, W.—Foster, G.—Jones, L.—Schuster, M.—Shazeer, N.—Parmar, N.—Vaswani, A.—Uszkoreit, J.—Kaiser, L.—Chen, Z.—Wu, Y.—Hughes, M.: The Best of Both Worlds: Combining Recent Advances in Neural Machine Translation. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 76–86, doi: 10.18653/v1/P18-1008.

[21] Wu, Y.—Schuster, M.—Chen, Z.—Le, Q. V.—Norouzi, M.—Macherey, W.—Krikun, M.—Cao, Y.—Gao, Q.—Macherey, K.—Klingner, J.—Shah, A.—Johnson, M.—Liu, X.—Kaiser, L.—Gouws, S.—Kato, Y.—Kudo, T.—Kazawa, H.—Stevens, K.—Kurian, G.—Patil, N.—Wang, W.—Young, C.—Smith, J.—Riesa, J.—Rudnick, A.—Vinyals, O.—Corrado, G.—Hughes, M.—Dean, J.: Google's Neural Machine Translation System: Bridging the Gap Between Human and Machine Translation. CoRR, 2016, arXiv: 1609.08144.

[22] Ren, P.—Xiao, Y.—Chang, X.—Huang, P.—Li, Z.—Chen, X.—Wang, X.: A Comprehensive Survey of Neural Architecture Search: Challenges and Solutions. CoRR, 2020, arXiv: 2006.02903.

[23] Chitty-Venkata, K. T.—Emani, M.—Vishwanath, V.—Somani, A. K.: Neural Architecture Search for Transformers: A Survey. IEEE Access, Vol. 10, 2022, pp. 108374–108412, doi: 10.1109/ACCESS.2022.3212767.

[24] Cui, J.—Chen, P.—Li, R.—Liu, S.—Shen, X.—Jia, J.: Fast and Practical Neural Architecture Search. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 6508–6517, doi: 10.1109/ICCV.2019.00661.

[25] Cai, H.—Zhu, L.—Han, S.: ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware. International Conference on Learning Representations, 2019, doi: 10.48550/arXiv.1812.00332.

[26] Zela, A.—Elsken, T.—Saikia, T.—Marrakchi, Y.—Brox, T.—Hutter, F.:

Understanding and Robustifying Differentiable Architecture Search. CoRR, 2019, arXiv: 1909.09656.

[27] Gao, J.—Xu, H.—Shi, H.—Ren, X.—Yu, P. L. H.—Liang, X.—Jiang, X.—Li, Z.: AutoBERT-Zero: Evolving BERT Backbone from Scratch. CoRR, 2021, arXiv: 2107.07445.

[28] Wistuba, M.: Deep Learning Architecture Search by Neuro-Cell-Based Evolution with Function-Preserving Mutations. In: Berlingerio, M., Bonchi, F., Gärtner, T., Hurley, N., Ifrim, G. (Eds.): Machine Learning and Knowledge Discovery in Databases. Springer International Publishing, Cham, 2019, pp. 243–258.

[29] Real, E.—Aggarwal, A.—Huang, Y.—Le, Q. V.: Regularized Evolution for Image Classifier Architecture Search. CoRR, 2018, arXiv: 1802.01548.

[30] Guo, Z.—Zhang, X.—Mu, H.—Heng, W.—Liu, Z.—Wei, Y.—Sun, J.: Single Path One-Shot Neural Architecture Search with Uniform Sampling. CoRR, 2019, arXiv: 1904.00420.

[31] Funika, W.—Koperek, P.: Co-Evolution of Fitness Predictors And deep Neural Networks. In: Wyrzykowski, R., Dongarra, J., Deelman, E., Karczewski, K. (Eds.): Parallel Processing and Applied Mathematics. Springer International Publishing, Cham, 2018, pp. 555–564.

[32] LeCun, Y.—Cortes, C.: MNIST Handwritten Digit Database. 2010, http://yann.lecun.com/exdb/mnist/.

[33] Natsiou, A.—O'Leary, S.: Audio Representations for Deep Learning in Sound Synthesis: A Review. CoRR, 2022, arXiv: 2201.02490.

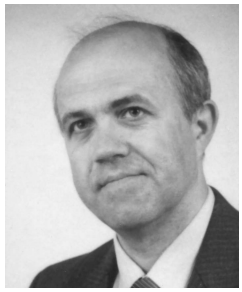
[34] Laguarta, J.—Hueto, F.—Subirana, B.: COVID-19 Artificial Intelligence Diagnosis Using Only Cough Recordings. IEEE Open Journal of Engineering in Medicine and Biology, Vol. 1, 2020, pp. 275–281, doi: 10.1109/OJEMB.2020.3026928.

[35] Krizhevsky, A.: Learning Multiple Layers of Features from Tiny Images. Technical Report, 2009, pp. 32–33, https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[36] Documentation of Torchaudio Library. (Accessed: 2021-01-07).

[37] Qu, X.—Wei, P.—Gao, M.—Sun, Z.—Ong, Y. S.—Ma, Z.: Synthesising Audio Adversarial Examples for Automatic Speech Recognition. Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, New York, NY, USA, KDD '22, 2022, pp. 1430–1440, doi: 10.1145/3534678.3539268.

[38] Chen, G.—Zhao, Z.—Song, F.—Chen, S.—Fan, L.—Wang, F.—Wang, J.: Towards Understanding and Mitigating Audio Adversarial Examples for Speaker Recognition. 2022, doi: 10.48550/ARXIV.2206.03393.



Włodzimierz Funika works at the Institute of Computer Science of the AGH University in Krakow (Poland). His main research interests are in distributed computing, tool construction, performance analysis and visualization, data science, and machine learning. Involved in many EU-funded projects and Polish-wide projects: PL-Grid and others.

Pawel Koperek is a researcher in the field of machine learning. He received his Master of Science degree in computer science in 2010 from the AGH University of Science and Technology, where he studied at the Faculty of Electrical Engineering, Automatics, Computer Science and Electronics. He has a passion for exploring new ideas in machine learning and is particularly interested in evolutionary algorithms, deep learning and deep reinforcement learning and their practical applications (e.g., in the automatic cloud resource management domain).

Tomasz Wiewióra graduated with his Master's in the field of computer science at the Faculty of Computer Science, Electronics and Telecommunications of the AGH University of Science and Technology in Krakow. He works as a programmer in a large company, where he deals with the implementation of solutions for clients from the banking industry.

Computing and Informatics, Vol. 42, 2023, 364–391, doi: 10.31577/cai_2023_2_364

AN APPROACH BASED ON GENETIC AND GRASSHOPPER OPTIMIZATION ALGORITHMS FOR DYNAMIC LOAD BALANCING IN CLOUDIOT

Sofiane Benabbes, Sofiane Mounine Hemam

ICOSI Laboratory, Abbes Laghrour University
Khenchela, Algeria
e-mail: {benabbes.sofiane, hemam.sofiane}@univ-khenchela.dz

Abstract. *CloudIoT* is a new paradigm, which has emerged as a result of the combination of Cloud Computing (CC) and the Internet of Things (IoT). It has experienced a growing and rapid development, and it has become more popular in information and technology (IT) environments because of the advantages it offers. However, due to a strong use of this paradigm, especially in smart cities, the problem of imbalance load has emerged. Indeed, to satisfy the needs of the user, the intelligent objects send the collected data to the virtual machines (VMs) of the cloud in order to be processed. So, it is necessary to have an idea about the load of its VM. Thus, the problem of load balancing between VMs is strongly related to the technique used for the VMs selection. To tackle this problem, we propose in this paper a task scheduler called Scheduler Genetic Grasshopper Algorithm (SGGA). It allows to ensure a dynamic load balancing, as well as the optimization of the makespan and the resource usage. Our proposed SGGA is based on the combination of Genetic Algorithm (GA) and Grasshopper Optimization Algorithm (GOA). First, the tasks sent by the IoTs are mapped to the VMs in order to build the initial population, then SGGA performs the genetic algorithm, which has expressed a considerable performance. However, the weakness of the GA is marked by its heaviness caused by the mutation operator, especially when the number of tasks increases. Because of this insufficiency, we have replaced the mutation operator with the grasshopper optimization algorithm. The results of the experiments show that our approach (SGGA) is the most efficient, compared to the recent approaches, in terms of the response time to obtain the optimal solution, makespan, throughput, an average resource utilization rate and the hypervolume indicator.

Keywords: CloudIoT, dynamic load balancing, GA, GOA, task scheduler

1 INTRODUCTION

Nowadays, IoT and Cloud Computing are two new distributed computing technologies. On the one hand, the IoT allows to transform real-world objects into smart objects [20], which share their data, their situations and their interactions with other interconnected objects. The IoT is generally characterized by widely distributed objects with limited processing and storage capabilities. These objects suffer from performance, reliability, privacy and security issues [4]. On the other hand, Cloud Computing is a technology that has a network with unlimited storage capacities and computing power. Moreover, it offers the flexibility and robustness of dynamic data integration from heterogeneous sources [19].

CloudIoT is a new paradigm that has emerged as a result of the combination of cloud and IoT. It enables intelligent use of applications, information and infrastructure in a fair and reasonable manner. Although IoT and Cloud Computing are two different technologies, their functionalities are almost complementary [17], in terms of nature of existence, processing capacity, storage capacity, connectivity and Big Data [19].

In this environment, IoTs send their tasks to Cloud Computing for processing or storage, by mapping them to the various hardware and software resources represented by virtual machines. When distributing data to be processed to virtual machines (VMs), some of them will be overloaded while others will be unloaded or inactive [20, 11]. Thus, a load balancing mechanism is then necessary because it allows to manage the allocation of VMs to tasks sent by IoTs. It thus allows the optimization of makespan, throughput and the rentable resource usage.

Several scientific research on load balancing in Cloud Computing has focused on task allocation. Each scheduling algorithm is based on one or more parameters. The most targeted objectives are the overall execution time, the cost, and the use of resources (which also indicates the quality of service (QoS)) [22, 7, 9]. Several task scheduling algorithms based on metaheuristic algorithms, such as BGA, HGOW-ABC, GWO and ACO algorithms have been proposed for cloud load balancing [9, 14, 23, 25]. The researchers have developed techniques to reduce makespan, and assign tasks to VMs in a balanced way, using different optimization techniques such as genetic algorithm, gray wolf algorithm, the bee colony algorithm and the ant colony algorithm.

To deal with the above cited problem, we propose in this work a scheduler which, ensures a dynamic load balancing in the CloudIoT. The proposed work allows the improvement of the makespan, the throughput and the average rate of the resource usage. Our proposed scheduler allows the tasks distribution, sent by the IoTs, over the virtual machines. It ensures maximum throughput with a shorter execution time, as well as a more efficient use of resources. Our proposed approach called SGGA, is based on the combination between genetic algorithm and grasshopper optimization algorithm. So, the proposed selection and crossing operators phases of our approach allow to avoid the appearance of the same chromosome several times on one hand, and to avoid the heaviness caused by the mutation operator phase [18],

especially when the number of tasks increases, we have replaced this phase by the grasshopper optimization algorithm on the second hand.

The results obtained show that the proposed approach is more efficient compared to the most recent works (BGA, HGOW-ABC, GWO and ACO) in terms of the time to reach the optimal solution, the makespan, the throughput, the average resource utilization ratio and the hypervolume indicator.

Our paper is structured as follows: after the introduction, Section 2 is devoted to related works. In Section 3 we present the proposed approach and its different components. Section 4 presents case studies. Section 5 is reserved for experimental results and discussion, and we end this paper with a conclusion in Section 6.

2 RELATED WORKS

In this section, we review the most recent works that address the problem of load balancing in the cloud environment. The majority of the proposed works are mainly based on task scheduling to achieve the objective of ensuring load balancing between different components. Metaheuristic algorithms are classified into four categories [27, 28]: Firstly, the swarm-intelligence algorithms such as [30] which proposed a new metaheuristic algorithm called Giant Trevally Optimizer (GTO) inspired by the hunting behaviour of giant trevally.

Secondly, human-based algorithms such as [31] which proposed a metaheuristic algorithm called Group Teaching Optimisation Algorithm (GTOA), where they adjusted additional control parameters for solving different optimisation problems. Thirdly, evolutionary algorithms such as [32] have proposed a new approach called the Tree Growth Algorithm (TGA). This approach is inspired by the competition of trees for light and food. And fourthly, science-based algorithms such as [28] who have proposed a new metaheuristic called Crystal Structure Algorithm (CryStAl). This approach is inspired by the principles underlying the formation of crystal structures from the addition of the base to lattice points. In this paper we focus on the first two categories for their impressive results when compared to each other.

Several swarm-intelligence algorithms and human-based algorithms have been proposed and applied for task scheduling in the cloud environment. There are two types of these algorithms:

1. based on the exploitation of the best solution among the previous results, called a local search, and
2. based on the exploration of new areas of the solution space or the sudden prospection of a new solution search space.

The most interesting work in this context is reviewed below. So, at first we present the human-based algorithms category, followed by the swarm-intelligence algorithms category.

Makasarwala and Hazari [24], Kaur and Sachdeva [22] used GA for load balancing in Cloud Computing. The proposal of Makasarwala and Hazari [24] provides load balancing and reduces response time without considering resource utilization rate and QoS. On the other hand, Kaur and Sachdeva [22] proposed an improved GA to reduce the execution time of task migration in Cloud Computing. This proposal not only ensures the proper use of resources, but it also saves energy. However, the response time is high.

Gulbaz et al. [9] presented the Balancer Genetic Algorithm (BGA) to improve makespan and load balancing. BGA relies on a load balancing mechanism that takes into account the actual load assigned to virtual machines. The need to opt for multi-objective optimization for the improvement of load balancing and Makespan is also highlighted. The simulation showed significant improvement on makespan, throughput and load balancing.

For the swarm-intelligence algorithms category, we can find several works. We cite in this paper the most recent and important ones.

The work presented by Li and Wu [10]; Shafahi and Yari [25] used the Ant Colony Algorithm (ACO) to dynamically schedule tasks. The scheduler acts as an ant looking for food. The experimental results of the simulations, of the two approaches, give better performance in comparison to others. They reduce task execution time and improve system resource utilization, and they keep the system balanced.

Muthsamy and Suganthe [12] and Shen et al. [26] used the Artificial Bee Colony Optimization (ABC) algorithm. Thereby, Muthsamy and Suganthe [12] proposed a task scheduler based on optimizing artificial bee foraging (TSABF) that takes in charge the QoS, makespan, response time, execution time and task priority. To achieve optimal scheduling, tasks are scheduled preemptively. Task preemption is done to reduce the response time and execution time of tasks belonging to different priorities. While the work of Shen et al. [26] presented a study to ensure load balancing in a cloud data center, based on efficient resource utilization and power consumption management. They have optimized the (ABC) method using a load balancing algorithm, and intelligent classification of virtual machines. This study was validated by a simulation on CloudSim.

Arulkumar [3] obtained their best simulation results from the Water Wave Algorithm (WWA). The latter was proposed for resource planning in a cloud environment. The proposed work takes into consideration the four parameters: throughput, response time, resource utilization, and scalability.

Alguliyev et al. [1] presented a novel multi-criteria optimization method for weighted task scheduling based on the Particle Swarm Optimization (PSO) algorithm. The simulation showed that the method migrates tasks from overloaded virtual machines to less loaded virtual machines, ensuring, thus, an overall balanced system.

Patel et al. [23] proposed a task allocation approach based on gray wolf optimization (GWO) for load balancing in the containerized cloud. The approach ensures the load balancing and makespan minimization. Gohil and Patel [21] proposed (IGWO)

which is an improvement of the GWO algorithm. They increased the coefficients of the best solutions α, β and δ to calculate the next solutions, which gives a perfect balance and guarantees a quasi-optimal solution.

Natesan and Chokkalingam [13] also improved (GWO). They proposed Performance-Cost Grey Wolf Optimization (PCGWO) to reduce both processing time and cost in accordance with the objective function. The simulation results of the proposed technique show a complete reduction in the time and cost of performing the tasks.

Ragmani et al. [15] proposed a hybrid algorithm, based on the concepts of Fuzzy Logic and Ant Colony Optimization (Fuzzy-ACO), to improve load balancing in Cloud Computing. This approach takes into account load balancing goals and response time. Simulations performed on CloudAnalyst have shown that the proposed approach improves load balancing in the Cloud, minimizing response time by up to 82 %, processing time by up to 90 % and total cost up to 9 %.

Ouhame et al. [14] integrated the GWO algorithm with Artificial Bee Colony (HGWOABC) to improve the cloud resource allocation system. This technique improved the parameters of load balancing in Cloud Computing by 1.25 %.

In Table 1, we conclude this section by specifying the different load balancing parameters of each approach.

Year	Approach	Makespan	Throughput	Res Utzt	QoS	Energy	Cost	HV
2016	[24]	Yes	No	No	No	No	No	No
2017	[22]	Yes	No	Yes	Yes	Yes	No	No
2019	[10]	Yes	No	Yes	Yes	No	No	No
2021	[25]	Yes	No	Yes	Yes	No	No	No
2020	[12]	Yes	Yes	Yes	Yes	No	No	No
2019	[26]	No	No	Yes	Yes	Yes	No	No
2021	[9]	Yes	Yes	Yes	Yes	No	No	No
2019	[1]	No	No	Yes	Yes	No	No	No
2020	[23]	Yes	Yes	No	No	No	No	No
2018	[21]	Yes	Yes	Yes	Yes	No	No	No
2019	[15]	Yes	No	Yes	Yes	No	Yes	No
2020	[14]	Yes	No	Yes	Yes	No	No	No

Table 1. Summary of balancing parameters for each approach

In this paper, a new dynamic load balancing approach has been proposed using a task scheduler based on the pairing between Genetic Algorithm (GA) and Grasshopper Optimization Algorithm (GOA) called Scheduler GA-GOA Algorithm (SGGA).

We have realized several hybridizations tests to replace the mutation by other algorithms, and the GA-GOA hybridization give the best result. As shown in the following Table 2 for example the makespan.

Our balancing based on the improvement of three parameters: makespan, throughput and resource utilization (QoS).

Tasks	GA	GA-GOA	GA-PSO	GA-ABC	GA-ANTLion	GA-ACO
100	0.32584	0.15874	0.32101	0.31524	0.18524	0.25471
200	0.78954	0.17543	0.78814	0.76214	0.20574	0.31241
500	1.01458	0.20145	1.01247	1.00024	0.26472	0.43120

Table 2. Testing the "makespan" result of the different GA hybrids

3 THE PROPOSED APPROACH

In this section, we will present the proposed global architecture; and the detailed architecture that contains the components of our scheduler.

3.1 The Overall Architecture Proposed

Our proposed approach is developed to provide load balancing in the CloudIoT. It contains several components on the IoT and Cloud sides. In our approach, we focus our interest in tasks that will be processed at the Cloud level. The smart objects send their tasks to the scheduler (SGGA) which will map and assign them to the different virtual machines. The role of the scheduler will be detailed in the next section. Figure 1 shows the overview of the overall architecture proposed in CloudIoT.

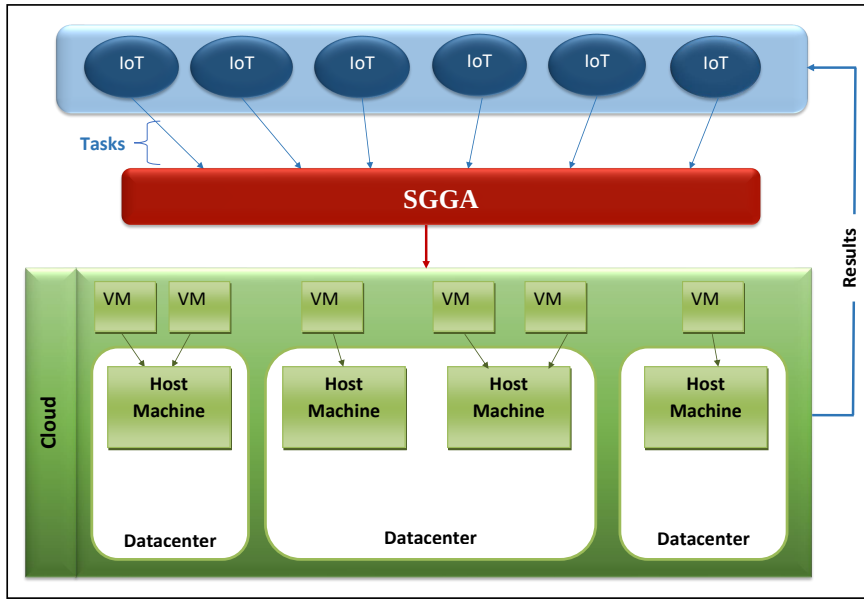


Figure 1. Presentation of the global architecture proposed in CloudIoT

3.2 Detailed Architecture of the Proposed Approach "SGGA"

An effective load balancing is relied to a robust and reliable task scheduler. To this end, we have developed a task scheduler based on both the grasshopper optimization algorithm and the genetic algorithm. The latter gives very satisfactory results mainly because of its flexibility and robustness. However, this algorithm suffers from a heaviness at the level of its mutation operator [18], especially when the tasks number is increased. For this reason, and in order to deal with this limitation, we propose to replace the mutation operator with the grasshopper optimization algorithm (see Figure 2). The latter is classified among the new meta-heuristic algorithms, and it is inspired by the behavior of grasshoppers [16].

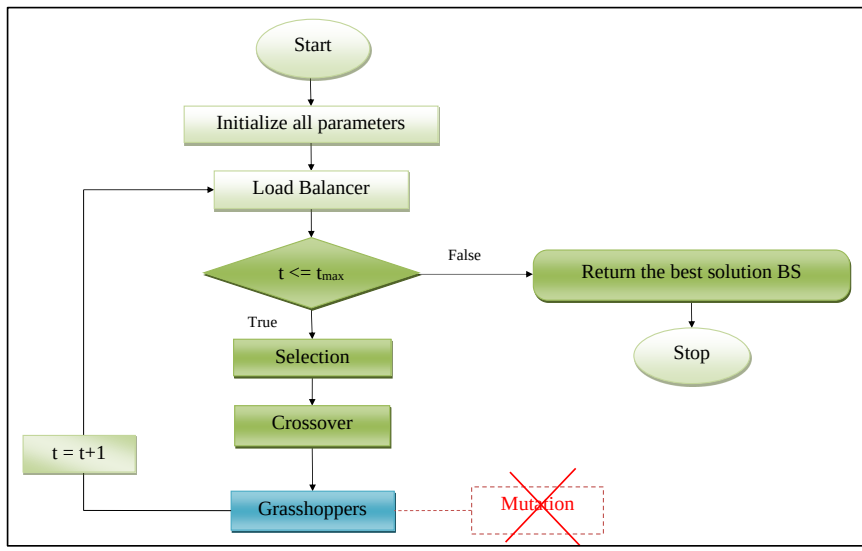


Figure 2. Flow diagram of the Scheduler GGA

The algorithm of the proposed approach; which will be detailed later, initializes randomly the position matrix. The rows and columns of this matrix are respectively solutions and tasks (Table 3).

The solutions number is equal to 100, and each matrix cell contains the CPU speed of a VM. After the initialization phase the SGGA, in each iteration, calculates the fitness function of each solution (row). then, it sorts the solutions in ascending order according to the calculated fitness functions. Thus, the best solution will be at the top of the population (100 solutions). To improve the population quality of the position matrix, we select the 7 % of the best solutions, and then we apply the selector and crossover operators of GA to obtain 50 % of the new populations (as indicated in Figure 3). At the end of an iteration, the grasshopper optimization algorithm is applied to obtain the second half-population (the second 50 %) from

M Positions	T1	T2	T3	T4	T5	T6	T7	...	Tm
P1	S-VM	S-VM	S-VM	S-VM	S-VM	S-VM	S-VM	...	S-VM
P2	S-VM	S-VM	S-VM	S-VM	S-VM	S-VM	S-VM	...	S-VM
P3	S-VM	S-VM	S-VM	S-VM	S-VM	S-VM	S-VM	...	S-VM
P4	S-VM	S-VM	S-VM	S-VM	S-VM	S-VM	S-VM	...	S-VM
P5	S-VM	S-VM	S-VM	S-VM	S-VM	S-VM	S-VM	...	S-VM
P6	S-VM	S-VM	S-VM	S-VM	S-VM	S-VM	S-VM	...	S-VM
P7	S-VM	S-VM	S-VM	S-VM	S-VM	S-VM	S-VM	...	S-VM
...	S-VM	S-VM	S-VM	S-VM	S-VM	S-VM	S-VM	...	S-VM
Pn	S-VM	S-VM	S-VM	S-VM	S-VM	S-VM	S-VM	...	S-VM

Table 3. The population Matrix loaded by CPU speed of VMs (S-VM)

the value of the first half-population.

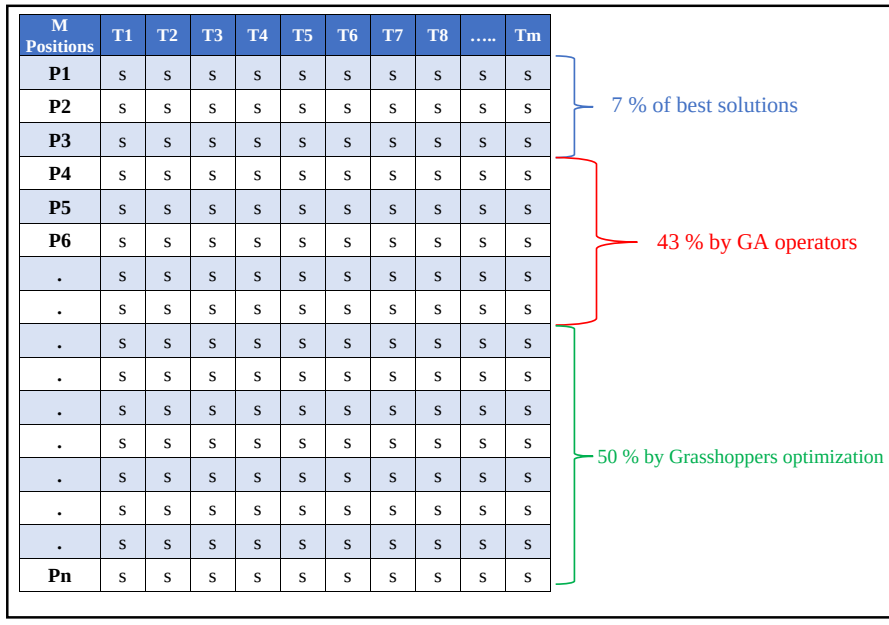


Figure 3. The new population Matrix composition

3.3 The Components of the SGGA

In this sub-section, we present, in details, the roles of the fourth components of our approach, as well as their algorithms.

Algorithm 1: SGGA

Input: Tasks vector, VMs vector
Output: Mapping of Tasks to VMs
$M[x, n] \leftarrow$ random(CPU, VMs);
$t \leftarrow 1$;
while $t \leq t_{max}$ **do**
  for $i \leftarrow 2$ *to* n **do**
    Fitness$[i] \leftarrow$ LoadBalancer$(M[x, n])$;
  end
  BS $\leftarrow M[1, n]$;
  newM $\leftarrow 1$;
  $j \leftarrow 1$;
  SelectionAlgo (Fitness$[x], M[x, n]$, M_Select$[y, n]$, Array_Select$[y^2 - y]$);
  for $i \leftarrow 1$ *to* y **do**
    $M[i, n] \leftarrow$ M_Select$[i, n]$;
  end
  while $newM < s$ **do**
    Parent1 $\leftarrow$ M_Select[Array_Select[newM].part1, n];
    Parent2 $\leftarrow$ M_Select[Array_Select[newM].part2, n];
    CrossoverAlgo(Parent1, Parent2, newParent1, newParent2);
    $M1[j, n] \leftarrow$ newParent1;
    $M1[j + 1, n] \leftarrow$ newParent2;
    newM $\leftarrow$ newM + 1;
    $j \leftarrow j + 2$;
  end
  while $newM \leq x$ **do**
    $M2[] \leftarrow$ GrasshoppersAlgo$(M1)$;
    newM $\leftarrow$ newM + 1;
  end
  $M[] \leftarrow M1[] + M2[]$; (concatenate the two matrices M1 and M2 in M);
  $t \leftarrow t + 1$;
end

In this context, we assume that we have a set of tasks $T = \{T_1, T_2, \ldots, T_n\}$, where each of them is characterized by its size in Kilobytes (KB), and a set of virtual machines $VM = \{VM_1, VM_2, \ldots, VM_m\}$, where each of them is characterized by its computing capacity (CPU) in million instructions per second (mips).

The population is represented by positions (solutions), each of them has its own fitness function. The solution can be represented using binary Mapping Matrix (MP), where rows indicate VMs and columns indicate Tasks. In the example below, the mapping matrix (MP) represents the set $VM_1(T_2, T_6)$, $VM_2(T_4, T_5)$, $VM_3(T_1, T_3)$.

$$\mathrm{MP} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}. \tag{1}$$

SGGA starts by initializing the population randomly, each chromosome is represented by a speed line of the virtual machines and the indices are the indices of the tasks.

In the rest of the section, we will detail the role of each component of the proposed approach.

3.3.1 The Load Balancer Component

The role of this component is to calculate the fitness functions of each solution by using formulas from (2 to 10) then, it stores them in the fitness vector. After that, it sorts the fitness vector in ascending manner as well as the solutions matrix according to the fitness vector. Thus, the best solution (BS) will be at the top of the solutions matrix.

Since our SGGA is a multi-objective thus, to improve load balancing, the proposed fitness function must combine between two objectives (the makespan and the average load utilization) as indicated in the formula (2).

$$f(P_i) = (\mathrm{makespan} + \mathrm{AvgLoad}), \tag{2}$$

where makespan is the maximum time taken by any virtual machine, given by:

$$\mathrm{makespan} = (\max(T_{V_M})_{j\in 1,\ldots,M}), \tag{3}$$

where T_{VMj} is the processing time of a specific VM_j. AvgLoad is between 0 and 1. It is the average load of all virtual machines for a specific position, it is calculated according to the formula (4):

$$\mathrm{AvgLoad} = \left(1 - \frac{\sum_{j=1}^{M} \mathrm{Load}_{VMj}}{m}\right). \tag{4}$$

The value of Load_{VMj} is calculated to find the part used by the virtual machine VM_j, according to the following equation:

$$\mathrm{Load}_{VMj} = \left(\frac{\mathrm{TaskMap}_{VMj}}{\mathrm{Mappe}_{VMj}} * 100\right). \tag{5}$$

Formula (6) normalizes the value of Load_{VMj} to avoid negative values that affect AvgLoad: [9].

$$\text{Load}_{VMj} = \begin{cases} \frac{100-(Load_{VMj}-100)}{100}, & \text{if } \text{Load}_{VMj} > 100, \\ 0, & \text{if } \text{Load}_{VMj} < 0, \\ \frac{\text{Load}_{VMj}}{100}, & \text{else.} \end{cases} \tag{6}$$

The Equation (7) presents the size of the tasks mapped by the virtual machine VM_j, using the binary Mapping Matrix already presented in (1):

$$\text{TaskMap}_{VMj} = \left(\sum_{i=1}^{N} \text{SizeTask}[i] * \text{MP}[i,j] \right). \tag{7}$$

Since each task is characterized by its value size and, each virtual machine is characterized by its computing power thus, these values are used to calculate the load according to formula (8):

$$\text{Mappe}_{VMj} = \left(\sum_{i=1}^{N} (\text{SizeTask}[i]) * \text{MappeRatio}_{(VMj)} \right), \tag{8}$$

where $\text{MappeRatio}_{(VMj)}$ is a ratio used to calculate the maximum size of tasks that can be mapped to the VM_j, it is calculated as follows:

$$\text{MappeRatio}_{(VMj)} = \left(\frac{\text{CPU}_{VMj}}{\sum_{j=1}^{M} \text{CPU}_{VMj}} \right). \tag{9}$$

Finally, formula (10) is used to calculate the average resource utilization rate (AVG_{UR}). This rate is between [0.1] and it is calculated as follows:

$$\text{AVG}_{UR} = \left(\frac{(\sum_{j=1}^{M} T_{VMj})/M}{\text{makespan}} \right). \tag{10}$$

The pseudo Algorithm 2 **LoadBalancer** presents the different actions that allow to calculate the fitness function and the combination between the two objectives of our approach.

3.3.2 The Selector Component

This component generates another matrix, called M_Select, and a vector, called Array_Select. At the beginning, this matrix contains, the best 7 % chromosomes. The Array_Select vector contains the indices of all chromosomes that can be generated and used for crossing. This vector makes it possible to avoid the crossing

Algorithm 2: Load Balancer component behavior

Input: $M[x, n]$
Output: Fitness$[x]$, $M[x, n]$
Initialize parameters;
LoadMP[];
Calculate:
SizeTask$[i]$, CPUVM$[j]$;
Mappe$_{VM}[j]$;
TaskMap$_{VM}[j]$;
Load$_{VM}[j]$;
AvgLoad;
Makespan;
Fitness $\leftarrow$ Makespan + AvgLoad;
Descending sort(vector of fitness functions);
Descending sort(matrix of positions) according Fitness sort;

between the same chromosome on one side, and also to avoid the crossing of two chromosomes several times on the other side (Figure 4).

The pseudo Algorithm 3 **SelectorAlgo**, allows to present the different actions which constitute the functionalities of this component.

Algorithm 3: Selector component behavior

Input: Fitness$[x]$, $M[x, n]$
Output: MSelect$[y, n]$, ArraySelect$[y^2 - y]$
Select the 7 % of best solutions;
Create the matrix of best solutions;
Create the array of parent index who go to crossover;

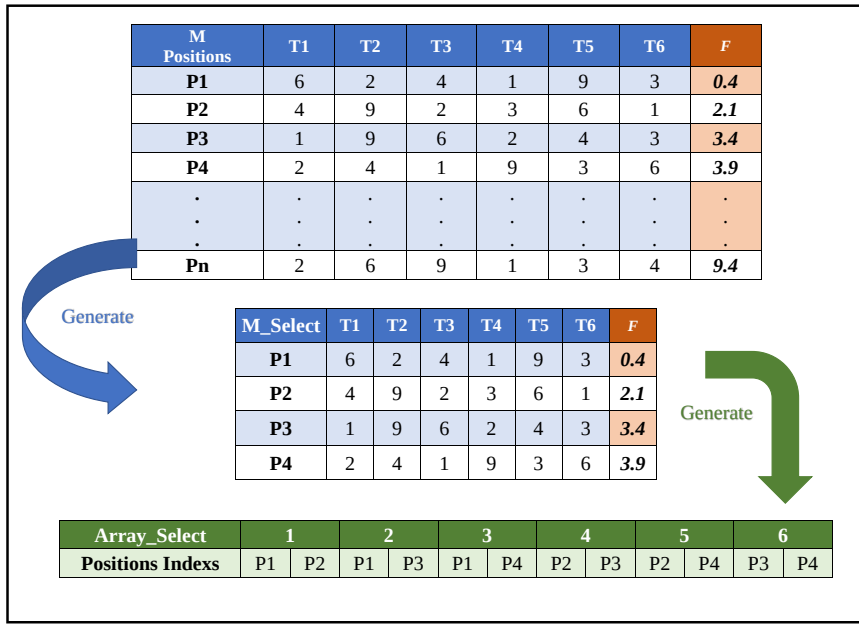
3.3.3 The Crossover Component

From the indices values of the Array_Select vector, this component performs the crossing between two parents. So, it randomly takes a series of genes from one parent and concatenates it with the remained genes from the second parent. At the end of the crossover operator execution, we obtain a new half-population (the first 50 % of the solutions).

The pseudo Algorithm 4 **CrossoverAlgo**, presents the actions of the crossing between two parents, to obtain new two parents.

3.3.4 The Grasshoppers Component

In genetic algorithm, to obtain a new generation, the mutation operator is based on random selection of genes, and the replacement of the latter by others closer to



M Positions	T1	T2	T3	T4	T5	T6	F
P1	6	2	4	1	9	3	***0.4***
P2	4	9	2	3	6	1	***2.1***
P3	1	9	6	2	4	3	***3.4***
P4	2	4	1	9	3	6	***3.9***
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
Pn	2	6	9	1	3	4	***9.4***

M_Select	T1	T2	T3	T4	T5	T6	F
P1	6	2	4	1	9	3	***0.4***
P2	4	9	2	3	6	1	***2.1***
P3	1	9	6	2	4	3	***3.4***
P4	2	4	1	9	3	6	***3.9***

Array_Select	1		2		3		4		5		6	
Positions Indexs	P1	P2	P1	P3	P1	P4	P2	P3	P2	P4	P3	P4

Figure 4. Relationship between M_Select and Array_Select

Algorithm 4: CrossoverAlgo

Input: Parent1, Parent2
Output: newParent1, newParent2
Randomly select part of the chromosome(cutPart);
Load the first cutPart of Parent1 and Parent2 to newParent1 and newParent2;
Load the rest of Parent1 and Parent2 to newParent2 and newParent1;

them. This allows a global optimal solution [9] to be found instead of a local optimal solution, and this is the strength of the genetic algorithm for global optimization.

The disadvantage of this operator is its heaviness [18], especially when the number of tasks increases. To overcome this problem, we replaced this operator by a component based on the grasshopper optimization algorithm, which allows to propose local optima [16]. The latter makes it possible to obtain the second half-population (a second 50 % of the solutions) from the first half-population. The obtained values of the second half-population will be normalized so that they correspond with the values of the CPU speeds of virtual machines. The normalization consists of bringing values of the matrix closer to those of the CPU speeds defined in the VMs vector. For example, an obtained value 6.3623, it will be normalized to 6 according to the list of CPU speeds VMs.

Finally, the two half-populations will be concatenated to form the new population. The advantage of this algorithm is that it offers meaningful exploration and exploitation [16].

In the remainder of this section, we explain the swarming behavior of grasshoppers which is mathematically modeled as follows:

$$P_i = (S_i + G_i + A_i)\,, \tag{11}$$

where P_i indicates the position of the grasshopper, S_i is the social interaction between the grasshoppers, G_i indicates the force of gravity on the grasshopper, and A_i is the advection of the wind. To produce random grasshopper behavior, Equation (11) can be rewritten as:

$$P_i = ((r_1 * S_i) + (r_2 * G_i) + (r_3 * A_i))\,, \tag{12}$$

where r_1, r_2 and r_3 are random numbers in the range $[0, 1]$. The social interaction S_i is defined as follows:

$$S_i = \left(\sum_{j=1, j\neq i}^{N} S(d_{ij})\widehat{d_{ij}} \right), \tag{13}$$

where N denotes the number of grasshoppers, $d_{ij} = |\mathbf{P}_j - \mathbf{P}i|$ defines the Euclidean distance between the i^{th} and the j^{th} grasshopper, and $\widehat{d}ij = \frac{\mathbf{P}_j - \mathbf{P}i}{dij}$ is a unit vector from the i^{th} to the j^{th} grasshopper, and S represents the social forces denoted by the following equation:

$$S(r) = \left(f * e^{\frac{-r}{l}} - e^{-r} \right), \tag{14}$$

where f and l are the attraction intensity and the attraction length scale respectively.

Improvements have been made to formula (12) so that it can be used to solve optimization problems, because grasshoppers quickly reach the comfort zone and the swarm does not converge on the objective [16].

$$P_i^d = c_1 \left(\sum_{j=1, j\neq i}^{N} c_2 \frac{ub_d - lb_d}{2} s(|P_j - P_i|) \frac{P_j - P_i}{d_{ij}} \right) + \widehat{T}_d, \tag{15}$$

where ub_d and lb_d respectively represent the upper and lower bounds in the d^{th} dimension (where d represents the objective number on the fitness function). $\widehat{T}_d$ denotes the best solution found so far in the d-dimensional space.

c_1 and c_2 are considered as a single parameter called c which is expressed in the following equation: [16].

$$c = \left(c_{max} - t \frac{c_{max} - c_{min}}{t_{max}} \right), \tag{16}$$

where c_{max} and c_{min} represent the maximum and minimum values of c, respectively, t is the current iteration, and t_{max} is the maximum number of iterations. The

algorithm coefficients are initialized as follows [16]: $c_{max} = 1$, $c_{min} = 0.00004$, $f = 0.5$ and $l = 1.5$.

The formulas (11), (12), (13), (14) and (15) are translated into pseudo Algorithm 5 **GrasshoppersAlgo**.

Algorithm 5: Grasshopper Algorithm

```
Input: M1[s, n]
Output: M2[x − s, n]
Initialize parameters;
while not last position do
    if i ≠ j then
        M2 ← P_i^d;
    end
end
while not last position do
    M2[] ← (M2[] ≈ CPUVM[]); /* Normalized matrix of positions */
end
```

4 CASE STUDY

In this part, we explain the steps to follow, of our approach, in order to achieve the objective. For this, we assume that we have an environment containing a set of 6 tasks $T = \{T_1, T_2, T_3, T_4, T_5, T_6\}$, and a set of 3 virtual machines $VM = \{VM_1, VM_2, VM_3\}$. The operation of the components of our approach is illustrated through the following steps:

Step 1: In this step, the SGGA prepares Tables 4 and 5 with the necessary information. Thus, Table 4, contains the characteristics of the tasks namely the size (Size) and the worst execution time (WCET) [29]. The Table 5 contains virtual machine information: CPU speed, and Storage capacity.

T	T1	T2	T3	T4	T5	T6
Size (KB)	6	2	7	12	3	1.5
WCET	4	2	4.5	5.3	2.3	1.1

Table 4. Tasks characteristics

VM	VM1	VM2	VM3
CPU (mips)	5	4	6
Storage	7	12	22

Table 5. VMs characteristics

We also assume that the size of the population is equal to 20 ($x = 20$), the maximum time for the execution of a given task is 100 ($t_{max} = 100$).

Once the two tables above are prepared, the SGGA randomly initializes the matrix by M chromosomes $[20, 6]$ as shown in Table 6.

M	T1	T2	T3	T4	T5	T6	F
P1	4	6	5	5	5	6	12.80
P2	5	5	4	6	5	6	8.40
P3	5	4	5	6	6	6	9.14
P4	5	4	6	4	4	4	11.39
P5	6	4	5	6	4	6	4.89
P6	4	6	4	5	6	6	9.69
P7	4	4	5	4	5	6	12.07
P8	6	6	4	5	5	4	7.87
P9	6	6	4	4	4	5	13.05
P10	6	6	4	6	5	6	6.53
P11	5	6	5	4	6	5	10.07
P12	5	4	5	6	4	6	8.74
P13	4	5	6	4	5	6	9.96
P14	6	4	5	4	6	5	7.68
P15	6	6	5	4	5	5	8.20
P16	4	5	6	4	6	6	9.98
P17	5	4	5	5	6	4	14.20
P18	5	5	4	6	6	6	8.94
P19	6	5	6	4	6	6	12.44
P20	4	6	5	6	5	4	7.39

Table 6. The matrix M before sorting with Fitness

Then, it invokes the LoadBalancer component to create the fitness vector. The latter contains the fitness of each $Pi \in [1, \ldots, 20]$. Finally, the matrix M will be sorted according to the values of the fitness vector (ascending order) as show in Table 7.

Thus, the best solution (BS $= (6, 4, 5, 6, 4, 6)$), having as fitness the smallest value, will be at the head of the matrix.

Step 2: In this step, the Selector component is invoked. It allows to create the MSelect matrix.

This latter contains, at the beginning, the 7 % best solutions. In this case study, they are the four best chromosomes sectioned from the position matrix (see Table 8). From the MSelect array, the Selector component creates the ArraySelect vector which contains the chromosomes indices that will participate in the crossover step, as shown in Table 9.

Step 3: Once step 2 is completed, the SGGA, invokes the Crossover component. The latter makes it possible to create the first half-population (matrix M1$[10, 6]$)

M	T1	T2	T3	T4	T5	T6	F
P1	6	4	5	6	4	6	4.89
P2	6	6	4	6	5	6	6.53
P3	4	6	5	6	5	4	7.39
P4	6	4	5	4	6	5	7.68
P5	6	6	4	5	5	4	7.87
P6	6	6	5	4	5	5	8.20
P7	5	5	4	6	5	6	8.40
P8	5	4	5	6	4	6	8.74
P9	5	5	4	6	6	6	8.94
P10	5	4	5	6	6	6	9.14
P11	4	6	4	5	6	6	9.69
P12	4	5	6	4	5	6	9.96
P13	4	5	6	4	6	6	9.98
P14	5	6	5	4	6	5	10.07
P15	5	4	6	4	4	4	11.39
P16	4	4	5	4	5	6	12.07
P17	6	5	6	4	6	6	12.44
P18	4	6	5	5	5	6	12.80
P19	6	6	4	4	4	5	13.05
P20	5	4	5	5	6	4	14.20

Table 7. The matrix M after sorting with Fitness

	T1	T2	T3	T4	T5	T6
P1	6	4	5	6	4	6
P2	6	6	4	6	5	6
P3	4	6	5	6	5	4
P4	6	4	5	4	6	5

Table 8. MSelect for the best chromosomes

as indicated in Table 10. The first half-population is obtained from the MSelect matrix and the MSelect vector resulting from step 2. Thus, the M1 matrix contains the 7 % of the population of the M-Select array, and the rest of the population is obtained by crossing chromosomes (Pi, Pj) of the ArraySelect vector.

Step 4: In this step, the Grasshoppers component uses the matrix M1 to create the second half-population and stores it in M2[10, 6]. For example, according to formula (15), the position P11 of M2 is calculated by taking into consideration

1	2	3	4	5	6
P1, P2	P1, P3	P1, P4	P2, P3	P2, P4	P3, P4

Table 9. ArraySelect for all positions chromosomes

	T1	T2	T3	T4	T5	T6
P1	6	4	5	6	4	6
P2	6	6	4	6	5	6
P3	4	6	5	6	5	4
P4	6	4	5	4	6	5
P5	6	4	4	6	5	6
P6	6	4	5	6	5	4
P7	6	4	5	4	6	5
P8	6	6	5	6	5	4
P9	6	6	4	6	6	5
P10	4	6	5	4	6	5

Table 10. The matrix of the first half population M1

the position P1 with all the positions Pi where $i \in [1, \ldots, 10]$ and the best position.

And so on for positions from P12 to P20. The obtained values from the M2 matrix, presented in Table 11, do not correspond to the values of the CPU speeds of the VMs as indicated in Table 5. The normalization operation makes it possible to bring the values of the M2 matrix closer to values of Table 5, as shown in Table 12.

M2	T1	T2	T3	T4	T5	T6
P11	6.36	3.74	5.10	5.80	3.80	5.79
P12	6.05	4.05	4.49	5.57	3.80	5.79
P13	6.05	4.26	5.10	5.57	4.41	5.79
P14	6.05	4.26	5.51	6.42	3.59	6.22
P15	5.64	4.26	5.10	5.57	3.59	5.79
P16	6.36	3.74	4.49	5.80	4.41	5.79
P17	6.36	4.26	5.10	6.42	3.80	5.79
P18	5.64	3.74	4.49	5.80	3.80	6.22
P19	5.64	3.74	4.59	6.42	3.59	6.40
P20	5.64	3.74	4.49	5.57	3.80	5.79

Table 11. The matrix of the second half population M2

Step 5: In the last step, the SGGA merges the two half-populations in the M matrix. The latter represents a new generation of the solutions for the first iteration ($t = 1$) as show in Table 13.

Then, it invokes the LoadBalancer component for the next iteration, as shown in Table 14, until reaching the maximum number of iterations ($t = 100$) as proposed at the beginning.

At the end of the last iteration ($t = 100$), the optimal solution is at the top of the M2 matrix, and it is indicated by the position P1: BS $= (6, 4, 5, 6, 4, 6)$.

M2	T1	T2	T3	T4	T5	T6
P11	6	4	5	6	4	6
P12	6	4	5	6	4	6
P13	6	4	6	6	4	6
P14	6	4	5	6	4	6
P15	6	4	5	6	4	6
P16	6	4	6	6	4	6
P17	6	4	5	6	4	6
P18	6	4	4	6	4	6
P19	6	4	6	6	4	6
P20	6	4	5	6	4	6

Table 12. The M2 normalized

M2	T1	T2	T3	T4	T5	T6	F
P1	6	4	5	6	4	6	19.89
P2	6	6	4	6	5	6	19.93
P3	4	6	5	6	5	4	28.33
P4	6	4	5	4	6	5	19.97
P5	6	4	4	6	5	6	21.09
P6	6	4	5	6	5	4	21.07
P7	6	4	5	4	6	5	22.01
P8	6	6	5	6	5	4	21.59
P9	6	6	4	6	6	5	20.16
P10	4	6	5	4	6	5	27.34
P11	6	4	5	6	4	6	19.89
P12	6	4	5	6	4	6	19.89
P13	6	4	6	6	4	6	20.22
P14	6	4	5	6	4	6	19.89
P15	6	4	5	6	4	6	19.89
P16	6	4	6	6	4	6	20.22
P17	6	4	5	6	4	6	19.89
P18	6	4	4	6	4	6	21.12
P19	6	4	6	6	4	6	20.22
P20	6	4	5	6	4	6	19.89

Table 13. The new matrix of positions M and vector of Fitness ($t = 1$)

5 EXPERIMENTAL RESULTS AND DISCUSSION

The purpose of this section is to verify the effectiveness of our approach. In order to validate our proposal, we used the CloudSim 3.0.3 simulator [6]. The latter is a framework used to model and simulate the Cloud Computing environment and services. It is developed by the CLOUDS Lab organization and is written entirely in Java.

M2	T1	T2	T3	T4	T5	T6	F
P1	**6**	**4**	**5**	**6**	**4**	**6**	**19.89**
P2	6	4	5	6	4	6	19.89
P3	6	4	5	6	4	6	19.89
P4	6	4	5	6	4	6	19.89
P5	6	4	5	6	4	6	19.89
P6	6	4	5	6	4	6	19.89
P7	6	4	5	6	4	6	19.89
P8	6	6	4	6	5	6	19.93
P9	6	4	5	4	6	5	19.97
P10	6	6	4	6	6	5	20.16
P11	6	4	6	6	4	6	20.22
P12	6	4	6	6	4	6	20.22
P13	6	4	6	6	4	6	20.22
P14	6	4	5	6	5	4	21.07
P15	6	4	4	6	5	6	21.09
P16	6	4	4	6	4	6	21.12
P17	6	6	5	6	5	4	21.59
P18	6	4	5	4	6	5	22.01
P19	4	6	5	4	6	5	27.34
P20	4	6	5	6	5	4	28.33

Table 14. The new matrix of positions M and vector of Fitness ($t = 1$)

The simulation environment is a "Dell inspiron" PC, equipped with an Intel(R) Core (TM) i7-3632QM CPU 2.20 GHz, 6 GB RAM, 1 TB hard drive, and it uses a Windows 10 operating system.

The experiment environment is as follows: we set the number of iterations to 1 000 ($t_{max} = 1\,000$), a single data center containing 30 hots machines, and 50 virtual machines. The overall memory of the host machine is 16.384 MB.

To evaluate the performance of our proposed approach, the experiments below are based on the following metrics:

1. the waiting time of the optimal solution (WTOS),
2. the maximum time required for the execution of a batch of tasks (makespan),
3. the throughput (AVGThroughput) which represents the average number of tasks executed per second and per iteration,
4. the average utilization resource ratio (AVGUR), and
5. the hypervolume indicator (HV).

The experimental results obtained are compared with the works closest to our approach, namely: BGA, HGWO-ABC, GWO and ACO. In the following experiments, we assume that we have a maximum set of 1 000 tasks.

Experiment 1: In this experiment, we will evaluate the waiting time of the optimal solution of our approach. Then compare the results obtained with the four approaches: BGA, HGWO-ABC, GWO and ACO.

In this experiment, we vary the number of iterations from 100 to 1 000 with a step of 100, and we set the number of tasks to 500. Then we observe the waiting time necessary to obtain the best solution. Through Figure 5, we can notice that our approach approximates the optimal solution at the end of iteration 670, while the two approaches BGA and HGWO-ABC stabilize respectively from iteration 780 and 900. The optimal solution of two last approaches namely ACO and GWO is reached during the last iteration (1 000).

From the above, we can see that our approach gives better results in terms of waiting time to reach the optimal solution as well as in terms of the number of iterations. This offers a very considerable economic gain.

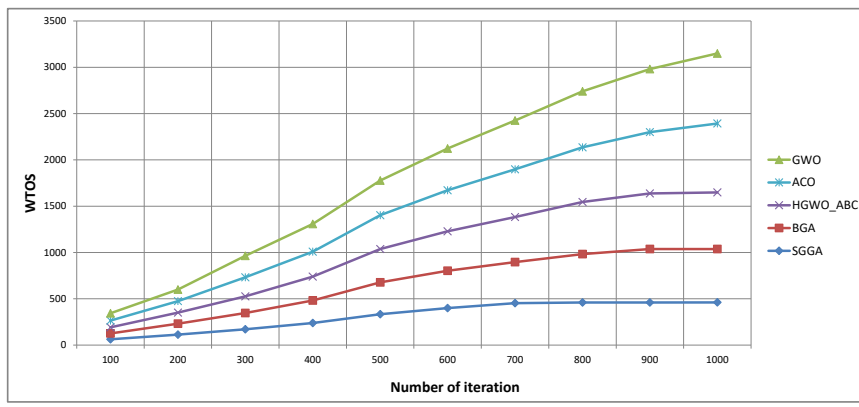


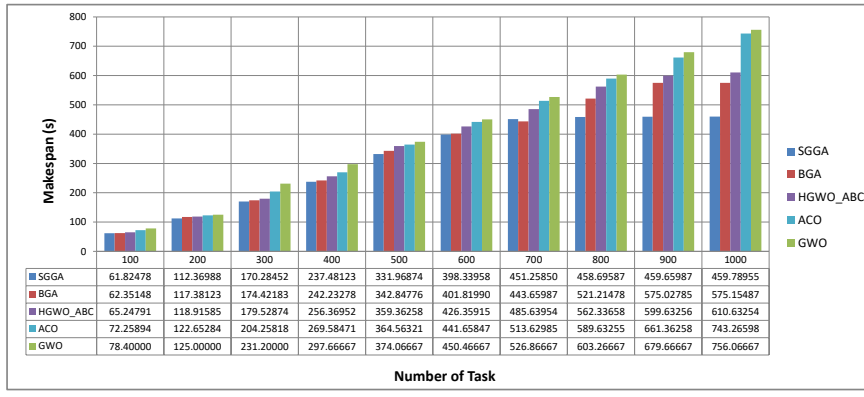
Figure 5. The results of the time to reach the optimal solution

Experiment 2: In this experiment, we will evaluate the makespan of our approach, then the obtained results are compared with those of the other approaches.

In this experimentation, we vary the number of tasks from 100 to 1 000 with a packet of 100, and we set the number of iterations to 1 000. Then we observe the maximum time required for the execution of a batch of tasks.

Figure 6 shows that in the first two packages, all the approaches give almost close values in terms of the makespan. However, from the 7^{th} package the difference between the different approach is clearly visible.

Experiment 3: After evaluating the first metric of the fitness function in the previous experiment, we evaluate its second metric which is the average resource utilization ratio (AVG_{UR}). This metric is between 0 and 1. This metric will be analyzed regarding to the number of tasks, then will compare the obtained results with the other approaches.

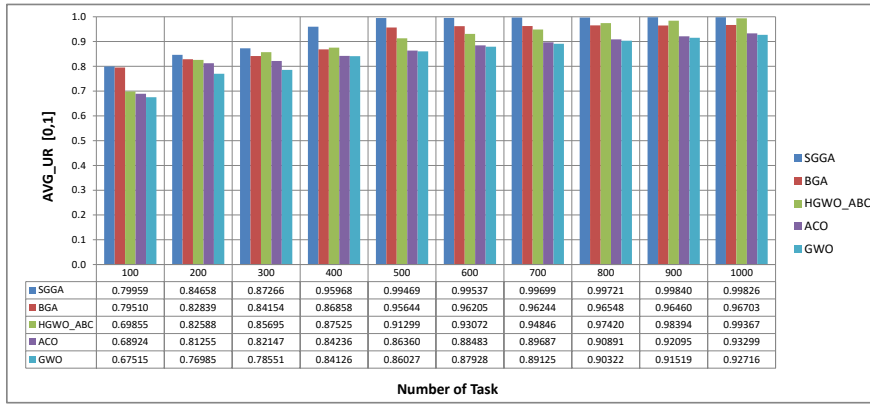


	100	200	300	400	500	600	700	800	900	1000
SGGA	61.82478	112.36988	170.28452	237.48123	331.96874	398.33958	451.25850	458.69587	459.65987	459.78955
BGA	62.35148	117.38123	174.42183	242.23278	342.84776	401.81990	443.65987	521.21478	575.02785	575.15487
HGWO_ABC	65.24791	118.91585	179.52874	256.36952	359.36258	426.35915	485.63954	562.33658	599.63256	610.63254
ACO	72.25894	122.65284	204.25818	269.58471	364.56321	441.65847	513.62985	589.63255	661.36258	743.26598
GWO	78.40000	125.00000	231.20000	297.66667	374.06667	450.46667	526.86667	603.26667	679.66667	756.06667

Figure 6. The results of the makespan compared to the number of tasks

In this experimentation, we vary the number of tasks from 100 to 1 000 with a packet of 100, and we set the number of iterations to 1 000. Then we observe the average rate of resource utilization regarding to the number of tasks. Figure 7 shows that, for the first packet of 100 tasks, our approach and the BGA approach give a higher average resource utilization (AVG_{UR}) than the other approaches. From the 8^{th} packet, the HGOW-ABC approach exceeds that of BGA with a rate of 2.86 %.

Our approach exceeds BGA, HGWO-ABC, ACO and GWO in terms of resource utilization by the order of 3.13 %, 0.46 %, 6.54 % and 7.12 %, respectively. This explains that SGGA gives better performance in terms of average resource utilization compared to other approaches.



	100	200	300	400	500	600	700	800	900	1000
SGGA	0.79959	0.84658	0.87266	0.95968	0.99469	0.99537	0.99699	0.99721	0.99840	0.99826
BGA	0.79510	0.82839	0.84154	0.86858	0.95644	0.96205	0.96244	0.96548	0.96460	0.96703
HGWO_ABC	0.69855	0.82588	0.85695	0.87525	0.91299	0.93072	0.94846	0.97420	0.98394	0.99367
ACO	0.68924	0.81255	0.82147	0.84236	0.86360	0.88483	0.89687	0.90891	0.92095	0.93299
GWO	0.67515	0.76985	0.78551	0.84126	0.86027	0.87928	0.89125	0.90322	0.91519	0.92716

Figure 7. The results of resource utilization rates against the number of tasks

Experiment 4: The fourth metric is evaluated in this experiment. It is the average number of tasks executed in a unit of time (AVG-Throughput). This parameter automatically depends on the makespan. This evaluation is also compared with the same four approaches.

In this experimentation, we set the number of iterations and the number of tasks to 1 000. At the end of the experiment, we calculate the average number of tasks executed per second which is the number of tasks divided by the global execution time. Figure 8 shows that our approach gives a gain of 6.46 %, 16.11 %, 18.81 % and 25.09 % compared to the BGA, HGWO-ABC, ACO and GWO approaches, respectively, which confirms our theoretical assumptions.

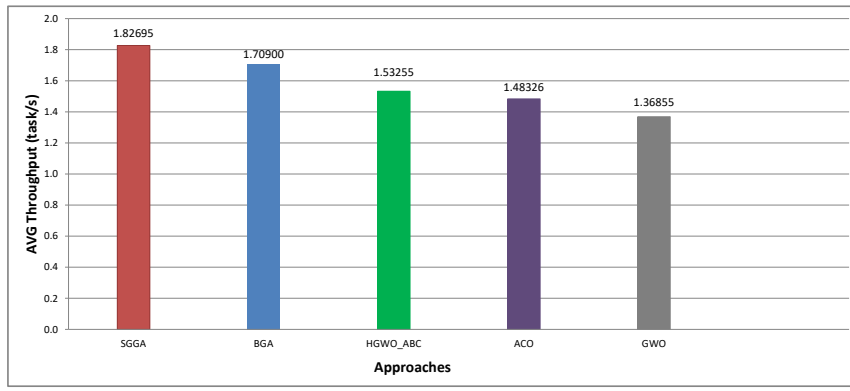


Figure 8. The average throughput comparison

Experiment 5: In this experiment, the SGGA is compared with the other approaches according to the hypervolume indicator, which is the most used metric to compare the performance of scalable multi-objective algorithms [8].

HV is a unary metric that calculates the volume of the area bounded by the set of solutions and a reference point, where a higher value indicates a better result [5].

Figure 9 shows that our approach SGGA and BGA have the best HV indicator with a slight superiority of our approach of around 1.70 %. Compared to other techniques, we find that our approach outperforms other approaches whose gain rate is 13.64 %, 22.87 % and 17.84 % compared to HGWO-ABC, ACO and GWO approaches, respectively.

The HV values confirm that our SGGA approach gives a much more efficient set of solutions than the other approaches.

As a conclusion of the realized experiments, our SGGA approach based on the combination of genetic algorithm and grasshopper optimization algorithm, gives an optimal solution for dynamic load balancing in the CloudIoT.

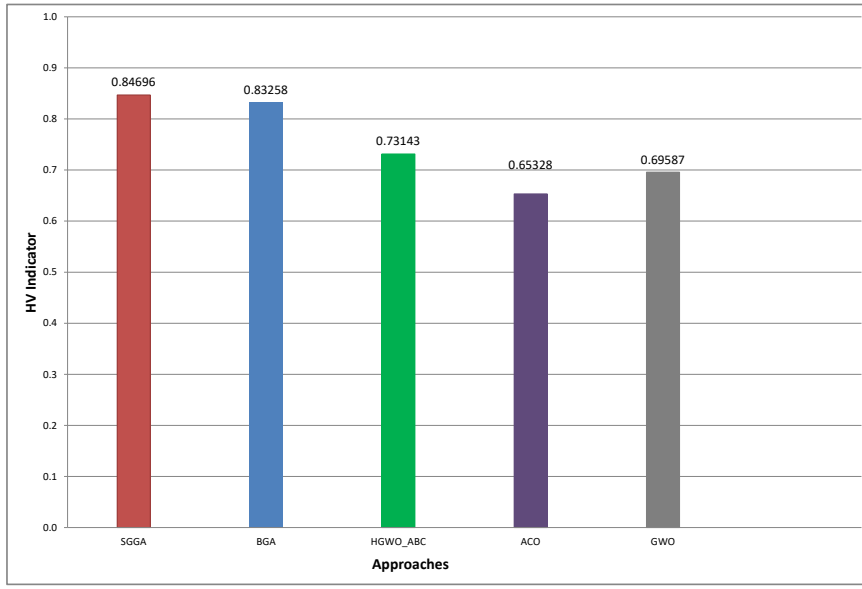


Figure 9. The hypervolume indicator comparison

6 CONCLUSION

The CloudIoT paradigm enables intelligent resource utilization in an equitable manner. In this paradigm, IoTs send their tasks to Cloud Computing for processing or storage, mapping them to the various hardware and software resources represented by virtual machines. When distributing the data to be processed to the virtual machines (VMs), some will be loaded while others will be less loaded or inactive. Load balancing is a mechanism that manages the allocation of VMs to tasks sent by IoTs. It thus allows the optimization of makespan, throughput and the rentable use of resources.

To achieve dynamic load balancing in the CloudIoT, a task scheduler (SGGA) based on the combination between genetic algorithm and grasshopper optimization algorithm has been proposed. The two operators of GA (selection and crossover), are developed to avoid redundancy in the choice of a chromosome several times. Then the mutation operator is replaced by a component based on the grasshopper optimization algorithm. Careful experiments are carried out on CloudSim, to demonstrate that SGGA is more efficient compared to more recent works (BGA, HGOW-ABC, GWO and ACO) in terms of time to reach the optimal solution, makespan, throughput, the average resource utilization ratio and the hypervolume indicator.

In future work, we aim to expand the parameters of load balancing so that SGGA can improve energy consumption and cost, and we implement our approach

on a real system such as a smart city. We aim, also, to evaluate it with the new CEC functions such as Ackley, Rosenbrock, Michalewicz, Dixon and Price function.

REFERENCES

[1] Alguliyev, R. M.—Imamverdiyev, Y. N.—Abdullayeva, F. J.: PSO-Based Load Balancing Method in Cloud Computing. Automatic Control and Computer Sciences, Vol. 53, 2019, No. 1, pp. 45–55, doi: 10.3103/S0146411619010024.

[2] Almezeini, N.—Hafez, A.: Task Scheduling in Cloud Computing Using Lion Optimization Algorithm. International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 8, 2017, No. 11, pp. 77–83, doi: 10.14569/IJACSA.2017.081110.

[3] Arulkumar, V.—Bhalaji, N.: Load Balancing in Cloud Computing Using Water Wave Algorithm. Concurrency and Computation: Practice and Experience, Vol. 34, 2019, No. 8, Art. No. e5492, doi: 10.1002/cpe.5492.

[4] Botta, A.—de Donato, W.—Persico, V.—Pescapé, A.: Integration of Cloud Computing and Internet of Things: A Survey. Future Generation Computer Systems, Vol. 56, 2016, pp. 684–700, doi: 10.1016/j.future.2015.09.021.

[5] Boucetti, R.—Hioual, O.—Hemam, S. M.: An Approach Based on Genetic Algorithms and Neural Networks for QoS-Aware IoT Services Composition. Journal of King Saud University – Computer and Information Sciences, Vol. 34, 2022, No. 8, Part B, pp. 5619–5632, doi: 10.1016/j.jksuci.2022.02.012.

[6] Calheiros, R. N.—Ranjan, R.—Beloglazov, A.—De Rose, C. A. F.—Buyya, R.: CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms. Software: Practice and Experience, Vol. 41, 2011, No. 1, pp. 23–50, doi: 10.1002/spe.995.

[7] Ehsanimoghadam, P.—Effatparvar, M.: Load Balancing Based on Bee Colony Algorithm with Partitioning of Public Clouds. International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 9, 2018, No. 4, pp. 450–455, doi: 10.14569/IJACSA.2018.090462.

[8] Guerreiro, A. P.—Fonseca, C. M.—Paquete, L.: The Hypervolume Indicator: Computational Problems and Algorithms. ACM Computing Surveys, Vol. 54, 2021, No. 6, Art. No. 119, doi: 10.1145/3453474.

[9] Gulbaz, R.—Siddiqui, A. B.—Anjum, N.—Alotaibi, A. A.—Althobaiti, T.—Ramzan, N.: Balancer Genetic Algorithm – A Novel Task Scheduling Optimization Approach in Cloud Computing. Applied Sciences, Vol. 11, 2021, No. 14, Art. No. 6244, doi: 10.3390/app11146244.

[10] Li, G.—Wu, Z.: Ant Colony Optimization Task Scheduling Algorithm for SWIM Based on Load Balancing. Future Internet, Vol. 11, 2019, No. 4, Art. No. 90, doi: 10.3390/fi11040090.

[11] Liu, Y.—Xiao, F.: Intelligent Monitoring System of Residential Environment Based on Cloud Computing and Internet of Things. IEEE Access, Vol. 9, 2021, pp. 58378–58389, doi: 10.1109/ACCESS.2021.3070344.

[12] Muthsamy, G.—Chandran, S. R.: Task Scheduling Using Artificial Bee Foraging Optimization for Load Balancing in Cloud Data Centers. Computer Applications in Engineering Education, Vol. 28, 2020, No. 4, pp. 769–778, doi: 10.1002/cae.22236.

[13] Natesan, G.—Chokkalingam, A.: An Improved Grey Wolf Optimization Algorithm Based Task Scheduling in Cloud Computing Environment. The International Arab Journal of Information Technology, Vol. 17, 2020, No. 1, pp. 73–81, doi: 10.34028/iajit/17/1/9.

[14] Ouhame, S.—Hadi, Y.—Arifullah, A.: A Hybrid Grey Wolf Optimizer and Artificial Bee Colony Algorithm Used for Improvement in Resource Allocation System for Cloud Technology. International Journal of Online and Biomedical Engineering (iJOE), Vol. 16, 2020, No. 14, pp. 4–17, doi: 10.3991/ijoe.v16i14.16623.

[15] Ragmani, A.—Elomri, A.—Abghour, N.—Moussaid, K.—Rida, M.: An Improved Hybrid Fuzzy-Ant Colony Algorithm Applied to Load Balancing in Cloud Computing Environment. Procedia Computer Science, Vol. 151, 2019, pp. 519–526, doi: 10.1016/j.procs.2019.04.070.

[16] Saremi, S.—Mirjalili, S.—Lewis, A.: Grasshopper Optimisation Algorithm: Theory and Application. Advances in Engineering Software, Vol. 105, 2017, pp. 30–47, doi: 10.1016/j.advengsoft.2017.01.004.

[17] Shradha, J.—Jayshree, J.—Chandraprbha, K.: Internet of Things Integrates with Cloud Computing. IRJCS: International Research Journal of Computer Science, Vol. 6, 2019, No. 1, pp. 1–3, doi: 10.26562/IRJCS.2019.JACS10082.

[18] Hachimi, H.: Hybridations d'Algorithmes Métaheuristiques en Optimisation Globale et Leurs Applications. Ph.D. Thesis. INSA de Rouen, France, École Mohammadia d'Ingénieurs, Université Mohammed V Agdal, Rabat, Morocco, 2013. `https://tel.archives-ouvertes.fr/tel-00905604` (in French).

[19] Atlam, H. F.—Alenezi, A.—Alharthi, A.—Walters, R. J.—Wills, G. B.: Integration of Cloud Computing with Internet of Things: Challenges and Open Issues. 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2017, pp. 670–675, doi: 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.105.

[20] Benabbes, S.—Hemam, S. M.: An Approach Based on (Tasks-VMs) Classification and MCDA for Dynamic Load Balancing in the CloudIoT. In: Hatti, M. (Ed.): Smart Energy Empowerment in Smart and Resilient Cities (ICAIRES 2019). Springer, Cham, Lecture Notes in Networks and Systems, Vol. 102, 2019, pp. 387–396, doi: 10.1007/978-3-030-37207-1_41.

[21] Gohil, B. N.—Patel, D. R.: An Improved Grey Wolf Optimizer (iGWO) for Load Balancing in Cloud Computing Environment. In: Hu, T., Wang, F., Li, H., Wang, Q. (Eds.): Algorithms and Architectures for Parallel Processing (ICA3PP 2018). Springer, Cham, Lecture Notes in Computer Science, Vol. 11338, 2018, pp. 3–9, doi: 10.1007/978-3-030-05234-8_1.

[22] Kaur, G.—Sachdeva, R.: Virtual Machine Migration Approach in Cloud Computing Using Genetic Algorithm. In: Goar, V., Kuri, M., Kumar, R., Senjyu, T. (Eds.): Advances in Information Communication Technology and Computing (AICTC 2019).

Springer, Singapore, Lecture Notes in Networks and Systems, Vol. 135, 2019, pp. 195–204, doi: 10.1007/978-981-15-5421-6_20.

[23] Patel, D.—Patra, M. K.—Sahoo, B.: GWO Based Task Allocation for Load Balancing in Containerized Cloud. 2020 International Conference on Inventive Computation Technologies (ICICT 2020), 2020, pp. 655–659, doi: 10.1109/ICICT48043.2020.9112525.

[24] Makasarwala, H. A.—Hazari, P.: Using Genetic Algorithm for Load Balancing in Cloud Computing. 2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI 2016), 2016, pp. 1–6, doi: 10.1109/ECAI.2016.7861166.

[25] Shafahi, Z.—Yari, A.: An Efficient Task Scheduling in Cloud Computing Based on ACO Algorithm. 2021 12th International Conference on Information and Knowledge Technology (IKT 2021), 2021, pp. 72–77, doi: 10.1109/IKT54664.2021.9685674.

[26] Shen, L.—Li, J.—Wu, Y.—Tang, Z.—Wang, Y.: Optimization of Artificial Bee Colony Algorithm Based Load Balancing in Smart Grid Cloud. 2019 IEEE Innovative Smart Grid Technologies – Asia (ISGT-Asia 2019), 2019, pp. 1131–1134, doi: 10.1109/ISGT-Asia.2019.8881232.

[27] Agushaka, J. O.—Ezugwu, A. E.—Abualigah, L.: Dwarf Mongoose Optimization Algorithm. Computer Methods in Applied Mechanics and Engineering, Vol. 391, 2022, Art. No. 114570, doi: 10.1016/j.cma.2022.114570.

[28] Talatahari, S.—Azizi, M.—Tolouei, M.—Talatahari, B.—Sareh, P.: Crystal Structure Algorithm (CryStAl): A Metaheuristic Optimization Method. IEEE Access, Vol. 9, 2021, pp. 71244–71261, doi: 10.1109/ACCESS.2021.3079161.

[29] Fahim, Y.—Ben Lahmar, E.—Labriji, E.—Eddaoui, A.: Une Nouvelle conception d'Equilibrage de Charge dans le Cloud Computing. 4eme Journée sur les Technologies d'Information et de Modélisation (TIM'16), 2016, pp. 1–6. `https://www.researchgate.net/publication/318686170` (in French).

[30] Sadeeq, H. T.—Abdulazeez, A. M.: Giant Trevally Optimizer (GTO): A Novel Metaheuristic Algorithm for Global Optimization and Challenging Engineering Problems. IEEE Access, Vol. 10, 2022, pp. 121615–121640, doi: 10.1109/ACCESS.2022.3223388.

[31] Zhang, Y.—Jin, Z: Group Teaching Optimization Algorithm: A Novel Metaheuristic Method for Solving Global Optimization Problems. Expert Systems with Applications, Vol. 148, 2020, Art. No. 113246, doi: 10.1016/j.eswa.2020.113246.

[32] Cheraghalipour, A.—Hajiaghaei-Keshteli, M.—Paydar, M. M.: Tree Growth Algorithm (TGA): A Novel Approach for Solving Optimization Problems. Engineering Applications of Artificial Intelligence, Vol. 72, 2018, pp. 393–414, doi: 10.1016/j.engappai.2018.04.021.

Sofiane Benabbes is a Ph.D. student on computer science, option: security and web technology. His research in the field of dynamic load balancing in the CloudIoT, IoT and cloud computing at the ICOSI Laboratory at the University of Abbes Laghrour Khenchela, Algeria.

Sofiane Mounine Hemam received his B.Sc. in computer science from the Mentouri University of Constantine, Algeria in 1996, and his M.Sc. in computer science from the Larbi Tebessi University of Tebessa, Algeria in 2005. Currently, he is working as Full Professor at the Department of Mathematics and Computer Science at Abbes Laghrour University of Khenchela, Algeria since 2005. He supervised many Ph.D., Master and License students. Between October 2008 and December 2013, he worked on his Ph.D. in computer science. He has published a number of articles in international journals and conferences. His research interests include database, P2P networks, cloud computing and distributed applications, load balancing, fault tolerance, artificial intelligence.

Computing and Informatics, Vol. 42, 2023, 392–410, doi: 10.31577/cai_2023_2_392

CTRANSNET: CONVOLUTIONAL NEURAL NETWORK COMBINED WITH TRANSFORMER FOR MEDICAL IMAGE SEGMENTATION

Zhixin Zhang, Shuhao Jiang, Xuhua Pan*

Information Engineering Department
Tianjin University of Commerce
Tianjin, 300134, China
e-mail: zhangzhixin010101@163.com, {Jiangshuhao, Panxuha}@tjcu.edu.cn

Abstract. The Transformer has been widely used for many tasks in NLP before, but there is still much room to explore the application of the Transformer to the image domain. In this paper, we propose a simple and efficient hybrid Transformer framework, CTransNet, which combines self-attention and CNN to improve medical image segmentation performance. Capturing long-range dependencies at different scales. To this end, this paper proposes an effective self-attention mechanism incorporating relative position information encoding, which can reduce the time complexity of self-attention from $O(n^2)$ to $O(n)$, and a new self-attention decoder that can recover fine-grained features in encoder from skip connection. This paper aims to address the current dilemma of Transformer applications: i.e., the need to learn induction bias from large amounts of training data. The hybrid layer in CTransNet allows the Transformer to be initialized as a CNN without pre-training. We have evaluated the performance of CTransNet on several medical segmentation datasets. CTransNet shows superior segmentation performance, robustness, and great promise for generalization to other medical image segmentation tasks.

Keywords: Medical image segmentation, deep learning, attention mechanism

* Corresponding author

1 INTRODUCTION

With the development and widespread use of medical imaging equipment, X-rays, CT examinations, magnetic resonance imaging (MRI), and ultrasound scans have become four necessary medical aids used to assist doctors in disease diagnosis, prognosis assessment, and surgery planning. To help doctors make accurate diagnoses, medical image segmentation is required to identify some critical targets in medical images and extract features from them for subsequent lesion diagnosis. In general, there are two main types of image segmentation tasks: semantic segmentation and instance segmentation. Image semantic segmentation is a pixel-level classification task that requires predicting each pixel point of an image. Image instance segmentation requires not only pixel-level classification but also the differentiation of instances based on specific categories. Medical image segmentation is unique in that there are significant differences between each organ or tissue, making instance segmentation of medical images less meaningful. Medical image segmentation usually refers to the semantic segmentation of medical images. Currently, the main medical image segmentation tasks include liver and liver tumour segmentation, brain and brain tumour segmentation, optic disc segmentation, cell segmentation, lung segmentation, and lung nodule segmentation. Many recent medical semantic segmentation approaches have adopted the U-Net framework with a codec structure. However, U-Net using a simple jump-join scheme is still challenging for modelling global multi-scale problems:

1. Not every jump-join setting is valid due to incompatible codec stage feature sets, and even some jump-join can negatively affect segmentation performance;
2. The original U-Net is worse than U-Net without jump-join on some datasets.

CNNs are widely used in computer vision tasks because of their excellent feature extraction capabilities; the encoder-decoder structure built on convolutional operations is currently well-suited for solving location-sensitive tasks such as semantic segmentation. With the help of convolution operations, texture information and local features between neighbouring pixels can be captured; then, by stacking the local features extracted at different levels, the perceptual field can be gradually expanded to obtain higher-level global features. However, this approach has two limitations: firstly, convolution can only extract information between neighbouring pixels and cannot model global associations effectively; secondly, the size and dimensions of the convolution kernel are often fixed and cannot be adjusted according to the input content.

The Transformer has been widely used for many tasks in NLP before [1, 2, 3], but there is still much room to explore the application of the Transformer to the image domain [4, 5, 6]. In this paper, we propose a simple and efficient hybrid Transformer framework, CTransNet, which combines self-attention and CNN to improve medical image segmentation performance and capturing long-range dependencies at different scales. To this end, this paper proposes an effective self-attention mechanism incorporating relative position information encoding, which can reduce the

time complexity of self-attention from $O(n^2)$ to $O(n)$, and a new self-attention decoder that can recover fine-grained features in encoder from skip connection. This paper aims to address the current dilemma of Transformer applications: i.e., the need to learn induction bias from large amounts of training data. The hybrid layer in CTransNet allows the Transformer to be initialized as a CNN without pre-training. We have evaluated the performance of CTransNet on several medical segmentation datasets. CTransNet shows superior segmentation performance, robustness, and great promise for generalization to other medical image segmentation tasks.

Based on the above findings, we propose a new medical image segmentation framework, CTransNet, which leverages channel attention mechanisms. Our approach utilizes a hierarchical cascaded self-attention module (MHSA) to address the inefficiency of multi-headed self-attention in the visual Transformer model caused by high computational and spatial complexity. We propose to split the image into patches, with each patch representing a token to learn feature relationships within a small grid. We group patches into each small grid and compute self-attention in each group, capturing local feature relationships and producing different local feature representations. The smaller grids are then merged into the larger grid, with the previous smaller grid treated as a new token for the next grid's attention computation. CTransNet combines self-attention [7] and convolutional neural network (CNN) techniques to improve segmentation performance, with self-attention modules incorporated into both the encoder and decoder parts to capture long-range dependencies at different scales with minimal overhead. Our approach uses an effective self-attention mechanism that includes relative position information encoding to reduce self-attention's time complexity from $O(n^2)$ to $O(n)$. Additionally, our self-attention decoder can recover fine-grained features in the encoder from skip connection. Experimental results demonstrate that CTransNet outperforms traditional architectures, including transformer and U-Shape frameworks, across different datasets, leading to more accurate and consistent improvements in semantic segmentation.

2 RELATED WORK

2.1 CNN-Based Methods

Early methods for segmenting medical images primarily relied on contour and conventional machine learning techniques [8, 9, 2, 10]. U-Net for medical picture segmentation was proposed in [11] with the introduction of deep CNNs. Numerous Unet-like techniques, like Res-UNet [12], have been developed as a result of the U-shaped structure's ease of use and high performance. U-Net++ [13], Dense-UNet [10], and UNet3+ [14]. Additionally, it has been applied to the segmentation of 3D medical images using methods like 3D-Unet [15] and V-Net [16]. In the field of medical picture segmentation right now, CNN-based techniques have had remarkable success. Because of its potent representation, CNN-based tech-

niques have now had tremendous success in the field of medical image segmentation.

2.2 Vision Transformers Methods

Transformer was initially put forth as a solution for machine translation tasks in [17]. In the field of NLP, techniques based on transformers have excelled in a range of tasks, achieving state-of-the-art performance. Multiple tasks have been completed with state-of-the-art performance [18]. Due to the popularity of Transformer, researchers at [19] developed the ground-breaking Visual Transformer (ViT), which demonstrated a remarkable speed-accuracy trade-off in picture recognition tasks. Because ViT needs pre-training on its own sizable dataset, it is less advantageous than CNN-based techniques. Deit et al. [20] outlines numerous training procedures that make it possible for ViT to be effectively trained on ImageNet in order to overcome the challenges associated with doing so. Recently, some outstanding papers on ViT have been produced [21, 22, 23]. Notably, the Swin Transformer was proposed as the visual backbone given in [23], and it is an effective hierarchical visual transformer. The Swin Transformer, which is based on the shift window mechanism, performs at the cutting edge on a range of vision tasks, including semantic segmentation, object detection, and image classification. In this paper, we try by employing the Swin Transformer block as the basis unit to create a U-shaped encoder-decoder architecture with skip connections for medical picture segmentation, we want to provide a benchmark for the advancement of transformers in the field of medical images. A benchmark comparison can be made using the Transformer's advancement in the realm of medical pictures.

2.3 Transformer to Complement CNNs

In recent years, researchers have attempted to increase network performance by incorporating self-attention mechanisms into CNNs [24, 25, 26, 27]. There are also a number of vision tasks on which Transformer and CNN [28, 29] have been combined, and significant improvements have been achieved. In [30], a U-shaped structure was integrated with skip connections and additive attention gates to analyse medical images. However, this strategy is still CNN-based. Efforts are currently being made to combine CNN with Transformer to challenge CNNs dominance in medical image segmentation. CNNs have advantages for medical picture segmentation [25, 31, 32]. The authors of [25, 33] have developed a potent encoder for the segmentation of two-dimensional medical images. Similar to [25, 31] and [34] use the complementary nature of the Transformer and CNN to enhance the segmentation capabilities of the model. Various combinations of Transformer and CNN are currently employed for the multimodal segmentation of brain tumors [35] and 3D medical picture segmentation [32, 2]. In contrast to the methodologies described above, we investigated the possibility of pure transformers for medical image segmentation applications. We redesigned the multi-headed attention mechanism of

the Transformer and perfectly fused the local information extraction of the convolutional neural network and the global context of the Transformer to make our method more applicable to image segmentation tasks.

3 METHODOLOGY

3.1 Self-Attention

The Transformer model is founded on a multi-head attention module (MHSA, Multi-head self-attention) that enables the model to incorporate attention learned from different subspaces. The output of the multi-heads is concatenated and supplied to the feedforward network (FFN) layer. Given the small sample size of medical datasets, we conducted several experiments and found that a large number of parameter calculations could have an adverse impact on model segmentation performance. Therefore, we determined that the head parameter setting of 6 achieved the best performance for our method. In this study, we applied head = 6 to the input X ($C \times W \times H$) to obtain the mapping Q, K, V after a 1×1 convolution, which is then divided into various heads. The following equation outlines the specific attention calculation:

$$Att(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V. \tag{1}$$

The computed attention is then processed by softmax and called: a contextual aggregation matrix, or similarity matrix, indicating how well each q matches is similar to all the keys; this similarity is then used as a weight and multiplied by the value, so that attention is computed, and is based on a global perceptual field that takes all the inputs into account. This self-attention-based contextual aggregation matrix dynamically adjusts with the input content, allowing for better feature aggregation; however, the dot product operation for $n \times d$ has a time complexity of $O(n^2)$, as n as a sequence length is generally much larger than the dimension d. Where $\sqrt{d}$ denotes approximate normalization, applying the Softmax function to each row of the matrix. Note that we have omitted the computation of multiple headers here for simplicity. The matrix product QK^T is done specifically by first computing the similarity between each pair of tokens. Then, each new token is obtained by derivative acquisition on top of the combination of all tokens. after the MHSA calculation, further residual joins can be added to facilitate optimization. We assume that the height of the input X feature map is H_0 and the width is W_0. We have $N = H_0 \times W_0$. Then, we can divide the feature map into small grids, each of size $G_0 \times G_0$. Therefore, we reconstruct the input feature map to obtain the new X':

$$X \in R^{C\times H_0\times W_0} \to X \in R^{C\times\left(\frac{H_0}{G_0}\times G_0\right)\times\left(\frac{W}{G_0}\times G_0\right)} \to X' \in R^{C\times\left(\frac{H_0}{G_0}\right)\times\left(\frac{W_0}{G_0}\right)\times(G_0\times G_0)}. \tag{2}$$

To simplify the network optimization, we also perform the following transformation for the generated local self-attentive Att:

$$Att_0 \in R^{C\times H_0\times W_0} \rightarrow Att_0 \in R^{C\times\left(\frac{H_0}{G_0}\times G_0\right)\times\left(\frac{W}{G_0}\times G_0\right)} \rightarrow Att_0' \in R^{C\times\left(\frac{H_0}{G_0}\right)\times\left(\frac{W_0}{G_0}\right)\times(G_0\times G_0)}. \tag{3}$$

This computational complexity is significantly reduced because the Att_0 computes each small $G \times G$ network faster. For the i^{th} step, we can consider the smaller network block obtained at the $i-1^{\text{st}}$ step as a new token, which can be achieved simply by downsampling the attentional features:

$$Att_0 = X + Att_0, \tag{4}$$

$$Att_{i-1}' = MaxPool(Att_{i-1}) + AvgPool(Att_{i-1}), \tag{5}$$

where $Att_{i-1}' \in R^{C\times H_i\times W_i}$, $H_i = H_0/(G_0G1\ldots G_{i-1})$, $W_i = W_0/(G_0G1\ldots G_{i-1})$, $MaxPool$ and $AvgPool$ denote maximum pooling and average pooling, respectively. We then similarly divide Att_{i-1}' into a grid of size $G_i \times G_i$ and re-obtain the following equation:

$$\begin{aligned} Att_{i-1}' \in R^{C\times H_i\times W_i} &\rightarrow Att_{i-1}' \in R^{C\times\left(\frac{H_i}{G_i}\times G_i\right)\times\left(\frac{W}{G_i}\times G_i\right)} \\ &\rightarrow Att_{i-1}' \in R^{C\times\left(\frac{H_i}{G_i}\right)\times\left(\frac{W_i}{G_i}\right)\times(G_i^2)}, \end{aligned} \tag{6}$$

$$Q = X_{i-1}'W^q, K = X_{i-1}'W^k, V = X_{i-1}'W^v, \tag{7}$$

finally, we obtain the mathematical representation of A_i as follows:

$$Att_i' \in R^{C\times H_i\times W_i} \rightarrow Att_i' \in R^{C\times\left(\frac{H_i}{G_i}\times G_i\right)\times\left(\frac{W}{G_i}\times G_i\right)} \rightarrow Att_i' \in R^{C\times\left(\frac{H_i}{G_i}\right)\times\left(\frac{W_i}{G_i}\right)\times(G_i^2)}. \tag{8}$$

We connect through the residuals and will keep iterating until it is small enough. Then we stop slicing the grid blocks. the final output of MHSA is:

$$MHSA(X) = (Att_0 + \ldots + Upsample(Att_M))W^p + X, \tag{9}$$

where $UPsample(.)$ denotes upsampling the attentional features to their original size and W^p is the weight matrix of the feature projection. m is the maximum number of iteration steps. In this way, our method can establish global feature dependencies. It is easy to prove that, under the assumption that all G_i are equal, the computational complexity of $MHSA$ is:

$$T_{time}(MHSA) = 3NC^2 + 2NG_0^2C. \tag{10}$$

Thus, we reduce the computational complexity significantly from $O(N^2)$ to $O(N)$, and here G_0 is much smaller than N. Likewise, the space complexity is greatly reduced.

In terms of network time complexity computation, our approach differs from some state-of-the-art not-transformer-based approaches in that we first divide the image into multiple patches, each of which can be considered as a token, and instead of computing attention across all patches, we further group the patches into each small grid and compute self-attention in instead of computing attention across all patches, we further group patches into each small grid and compute self-attention in each grid, thus capturing local feature relationships and producing distinguishable local feature representations. Then, the smaller grids are merged into the larger grid, and the attention in the next grid is recomputed by treating the smaller grid in the previous step as a new token. This process is repeated iteratively to gradually reduce the number of tokens. Throughout the process, our MHSA module progressively computes self-attention in increasing regional network sizes and naturally models the global feature relationships in a hierarchical manner. Since each grid has only a small number of tokens at each step, we can significantly reduce the computational/spatial complexity of the vision Transformer.

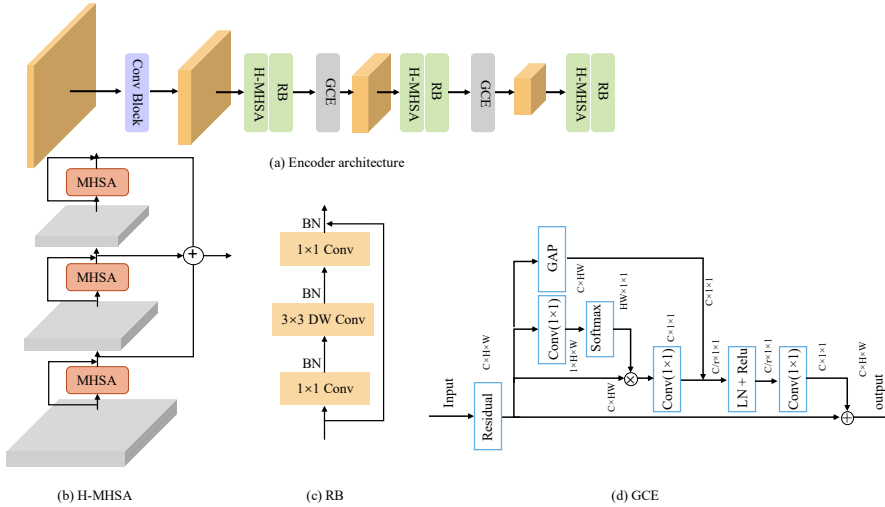


Figure 1. Illustration of the proposed CTransNet. GAP: global average pooling; DW Conv: deepthwise separable convolution; RB: Residual Bottleneck; GCE: global context extraction.

3.2 Network Architecture

Figure 1 illustrates the network structure of CTransNet. The purpose of this study is to combine the benefits of convolution with self-attention so that, on the one hand, convolution can be used to learn inductive bias and avoid pre-training the

Transformer on big datasets, and on the other hand, the Transformer can be used to capture global characteristics. The effective self-attention and relative location coding suggested in this research enable the Transformer to accumulate global contextual data at various scales efficiently. Since miss segmentation frequently happens at the edges of the ROI region, high-resolution contextual information is required for precise segmentation. Instead of only computing self-attention on the CNN-extract feature map, this research employs a transformer at each level of the encoder-decoder to collect long-range dependencies at various scales. However, the raw input was not processed using the Transformer, as employing the Transformer at a superficial level would be of limited benefit and raise the computing cost. One possible explanation for this is that the shallow feature map is more concerned with fine-grained textures than global information. Since the Euclidean distance possesses symmetry, the disease-centric learning strategy, in this case, can be substituted by r. Figure 3 depicts a symmetric metric learning approach centred on drugs and diseases under the explicit treatment relationship. In summary, the disease-centric metric is symmetric with the drug-centric metric, and the objective of symmetric metric learning is to push drugs or diseases that are not associated out of the ball, pull drugs or diseases that are associated or potential associations into the ball, and guarantee that the distance of known drug-disease pairs is smaller than the distance between unknown associations.

3.3 Loss Fuction

Our proposed approach employs the widely-used cross-entropy as the loss function, which serves as a metric to evaluate the degree of agreement between the predicted and ground-truth outputs. In the context of classification training, for a given sample belonging to the K^{th} class, the corresponding output node should have a value of 1 while the remaining nodes have values of 0, forming the target label. By calculating the cross-entropy loss function, we quantify the discrepancy between the predicted output and the target label, and use this difference to update the network parameters through backpropagation. The cross-entropy loss function measures the divergence between the predicted probability distribution and the true probability distribution, where lower cross-entropy implies greater similarity between the two distributions. Formally, assuming p and q as the target and predicted probability distributions, respectively, the cross-entropy loss function is defined as follows:

$$\mathcal{L}_{CE} = -\sum_{x}(p(x)\log q(x) + (1 - p(x)\log(1 - q(x))), \tag{11}$$

where $p(x)$ is the expected output and the probability distribution $q(x)$ is the actual output.

4 EXPERIMENTS AND DISCUSSION

In this section we will focus on some of the details and steps in the experimental process, and the comparative results of some of the most advanced methods and the visualisation of the experimental results on the graphs.

4.1 Datasets and Evaluation

4.1.1 Kvasir-SEG Datasets

Kvasir-SEG is an open-access collection of gastrointestinal polyp pictures and related segmentation masks that were manually annotated by a medical practitioner and subsequently validated by a seasoned gastroenterologist. The Kvasir-SEG dataset includes one thousand polyp pictures and their related ground truth from the Kvasir Dataset v2. The resolution of the photos contained in Kvasir-SEG ranges from 332×487 to $1\,920 \times 1\,072$ pixels. The photos and their respective masks are saved in two distinct folders with the same name. The image files are compressed using the JPEG format, which facilitates online viewing. The publicly available dataset is freely downloadable for research and teaching purposes. The bounding box (coordinate points) for the respective photos is saved in a JSON file. This data collection is intended to further the current best method for polyp identification.

4.1.2 DRIVE Datasets

The DRIVE database was designed to facilitate comparative research on the segmentation of blood vessels in retinal pictures. Retinal vessel segmentation and delineation of morphological attributes of retinal vessels, such as length, width, tortuosity, branching patterns, and angles, are utilized for the diagnosis, screening, treatment, and evaluation of numerous cardiovascular and ophthalmic diseases, such as diabetes, hypertension, atherosclerosis, and choroidal neovascularisation. Automated detection and analysis of blood vessels can help create screening programs for diabetic retinopathy, research the association between vascular tortuosity and hypertensive retinopathy, and aid in computer-assisted laser surgery. For temporal or multimodal image registration and retinal image mosaic synthesis, automatic retinogram generation and branch point extraction have been employed. In addition, it was discovered that the retinal vascular tree is unique to each individual and can be utilized for biometric purposes.

4.1.3 Evaluation

In Equation (12), the accuracy, sensitivity, IoU, and Dice are shown as a criterion group to completely evaluate the experimental outcomes.

$$\begin{cases} Accuracy &= \frac{TP+TN}{TP+FP+FN+TN}, \\ Sensitivity &= \frac{TP}{TP+FN}, \\ IoU &= \frac{TP}{TP+FN+FP}, \\ Dice &= \frac{2\times TP}{(TP+FN)+(TP+FP)}. \end{cases} \tag{12}$$

In this study, the performance of the predictive model is evaluated using several metrics, including true positive (TP), true negative (TN), false positive (FP), and false negative (FN). These metrics represent the number of correctly predicted positive and negative samples, the number of negative samples that were incorrectly predicted as positive, and the number of positive samples that were incorrectly predicted as negative, respectively. Additionally, the sensitivity against specificity is assessed using the Area Under the ROC Curve (AUC) metric. This measure is commonly used to evaluate the performance of binary classification models, where sensitivity is the true positive rate and specificity is the true negative rate.

4.2 Implementation Details

Our CTransNet was implemented using the Pytorch deep learning framework, and we conducted a range of hyperparameter tuning experiments, such as adjusting the learning rate, batch size, weight decay rate, and resize parameters. Both training and testing were carried out on Ubuntu 18.04, using two RTX 2080Ti graphics cards with 12 GB of video memory each. The small batch stochastic gradient descent (SGD) method was employed for training, with a batch size of 8 and a learning rate of 0.0001 on the DRIVE dataset, and a batch size of 8 and a learning rate of 0.001 on the Kvasir-SEG dataset. We compared Adam optimization with SGD and found that SGD typically outperforms Adam, albeit at a slower convergence rate. Despite Adam converging faster, we prioritized performance in both time and accuracy. To validate the effectiveness of our approach, we conducted experiments on multiple datasets, as shown in the figure below, and demonstrated that our approach consistently achieved favourable results.

4.3 Experimental Results

4.3.1 Result on DRIVE Dataset

DRIVE is a dataset that permits the segmentation of retinal blood vessels. It consists of forty color retinal images, twenty of which are used for training and twenty of which are used for evaluation. Originally, the dimensions of the images were 565×584 pixels. A dataset sample of this size is insufficient for training a deep neural network. Consequently, we apply the following strategy to overcome this issue: Beginning with the provided images, random blocks are generated. The remaining photos

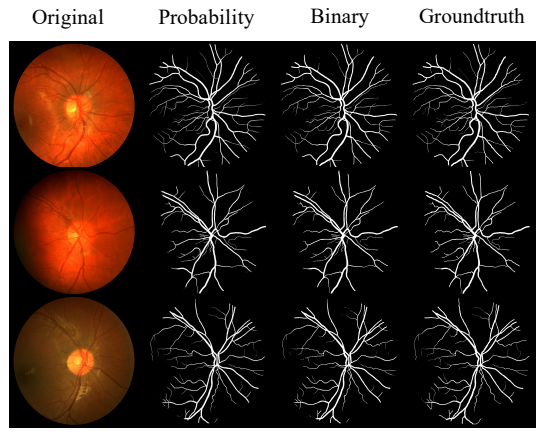


Figure 2. The segmentation results of CTransNet on DRIVE dataset

Methods	**Accuracy**	**Specificity**	**Sensitivity**	**AUC**
Backbone	0.9477	–	0.7781	0.9705
UNet [11]	0.9531	0.9820	0.7537	0.9680
R2-Uet [36]	0.9652	0.8303	0.7792	0.9245
Deep Model [37]	0.9495	0.9768	0.7763	0.9720
RU-net [38]	0.9553	0.9820	0.7726	0.9779
Attention-Unet [39]	0.9629	0.9725	0.7884	0.9740
Unet++ [13]	0.9656	0.9867	0.8234	0.9628
BCD-Unet [40]	0.9560	0.9786	0.8007	0.9789
CENet [41]	0.9545	0.9851	0.8309	0.9779
Fusion Mechanism [42]	0.8247	0.9847	0.8140	0.9782
CTtansNet(Ours)	**0.9660**	**0.9870**	**0.8433**	**0.9785**

Table 1. Performance comparison of the proposed network and the State-of-the-Art methods on DRIVE dataset

were utilized to validate 19,000 segmentation findings using DRIVE. The batch size used as input data for the network was 64×64.

The Figure 2 illustrates some precise of CTransNet and promising segmentation results. In the four columns are listed the original RGB image, the anticipated probability image, the predicted binary image, and the ground truth. Table 1 offers further state-of-the-art research and quantitative findings produced by the proposed network CTransNet on the DRIVE dataset. Our studies were assessed using five unique measures. CTransNet performs brilliantly in terms of accuracy, specificity, sensitivity, and AUC, with respective values of 0.9660, 0.9870, 0.8433, and 0.9785.

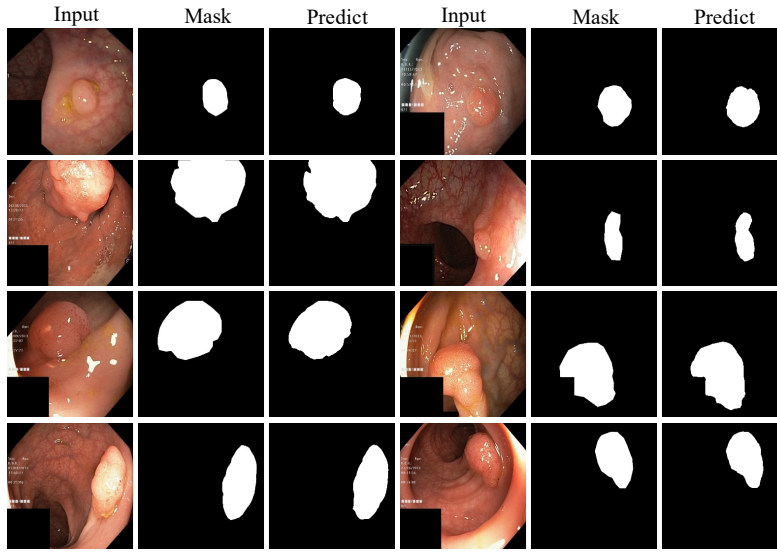


Figure 3. The segmentation results of CTransNet on Kvasir-SEG dataset datasets

4.3.2 Result on Kvasir-SEG Dataset

The results of our CTransNet visualization test on the Kvasir-SEG dataset are shown in Figure 3, from left to right, Input, Mask and Predict. It can be seen that our algorithm has a low error value with Mask. In addition, we also compared it with some classical methods, as shown in Table 2, where our method achieves state-of-the-art performance in several metrics. the values of Precision, Recall, mIOU, and Dice for UNet on the Kvasir-SEG dataset are 92.22, 63.06, 43.43, and 81.80, respectively. resUNet The values of Precision, Recall, mIOU, and Dice on the Kvasir-SEG dataset are 72.92, 50.41, 43.64, and 51.44, respectively. The values of Precision, Recall, mIOU, and Dice on the Kvasir-SEG dataset for MSRF-Net are 96.66, 91.88, 89.14, 92.17. The values of Precision, Recall, mIOU, and Dice for CTransNet on the Kvasir-SEG dataset are 96.75, 90.15, 89.32, and 93.21, respectively. MSRF-Net exceeds our Recall metric by 1.83 %, and their different perceptual fields and multi-scale residual fusion network have significant advantages for the image segmentation task. Experimental results show that our method outperforms the existing state-of-the-art methods in several evaluation metrics, and we analyze some specific reasons why our method efficiently combines visual local attention and contextual information, which is crucial for our semantic segmentation task. Experimental results show that our method outperforms existing state-of-the-art methods in several evaluation metrics, and we analyze some specific reasons why our method effectively combines visual local attention and contextual information, which is crucial for our semantic segmentation task, especially for small dataset tasks where global information is

more important. However, MSRF-Net is currently 1.83 % ahead of us in the Recall metric, which may be an advantage for the Recall metric as MSRF-Net is able to use dual-scale dense fusion to exchange multi-scale features from different perceptual fields.

Methods	**Precision**	**Recall**	**mIoU**	**Dice**
Unet [11]	92.22	63.06	43.34	81.80
ResUNet [43]	72.92	50.41	43.64	51.44
ResUnet-mod [44]	87.13	69.09	42.87	79.09
ResUnet++ [45]	70.64	70.64	79.27	81.33
DeeplabV3+ [46]	94.96	89.84	85.75	89.65
DDANet [47]	86.43	88.80	78.00	85.76
MSRF-Net [48]	96.66	**91.98**	89.14	92.17
CTransNet (Ours)	**96.75**	90.15	**89.32**	**93.21**

Table 2. Performance comparison of the proposed network and the State-of-the-Art methods on n Kvasir-SEG dataset

5 SENSITIVITY ANALYSIS

In this section, in order to verify the effective performance of our method, we conducted a series of ablation experiments aimed at verifying the role of each component on the whole network, and we chose the dataset Kvasir-SEG for this purpose, and the results of the experiments are shown in Table 3. We obtained an mIoU metric of 0.782 on the original CNN-based network, which then increased to 0.792 after embedding the RB module in it. We obtained an mIoU of 0.821 on Kvasir-SEG after using the vision transformer as the backbone, which proves that transformer added as a CNN has a more significant effect than the original CNN. The final mIoU metric of our method on the Kvasir-SEG dataset is 0.893.

Method	mIoU
Encoder + Decoder	0.782
Encoder + RB + Decoder	0.792
Trans + Decoder	0.821
Trans + RB + Decoder	0.834
Trans + GCE + Decoder	**0.842**
Trans + GCE + RB + Decoder	**0.867**
Trans + GCE + MHSA + RB + Decoder (CTransNet)	**0.893**

"Trans" represents vision transformer.

Table 3. mIoU with different setting on Kvasir-SEG dataset

6 CONCLUSION

In this paper, the proposed CTransNet effectively combines CNN with the self-attention mechanism in Transformer to improve the performance of medical image segmentation. This hybrid framework does not require Transformer to be pre-trained on large-scale datasets, where self-attention can effectively capture different levels of long-range information. We believe that this design will help design richer Transformer models that are more suitable for medical image segmentation tasks; in addition, the excellent ability to handle long-range sequences in CTransNet opens up the possibility of migration to other downstream tasks. In the future, we will be working on the task of analysing medical image segmentation from a semi-supervised or weakly supervised perspective. This will give us access to fewer datasets and a more scientific approach to deep learning, and we will also be working on the segmentation of small medical targets.

Acknowledgements

The authors express their gratitude to the reviewers for their valuable feedback and recommendations, which have significantly contributed to improving the quality of the manuscript. The authors would also like to acknowledge the support and assistance provided by their colleagues and students in the laboratory during the course of this research.

REFERENCES

[1] Beltagy, I.—Peters, M. E.—Cohan, A.: Longformer: The Long-Document Transformer. 2020, arXiv: 2004.05150.

[2] Xie, Y.—Zhang, J.—Shen, C.—Xia, Y.: CoTr: Efficiently Bridging CNN and Transformer for 3D Medical Image Segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2021, pp. 171–180, doi: 10.1007/978-3-030-87199-4_16.

[3] Wolf, T.—Debut, L.—Sanh, V.—Chaumond, J.—Delangue, C.—Moi, A.—Cistac, P.—Rault, T.—Louf, R.—Funtowicz, M. et al.: Transformers: State-of-the-Art Natural Language Processing. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 38–45.

[4] Yu, S.—Ma, K.—Bi, Q.—Bian, C.—Ning, M.—He, N.—Li, Y.—Liu, H.—Zheng, Y.: Mil-Vt: Multiple Instance Learning Enhanced Vision Transformer for Fundus Image Classification. International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2021, pp. 45–54, doi: 10.1007/978-3-030-87237-3_5.

[5] Zhang, Z.—Sun, B.—Zhang, W.: Pyramid Medical Transformer for Medical Image Segmentation. 2021, doi: 10.48550/arXiv.2104.14702.

[6] Dai, Y.—Gao, Y.—Liu, F.: Transmed: Transformers Advance Multi-Modal Medical Image Classification. Diagnostics, Vol. 11, 2021, No. 8, Art. No. 1384.

[7] Shaw, P.—Uszkoreit, J.—Vaswani, A.: Self-Attention with Relative Position Representations. 2018, arXiv: 1803.02155.

[8] Tsai, A.—Yezzi, A.—Wells, W.—Tempany, C.—Tucker, D.—Fan, A.—Grimson, W. E.—Willsky, A.: A Shape-Based Approach to the Segmentation of Medical Imagery Using Level Sets. IEEE Transactions on Medical Imaging, Vol. 22, 2003, No. 2, pp. 137–154, doi: 10.1109/TMI.2002.808355.

[9] Held, K.—Kops, E. R.—Krause, B. J.—Wells, W. M.—Kikinis, R.—Muller-Gartner, H. W.: Markov Random Field Segmentation of Brain MR Images. IEEE Transactions on Medical Imaging, Vol. 16, 1997, No. 6, pp. 878–886, doi: 0.1016/j.eswa.2019.05.038.

[10] Li, X.—Chen, H.—Qi, X.—Dou, Q.—Fu, C. W.—Heng, P. A.: H-DenseUNet: Hybrid Densely Connected UNet for Liver and Tumor Segmentation from CT Volumes. IEEE Transactions on Medical Imaging, Vol. 37, 2018, No. 12, pp. 2663–2674, doi: 10.1109/TMI.2018.2845918.

[11] Ronneberger, O.—Fischer, P.—Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015, pp. 234–241, doi: 10.1007/978-3-319-24574-4_28.

[12] Xiao, X.—Lian, S.—Luo, Z.—Li, S.: Weighted Res-Unet for High-Quality Retina Vessel Segmentation. 2018 9th International Conference on Information Technology in Medicine and Education (ITME), IEEE, 2018, pp. 327–331, doi: 10.1109/ITME.2018.00080.

[13] Zhou, Z.—Rahman Siddiquee, M. M.—Tajbakhsh, N.—Liang, J.: Unet++: A Nested U-Net Architecture for Medical Image Segmentation. Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, Springer, 2018, pp. 3–11, doi: 10.1007/978-3-030-00889-5_1.

[14] Huang, H.—Lin, L.—Tong, R.—Hu, H.—Zhang, Q.—Iwamoto, Y.—Han, X.—Chen, Y. W.—Wu, J.: Unet 3+: A Full-Scale Connected Unet for Medical Image Segmentation. ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2020, pp. 1055–1059.

[15] Çiçek, Ö.—Abdulkadir, A.—Lienkamp, S. S.—Brox, T.—Ronneberger, O.: 3d U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2016, pp. 424–432, doi: 10.1007/978-3-319-46723-8_49.

[16] Milletari, F.—Navab, N.—Ahmadi, S. A.: V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. 2016 Fourth International Conference on 3D Vision (3DV), IEEE, 2016, pp. 565–571, doi: 10.1109/3DV.2016.79.

[17] Vaswani, A.—Shazeer, N.—Parmar, N.—Uszkoreit, J.—Jones, L.—Gomez, A. N.—Kaiser, Ł.—Polosukhin, I.: Attention Is All You Need. Advances in Neural Information Processing Systems, Vol. 30, 2017.

[18] Devlin, J.—Chang, M. W.—Lee, K.—Toutanova, K.: Bert: Pre-Training

of Deep Bidirectional Transformers for Language Understanding. 2018, arXiv: 1810.04805.

[19] Dosovitskiy, A.—Beyer, L.—Kolesnikov, A.—Weissenborn, D.—Zhai, X.—Unterthiner, T.—Dehghani, M.—Minderer, M.—Heigold, G.—Gelly, S. et al.: An Image Is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. 2020, arXiv: 2010.11929.

[20] Touvron, H.—Cord, M.—Douze, M.—Massa, F.—Sablayrolles, A.—Jégou, H.: Training Data-Efficient Image Transformers & Distillation Through Attention. International Conference on Machine Learning, PMLR, 2021, pp. 10347–10357.

[21] Wang, W.—Xie, E.—Li, X.—Fan, D. P.—Song, K.—Liang, D.—Lu, T.—Luo, P.—Shao, L.: Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction Without Convolutions. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 568–578, doi: 10.1109/ICCV48922.2021.00061.

[22] Han, K.—Xiao, A.—Wu, E.—Guo, J.—Xu, C.—Wang, Y.: Transformer in Transformer. Advances in Neural Information Processing Systems, Vol. 34, 2021, pp. 15908–15919.

[23] Liu, Z.—Lin, Y.—Cao, Y.—Hu, H.—Wei, Y.—Zhang, Z.—Lin, S.—Guo, B.: Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10012–10022, doi: 10.1109/ICCV48922.2021.00986.

[24] Wang, X.—Girshick, R.—Gupta, A.—He, K.: Non-Local Neural Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7794–7803, doi: 10.1109/CVPR.2018.00813.

[25] Chen, J.—Lu, Y.—Yu, Q.—Luo, X.—Adeli, E.—Wang, Y.—Lu, L.—Yuille, A. L.—Zhou, Y.: Transunet: Transformers Make Strong Encoders for Medical Image Segmentation. 2021, arXiv: 2102.04306.

[26] Li, Z.—Chen, G.—Zhang, T.: A CNN-Transformer Hybrid Approach for Crop Classification Using Multitemporal Multisensor Images. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol. 13, 2020, pp. 847–858, doi: 10.1109/JSTARS.2020.2971763.

[27] Luo, X.—Hu, M.—Song, T.—Wang, G.—Zhang, S.: Semi-Supervised Medical Image Segmentation via Cross Teaching Between CNN and Transformer. 2021, arXiv: 2112.04894.

[28] Weng, W.—Zhang, Y.—Xiong, Z.: Event-Based Video Reconstruction Using Transformer. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 2563–2572, doi: 10.1109/ICCV48922.2021.00256.

[29] Liang, J.—Cao, J.—Sun, G.—Zhang, K.—Van Gool, L.—Timofte, R.: Swinir: Image Restoration Using Swin Transformer. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 1833–1844, doi: 10.1109/ICCVW54120.2021.00210.

[30] Schlemper, J.—Oktay, O.—Schaap, M.—Heinrich, M.—Kainz, B.—Glocker, B.—Rueckert, D.: Attention Gated Networks: Learning to Leverage Salient Regions in Medical Images. Medical Image Analysis, Vol. 53, 2019,

pp. 197–207, doi: 10.1016/j.media.2019.01.012.

[31] Valanarasu, J. M. J.—Oza, P.—Hacihaliloglu, I.—Patel, V. M.: Medical Transformer: Gated Axial-Attention for Medical Image Segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2021, pp. 36–46, doi: 10.1007/978-3-030-87193-2_4.

[32] Hatamizadeh, A.—Tang, Y.—Nath, V.—Yang, D.—Myronenko, A.—Landman, B.—Roth, H. R.—Xu, D.: Unetr: Transformers for 3d Medical Image Segmentation. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 574–584, doi: 10.1109/WACV51458.2022.00181.

[33] Liu, Y.—Hu, J.—Kang, X.—Luo, J.—Fan, S.: Interactformer: Interactive Transformer and CNN for Hyperspectral Image Super-Resolution. IEEE Transactions on Geoscience and Remote Sensing, Vol. 60, 2022, pp. 1–15, doi: 10.1109/TGRS.2022.3183468.

[34] Zhang, Y.—Liu, H.—Hu, Q.: Transfuse: Fusing Transformers and Cnns for Medical Image Segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2021, pp. 14–24, doi: 10.1007/978-3-030-87193-2_2.

[35] Wang, W.—Chen, C.—Ding, M.—Yu, H.—Zha, S.—Li, J.: Transbts: Multimodal Brain Tumor Segmentation Using Transformer. International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2021, pp. 109–119, doi: 10.1007/978-3-030-87193-2_11.

[36] Alom, M. Z.—Hasan, M.—Yakopcic, C.—Taha, T. M.—Asari, V. K.: Recurrent Residual Convolutional Neural Network Based on U-Net (r2u-Net) for Medical Image Segmentation. 2018, arXiv: 1802.06955.

[37] Shin, S. Y.—Lee, S.—Yun, I. D.—Lee, K. M.: Deep Vessel Segmentation by Learning Graphical Connectivity. Medical Image Analysis, Vol. 58, 2019, Art. No. 101556, doi: 10.1016/j.media.2019.101556.

[38] Jaeger, P. F.—Kohl, S. A.—Bickelhaupt, S.—Isensee, F.—Kuder, T. A.—Schlemmer, H.—Maier-Hein, K. H.: Retina U-Net: Embarrassingly Simple Exploitation of Segmentation Supervision for Medical Object Detection. ML4H Workshop, PMLR, 2020, pp. 171–183.

[39] Oktay, O.—Schlemper, J.—Folgoc, L. L.—Lee, M.—Heinrich, M.—Misawa, K. et al.: Attention U-Net: Learning Where to Look for the Pancreas. 2018, arXiv:1804.03999.

[40] Azad, R.—Asadi-Aghbolaghi, M.—Fathy, M.—Escalera, S.: Bi-Directional ConvLSTM U-Net with Densley Connected Convolutions. Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019, pp. 406–415, doi: 10.1109/ICCVW.2019.00052.

[41] Gu, Z.—Cheng, J.—Fu, H.—Zhou, K.—Hao, H.—Zhao, Y. et al.: CE-Net: Context Encoder Network for 2D Medical Image Segmentation. IEEE Trans. Med. Imaging, Vol. 38, 2019, No. 10, pp. 2281–2292, doi: 10.1109/TMI.2019.2903562.

[42] Ding, J.—Zhang, Z.—Tang, J.—Guo, F.: A Multichannel Deep Neural Network for Retina Vessel Segmentation via a Fusion Mechanism. Frontiers in Bioengineering and Biotechnology, 2021, 663 pp., doi: 10.3389/fbioe.2021.697915.

[43] Zhang, Z.—Liu, Q.—Wang, Y.: Road Extraction by Deep Residual U-Net. IEEE Geoscience and Remote Sensing Letters, Vol. 15, 2018, No. 5, pp. 749–753, doi: 10.1109/LGRS.2018.2802944.

[44] Jha, D.—Smedsrud, P. H.—Johansen, D.—de Lange, T.—Johansen, H. D.—Halvorsen, P.—Riegler, M. A.: A Comprehensive Study on Colorectal Polyp Segmentation with ResUNet++, Conditional Random Field and Test-Time Augmentation. IEEE Journal of Biomedical and Health Informatics, Vol. 25, 2021, No. 6, pp. 2029–2040.

[45] Jha, D.—Smedsrud, P. H.—Riegler, M. A.—Johansen, D.—De Lange, T.—Halvorsen, P.—Johansen, H. D.: Resunet++: An Advanced Architecture for Medical Image Segmentation. 2019 IEEE International Symposium on Multimedia (ISM), IEEE, 2019, pp. 225–2255, doi: 10.1109/ISM46123.2019.00049.

[46] Chen, L. C.—Zhu, Y.—Papandreou, G.—Schroff, F.—Adam, H.: Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 801–818.

[47] Tomar, N. K.—Jha, D.—Ali, S.—Johansen, H. D.—Johansen, D.—Riegler, M. A.—Halvorsen, P.: DDANet: Dual Decoder Attention Network for Automatic Polyp Segmentation. International Conference on Pattern Recognition, Springer, 2021, pp. 307–314, doi: 10.1007/978-3-030-68793-9_23.

[48] Srivastava, A.—Jha, D.—Chanda, S.—Pal, U.—Johansen, H. D.—Johansen, D.—Riegler, M. A.—Ali, S.—Halvorsen, P.: Msrf-Net: A Multi-Scale Residual Fusion Network for Biomedical Image Segmentation. IEEE Journal of Biomedical and Health Informatics, Vol. 26, 2021, No. 5, pp. 2252–2263, doi: 10.1109/JBHI.2021.3138024.

Zhixin Zhang graduated from the Tianjin University of Technology. He is currently Lecturer in the College of Information Engineering, Tianjin University of Commerce. His research interests include image recognition and intelligence computing.

Shuhao Jiang received his Master of Engineering in the Tianjin Normal University, Ph.D. in Tianjin University. He is currently Professor in the College of Information Engineering, Tianjin University of Commerce. His research interests include intelligence computing and Natural Language Processing.

Xuhua Pan graduated from the Jilin University. He is currently Professor in the College of Information Engineering, Tianjin University of Commerce. His research interests include intelligence computing and data handling.

Computing and Informatics, Vol. 42, 2023, 411–435, doi: 10.31577/cai_2023_2_411

MGCN: MEDICAL RELATION EXTRACTION BASED ON GCN

Yongpan Wang, Yong Liu*, Jianyi Zhang

College of Information Science and Technology
Qingdao University of Science and Technology
Qingdao, 266061, China
e-mail: liuyong@qust.edu.cn

Abstract. With the progress of society and the improvement of living standards, people pay more and more attention to personal health, and WITMED (Wise Information Technology of med) has occupied an important position. The relationship prediction work in the medical field has high requirements on the interpretability of the method, but the relationship between medical entities is complex, and the existing methods are difficult to meet the requirements. This paper proposes a novel medical information relation extraction method MGCN, which combines contextual information to provide global interpretability for relation prediction of medical entities. The method uses Co-occurrence Graph and Graph Convolutional Network to build up a network of relations between entities, uses the Open-world Assumption to construct potential relations between associated entities, and goes through the Knowledge-aware Attention mechanism to give relation prediction for the entity pair of interest. Experiments were conducted on a public medical dataset CTF, MGCN achieved the score of 0.831, demonstrating its effectiveness in medical relation extraction.

Keywords: Relation extraction, co-occurrence graph, attention mechanism, open-world assumption, graph convolutional network

Mathematics Subject Classification 2010: 68U15

* Corresponding author

1 INTRODUCTION

Relation Extraction (RE) is the key task of information extraction, mainly extracting semantic relations between entities from natural language texts, and the results are usually expressed in the form of a triad (subject, relation, object), i.e. (s, r, o). Relation prediction is complementary to this narrow relation extraction, which can combine natural language text information and existing relation triads to reason about the relation between two named entities of interest. It is a complement to the relational network and is also expressed in the form of relational triads. The relation extraction mentioned in this paper is a broad concept, including narrow relation extraction and relation prediction. This type of work is now widely used in knowledge graphs, diagnostic systems, intelligent question and answer, information retrieval, and other related fields.

So far researchers have done a lot of work on relation extraction [1, 2, 3, 4]. At the beginning, people adopted the method based on template matching, using predefined rules or constraints to achieve relationship extraction, the most representative of which is the FASTUS system [5]. With the advancement of technology, traditional machine learning based relation extraction methods have emerged, such as the feature vector based relation extraction method proposed by Miller et al. [6] and the kernel function based relation extraction method proposed by Zelenko et al. [7]. In recent years, deep learning-based relation extraction methods have been proposed by scholars, which have greatly improved the performance of relation extraction. The most classic is LSTM + CRF [8], which is an end-to-end discriminative method. LSTM utilizes past input features, and CRF utilizes sentence-level annotation information, which can effectively utilize past and future annotations to predict current annotations. However, most of the decisions of such neural network methods are black-box operations performed internally, which are difficult to meet the necessary interpretability in the medical field and cannot be directly applied to medical relation extraction tasks.

This paper proposes a medical information relation extraction method MGCN, which can be used in a wide range of medical texts, such as electronic medical records, test reports, medical papers, etc. MGCN uses the co-occurrence graph for modeling to remove sensitive information, which is beneficial to protect patient privacy. The method also utilizes graph convolutional network, which fully incorporate contextual information from medical texts. The open-world assumption is used for relation construction, and the Knowledge-aware Attention mechanism is used to give the final prediction, which provides a reliable basis for the final result. After experiments on dataset, MGCN achieves the F1 score of 0.831, which proves its effectiveness.

The main contributions of this paper are as follows:

1. Aiming at the rigor of the medical field, a highly interpretable method is proposed, which transforms the traditional black-box network computation into relational reasoning.

2. GCN is introduced to compute the relationship between entities, and the information of graph structure data is further utilized while paying attention to the context text information, and the utilization rate of information is improved.
3. The knowledge-aware attention instance encoder is introduced to supervise the reasoning process of the method and it further improves the performance of the method.

2 RELATED WORK

Relation Extraction (RE): RE is to find out the relation between entities in unstructured or semi-structured data, which is part of information extraction and a key step in building a knowledge graph. The existing mainstream relational extraction techniques are classified into three types: supervised learning methods, semi-supervised learning methods, and unsupervised learning methods. The supervised learning methods treat the relation extraction task as a classification problem, design effective features to learn various classification methods based on the training data, and then use the trained classifiers to predict relations. Such methods include many classical methods, such as the DNN [9] proposed by Daojian Zeng, which for the first time equates the relation extraction problem to a relation classification problem and uses deep convolutional neural networks to solve the relation extraction task. The BLSTM [10] proposed by Shu Zhang uses the classical BiLSTM as the main module of the method, reconsiders the lexical feature, and proves its effectiveness. The problem with this type of method is that it requires a large amount of manual annotation of the training corpus, and the corpus annotation work is usually very time-consuming and labor-intensive. The semi-supervised learning methods mainly use bootstrapping for relation extraction. For the relations to be extracted, several seed instances are first set manually, and then the relation template corresponding to the relation and more instances are iteratively extracted from the data. Some representative systems of this type are DIPRE (Dual Iterative Pattern Relation Expansion) [11] proposed by Brin et al. in 1998, NELL (Never-Ending Language Learner) [12] developed by a team led by Professor Tom Mitchell at CMU in 2010, and so on. The semi-supervised learning method of entity relation extraction can partially solve the problem of insufficient number of annotations, but the problem of low accuracy will remain its main challenge for a long time in the future. The unsupervised learning methods assume that pairs of entities with the same semantic relations have similar contextual information. Therefore, we can use the corresponding contextual information of each entity pair to represent the semantic relation of that entity pair, and cluster the semantic relation of all entity pairs. Rozenfeld et al. in 2007 proposed an unsupervised relation identification and extraction system URIES [13], which uses a schema-based contextual representation instead of the context of entity pairs. Yao et al. [14] proposed an unsupervised relation discovery method based on semantic digestion in 2013.

This method uses topic methods to assign entity pairs and their corresponding relation templates to different semantic categories, and then uses clustering methods to map these semantic categories to semantic relations. The effectiveness of such methods depends heavily on how well constraints and heuristics are constructed, and relationships are not as prescriptive as pre-specified relationship types. In comparison, supervised learning methods can extract more effective features with higher accuracy and recall. Therefore, supervised learning methods have received more and more attention from scholars.

In addition, some interesting methods have emerged in the field of relation extraction in the past two years. Xiang Chen proposed an optimization method KnowPrompt [15] based on knowledge co-optimization for text relation extraction (knowledge retrieval, dialogue, question answering) in few-shot scenarios. By learning template words and answer words, knowledge of entities and relations is injected into the methods and their representation is collaboratively optimized under knowledge constraints. Zexuan Zhong proposed a simple and effective end-to-end relation extraction method PURE [16]. The method learns two independent encoders for entity recognition and relation extraction and proposes a new efficient approximation method that achieves large runtime improvements with a small drop in accuracy. Deming Ye proposes a new span representation method PL-Marker [17] that considers the interrelationships between spans (pairs) by strategically wrapping tokens in the encoder. And a neighborhood-oriented packing strategy is proposed to pack the spans with the same starting token into a training instance as much as possible to better distinguish entity boundaries.

Graph Convolutional Network (GCN): Since CNN [18], deep learning methods have achieved high performance for all types of tensors on Euclidean space. However, in addition to the regular data on Euclidean space, there is a large amount of data in the form of topological graphs on non-Euclidean space. A graph data form is shown in Figure 1, which consists of nodes and edges, and nodes connected by edges are neighbors of each other, the number of neighbors of each node is not specified, and there are corresponding signals(information) on each node. Many domain data are represented in this form, such as traffic networks, molecular structures, joint nodes, etc. Traditional convolutional networks are unable to learn such graph-structured data. Based on the need to deal with this topology, the graph convolution method was created. In 2014, Joan Bruna [19] first proposed two different graph convolution construction methods in spatial domain and spectral domain, which laid the foundation for the development of GCN. But its excessive computational complexity and overly large computational parameters limited practical application. Thomas N. Kipf [20] proposed the algorithmic idea of GCN in 2016, and after publishing a related article in 2017, GCN really started to be applied and developed. Then, Michaël Defferrard [21] proposed a second-generation version of the GCN. He has cleverly designed the convolution kernel formula to reduce the number of parameters, re-

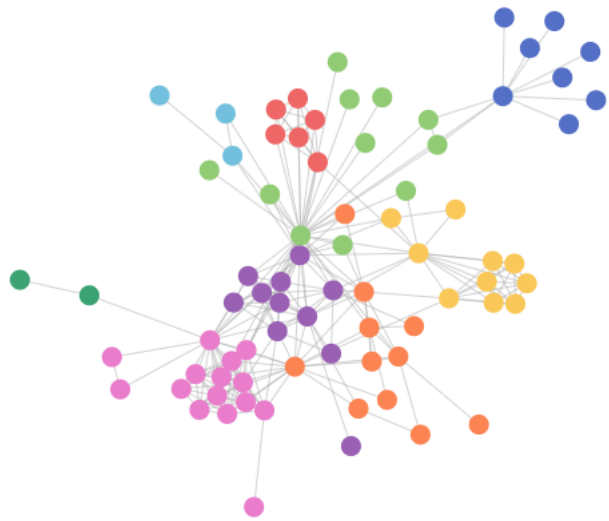


Figure 1. Example of graph data structure

duce the matrix computation, and greatly reduce the computational cost. The rise of GCN has also provided new ideas for solving many natural language processing (NLP) problems. Currently, the way of constructing graph structures by syntactic dependency trees and applying GCN for NLP downstream tasks based on this has been widely used. In addition, there is also work on building graph structures in text through TF-IDF (Term Frequency-Inverse Document Frequency), PMI (Point-wise Mutual Information), sequence relations, lexicon and other information to solve problems using GCN [22, 23, 24, 25]. AGGCN [26] develops a "soft pruning" strategy for the entire dependency tree, transforming the original dependency tree into a fully connected weighted graph. The weights of these graphs are regarded as the correlation strength between nodes and are learned in an end-to-end manner using a self-attention mechanism. At present, there is still a lot of room for GCN to develop in the field of RE.

Open World Assumption (OWA): When making formal descriptions of real-world problems, inevitably the information available is incomplete. For example, we don't know if ibuprofen can cure toothache, but again, this information is indeed useful. A common approach is to use Closed World Assumption (CWA), i.e., if we cannot deduce P or the negation of P in the knowledge base, we add the negation of P to the knowledge base. Another way to deal with incomplete knowledge is to use the Open World Assumption (OWA), which is the opposite of the CWA. OWA is honest about the fact that it does not know the correctness of a proposition that it cannot deduce, with the consequence that the number

of conclusions that can be deduced from the knowledge base is greatly reduced. However, in the semantic Web environment, because of the openness of the Web, the relevant knowledge is likely to be distributed in different places on the Web, so it is inappropriate to use CWA for reasoning on the semantic Web. So, if we want to gather knowledge from different sources in the Semantic Web, we should use OWA. The reasoning in description logic happens to use OWA, so it is indeed suitable as a logical basis for the Semantic Web. In 2016 Ismail Ilkan Ceylan et al. [27] proposed open-world probabilistic databases, as a new probabilistic data method. For unknown facts, this data method assigns any probability value to them from a default probability interval. In 2020 Zhen Wang [28] performed medical entity relation prediction based on corpus-level data and OWA with good results.

3 METHODS

Predicting the relation between entities from a natural language is a very critical task, which can help construct structured knowledge to support a series of downstream tasks such as question answering systems, dialogue systems, inference systems, knowledge graphs, etc. Most of the existing medical information relation extraction methods build deep methods through source texts, and use the attention mechanism to provide local interpretability, which lacks overall global understanding and interpretation. The method MGCN proposed in this paper, for the two medical entities concerned, combines the context information in the medical text and the globality of the medical co-occurrence graph to find their associated entities. Then, the potential relation is constructed using OWA, and finally the final relation prediction is given through the decision module. The overall structure of the method is shown in Figure 2.

3.1 Associated Entity

The first step of the method is to find the associated entities of the entity pair (s, o) of interest. The text information is input into the Bi-LSTM network for word embedding, and then the weights of the relations between nodes are obtained by GCN, as shown in Figure 3. Finally, the top-N nodes that are most closely related to s and o respectively (N is a variable hyperparameter) are found to obtain the set of associated entities.

First, the text information from the library is fed into the Bi-LSTM network to generate word vectors with context, which are then used as the $h^{(0)}$ in the original model. This Bi-LSTM layer is trained jointly with other parts of the network. This has the advantage that the resulting word vector contains both contextual information about word order or disambiguation and provides the correct parse tree on which GCN relies heavily, allowing for more efficient extraction of key information from the sentence. In an L-layer GCN, the input vector of the i^{th} node in the l^{th} layer

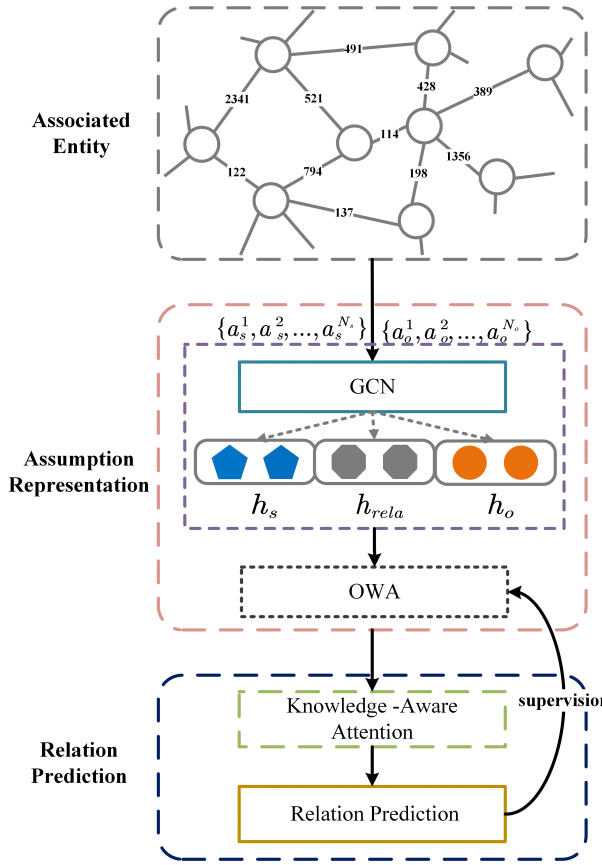


Figure 2. MGCN model overall architecture diagram

is denoted as $h_i^{(l-1)}$ and the output vector is denoted as $h_i^{(l)}$. The graph convolution formula is as follows.

$$h_i^{(l)} = \sigma\left(\sum_{j=1}^{n} A_{ij} W^{(l)} h_j^{(l-1)} + b^{(l)}\right), \tag{1}$$

where $W^{(l)}$ is a linear transformation, $b^{(l)}$ is a bias term, and σ is a nonlinear function (e.g., Relu).

Briefly, during graph convolution, each node collects and aggregates information from neighboring nodes. Convert each dependency tree into an adjacency matrix A and model it uses the graph convolution operation, where $A_{ij} = 1$ if there is dependency edge between nodes i and j. However, because the degree of nodes varies greatly, a direct graph convolution operation in Equation (1) above may lead to very different results for node representation. This may bias the sentence representation

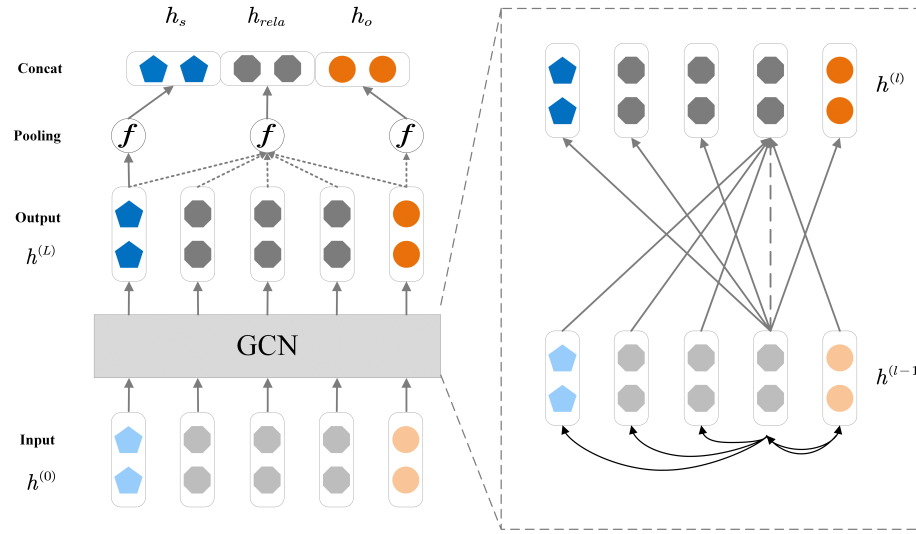


Figure 3. GCN network architecture diagram. The overall architecture is shown on the left, and the detailed calculation method of one-layer graph convolution is shown on the right.

towards nodes with multiple degrees and ignore the information carried by the nodes. In addition, because the nodes in the adjacency matrix have no edge connected to themselves, the information in $h_i^{(l-1)}$ is never passed to $h_i^{(l)}$. To solve these problems, a normalization operation is performed before the data is passed into the nonlinear layer and a self-loop is added to each node in the graph with the following equation:

$$h_i^{(l)} = \sigma\left(\sum_{j=1}^{n} \tilde{A}_{ij} W^{(l)} h_j^{(l-1)} / d_i + b^{(l)}\right), \tag{2}$$

where $\tilde{A} = A + I$, I is the unit matrix of $n \times n$ and $d_i = \sum_{j=1}^{n} \tilde{A}_{ij}$ is the degree of node i in the graph.

This operation is superimposed on the L-layer to obtain a deep GCN network, where $h_1^{(0)}, \ldots, h_n^{(0)}$ is used to represent the input word vectors and $h_1^{(L)}, \ldots, h_n^{(L)}$ to represent the output word vectors. The information transfer between nodes is parallel, and the operations in the network can all be done efficiently by matrix multiplication. After calculating the proximity T of all predicted frames, the confidence of the optimal class of predicted frames is introduced, and the calculation of proximity and confidence is done to describe LT and defined as J. The formula is shown below.

Next, define the model tasks. Let $\mathcal{X} = [x_1, \ldots, x_n]$ denote the sentence, where x_i is the i^{th} word. Identify the subject entity s and the object entity o and correspond them to the two intervals in the sentence: $\mathcal{X}_s = [x_{s_1}, \ldots, x_{s_n}]$ and $\mathcal{X}_o = [x_{o_1}, \ldots, x_{o_n}]$.

Given $\mathcal{X}$, $\mathcal{X}_s$ and $\mathcal{X}_o$, the goal of model is to predict the relation $r \in \mathcal{R}$ ($\mathcal{R}$ is a predefined set of relations) or "no relation" between entities. After applying the L-layer GCN to the word vectors, the implicit representation of each word is obtained, and these representations are directly influenced by their neighbors. In order to use these word representations for relation extraction, the following sentence representations were first obtained (as shown on the left in Figure 3):

$$h_{rela} = f\left(\mathsf{h}^{(L)}\right) = f\left(GCN\left(\mathsf{h}^{(0)}\right)\right), \tag{3}$$

where $\mathsf{h}^{(L)}$ denotes the implicit representation of the overall GCN layer, and $f : \mathbb{R}^{d\times n} \to \mathbb{R}^d$ is the maximum pooling function that maps from the n output vectors to the sentence vectors.

The information close to the entity is usually the core of the relation extraction, and the representation h_s of entity s can be obtained from $\mathsf{h}^{(L)}$, and similarly the representation h_o of entity o can be obtained:

$$h_s = f\left(\mathsf{h}^{(L)}_{s_1:s_2}\right). \tag{4}$$

The final representation for classification is obtained by concatenating the sentence representation and the entity representation and feeding them into a feedforward neural network (FFNN):

$$h_{\text{final}} = \text{FFNN}\left([h_{rela}; h_s; h_o]\right). \tag{5}$$

Then h_{final} is input to the linear layer for Softmax operation to obtain the probability distribution over the relation. The top-N entities are finally selected as associative entities of s/o for subsequent assumption representation.

3.2 Assumption Representation

With associated entities, it is possible to represent assumptions. This method defines the model assumptions as relational interactions between associated entities, as shown in Figure 4. The model can identify (caffeine, may treat, migraine) as a hypothesis, which can help predict that aspirin can treat headache (caffeine and migraine are associated entities of aspirin and headache, respectively). This relational rationale is more specific and easier to understand than the local attention-based explanation strategies widely adopted in NLP. A direct way to obtain this presence relation is to consult the existing medical Knowledge Base (KB), for example (caffeine, may treat, migraine) may be present in SNOMED CT5. This way of obtaining theorems is known as CWA, but in the medical field, the problems of sparsity and incompleteness of KB are serious. Therefore, this method uses OWA to discover more diverse theorems by constructing all potential relations between associated entities.

In OWA, given a pair of entities $e_s, e_o \in \mathcal{V}$, the set of associated entities is defined as $\mathcal{A}(e_s) = \{a_s^i\}_{i=1}^{N_s}$ and $\mathcal{A}(e_o) = \{a_o^j\}_{j=1}^{N_o}$, where N_s, N_o denotes the total number

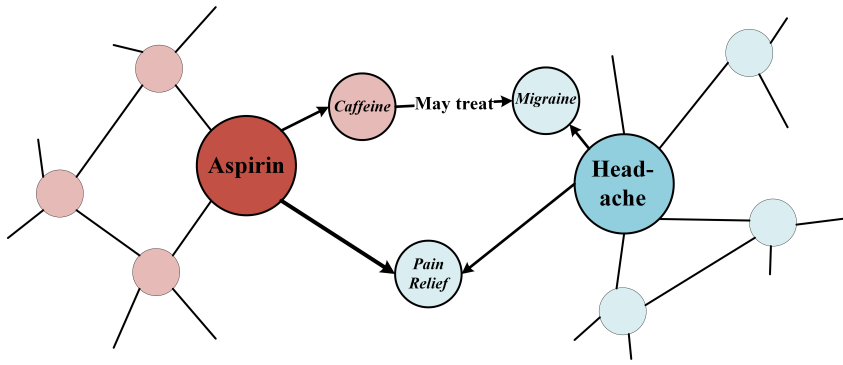


Figure 4. Schematic diagram of potential relation construction

of associated entities. After the previous step, each entity is assigned an embedding vector, which can then be used to measure the probability of maintaining the relation between pairs of associated entities. Given $a_s^i \in \mathcal{A}(e_s), a_o^i \in \mathcal{A}(e_o)$ and relation $r_k \in \mathcal{R}$, define a scoring function to assign a score to the triplet:

$$c_k^{ij} = f\left(a_s^i, r_k, a_o^j\right) = -\left\|h_{a_s^i} + \xi_k - h_{a_o^j}\right\|_1, \tag{6}$$

where $h_{a_s^i}$ and $h_{a_o^j}$ are embedding vectors, the relations are parameterized by a relation matrix $R \in \mathbb{R}^{N_r \times d}$, and ξ_k is a k-level row vector.

Higher scores are obtained when entity pairs and relations are correctly matched. To avoid extremely unreasonable assumptions, the NA relation is defined to represent other irrelevant relations or no relations, and the score is $c_{NA}^{ij} = f\left(a_s^i, \mathrm{NA}, a_o^j\right)$. The OWA principle is expressed by calculating the conditional probability of a relation between a pair of associated entities, the formula is as follows:

$$p\left(r_k | a_s^i, a_o^j\right) = \begin{cases} \frac{\exp(c_k)}{\sum_{s_k \geqslant s_{\mathrm{NA}}} \exp(c_k)}, & c_k > c_{\mathrm{NA}}, \\ 0, & c_k \leqslant c_{\mathrm{NA}}. \end{cases} \tag{7}$$

For each associated entity pair (a_s^i, a_o^j), when the highest value of the relation r is calculated through Equation (7), only the assumption related to r is finally formed. To represent the assumptions, information about all relations for each association pair is integrated into a vector representation, while $p\left(r_k | a_s^i, a_o^j\right)$ is used as the weight of all relations to calculate the assumption representation:

$$\mathsf{a}_{ij} = \rho\left(a_s^i, a_o^j, \mathcal{R}\right) = \sum_{k'=1}^{N_r} p\left(r_{k'} | a_s^i, a_o^j\right) \cdot \xi_{k'}. \tag{8}$$

Combining the entity vector and the relation vector, the final representation of the associated entity's assumption about $\left(a_h^i, a_t^j\right)$ is obtained:

$$e_{ij} = \tanh\left(\left[h_{a_s^i}; h_{a_o^j}; \mathsf{a}_{ij}\right] W_p + b_p\right), \tag{9}$$

where $[\cdot;\cdot]$ denotes the vector connection and $W_p \in \mathbb{R}^{3d\times d_p}$, $b_p \in \mathbb{R}^{d_p}$ are the weight matrix and bias terms of the fully connected network, respectively.

3.3 Relation Prediction

Next comes the relation prediction module which collects all the assumptions, and uses the weighted assumptions information of the target pair to calculate the predicted probability of the relation r. The traditional relation extraction methods only perform relation extraction based on a closed knowledge base, that is, use known factual knowledge for knowledge reasoning. MGCN adopts OWA for relation extraction and uses the calculated probability relation as a given fact to assist the relational reasoning process. Such assumption-based reasoning may lead to certain results that are based entirely on assumptions and are too far from reality. Therefore, we introduce Knowledge-aware Attention to supervise the inference process. The vector v of each instance x of the concerned entity pair is computed using the instance encoder, resulting in a context-based instance representation, which is completely based on known facts. Knowledge-aware Attention will perform attention calculation on assumptions representation and instance representation, to obtain textual relation representation that pays attention to both hypothesis and fact. The introduction of Knowledge-aware Attention will impose certain constraints on assumptions, avoid prediction results that are very inconsistent with facts, and play a supervisory role in the process of assumption reasoning.

This paper designs a new scoring method to measure the confidence of the relation between target entity pairs. Given an entity pair (s, o) and its instance pocket $X_{s,o} = \{x_1, x_2, \ldots, x_m\}$, use the sentence encoder for instance embedding to get $V_{s,o} = \{v_1, v_2, \ldots, v_m\}$. The instance representation thus obtained by the instance encoder is the sentence encoding of each occurrence of the entity of interest in the text. Then use the Knowledge-aware Attention mechanism to get the textual relation representation, which is then used to calculate the relation probability, as shown in Figure 5.

First, the attention weights (similarity or association) between each instance feature vector v_k and assumption representation e_{ij} are calculated:

$$e_k = W_s\left(\tanh\left[v_k : e_{ij}\right]\right) + b_s A_k^i = \frac{\exp\left(c_k\right)}{\sum_{j=1}^{m}\exp\left(c_j\right)}, \tag{10}$$

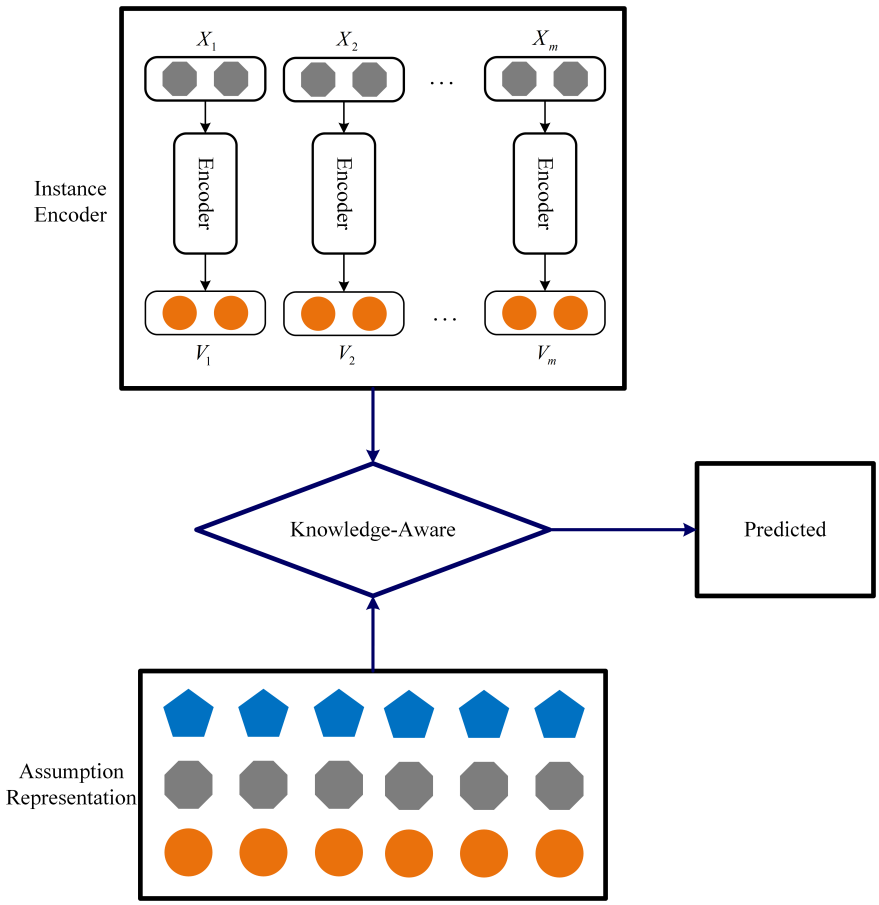


Figure 5. Schematic diagram of the relation prediction stage

where $[x_1 : x_2]$ denotes the vertical connection of x_1 and x_2, W_s is the weight matrix, and b_s is the bias. Then the attention operation is performed on the target entity pair to obtain the corresponding textual relation representation:

$$r^i_{s,o} = ATT\left(e_{ij}, \{v_1, v_2, \ldots, v_m\}\right) g_i = W_g \tanh\left(r_{s,o}\right) \beta_i = \frac{\exp\left(g_i\right)}{\sum_{j=0}^{L-1} \exp\left(g_j\right)}, \tag{11}$$

where W_g is a weight matrix and $r_{s,o}$ is referred to as a query-based function that scores the degree of match between the input textual relation representation and the predicted relation r. The textual relation representation is calculated by:

$$r^i_{s,o} = \beta_i r^i_{s,o}. \tag{12}$$

The textual relation representations of different GCN layers are simply concatenated as the final representation and used to compute the conditional probability $\mathcal{P}(r|s,o)$.

$$r_{s,o} = Concat\left(r_{s,o}^{0}, \ldots, r_{s,o}^{L-1}\right), \tag{13}$$

$$\mathcal{P}(r|s,o) = \frac{\exp(c_r)}{\sum_{\tilde{r}\in R}\exp(\tilde{c}_r)}. \tag{14}$$

At this point, the complete prediction of the relation of the entity pair and their confidence scores are obtained. In addition, to reflect the interpretability of the model, we designed a contribution function O to measure the contribution of all assumptions' representation in the relational inference process:

$$O\left(a_s^i, r_k, a_o^j\right) = \beta_i \times p\left(r_k|a_s^i, a_o^j\right), \tag{15}$$

where β_i is the textual relation representation in Equation (11) and $p(r_k|a_s^i, a_o^j)$ is the relation probability of the associated entity pair in Equation (7).

4 EXPERIMENTS AND RESULTS

This section describes the configuration of the experiments in detail. First, the dataset and evaluation metrics are introduced, and the parameter settings of the experiment and the code running environment are described. Then, MGCN is experimented with a set of comprehensively competitive baseline method on the dataset, and the experimental results are compared and analyzed. Furthermore, to verify the validity of the method rationale, an ablation study was performed.

4.1 Dataset and Evaluation Metrics

Dataset: In order to make full use of the rich resources in the medical field, Finlayson [29] proposed the clinical text frequency (CTF) dataset based on electronic health records (EHR) in 2014. It quantifies pairwise mentions of 3 million terms mapped to 1 million clinical concepts, calculated from the raw text of 20 million clinical records spanning 19 years. The dataset quantifies the correlation between medical entities and eliminates patient privacy information, and its database-level knowledge reserve also provides a reasoning basis for the prediction of medical entity relations. The co-occurrence graph contains 52 804 nodes and 16 197 319 edges, which provides a more concise data form for information researchers in the medical field and greatly promotes the development and utilization of EMR (Electronic Medical Record) resources. After a study of distant supervision of medical texts [30], five medical relations that are more important for clinical decision making were selected. An equal number of negative pairs were extracted by randomly pairing the head and tail entities with the

correct parameter types [31] to help method training. Using the mapping between medical terms and concepts provided by Finlayson et al., relation labels are automatically collected from UMLS (Unified Medical Language System) for training relation prediction. To validate the effectiveness of the method, the dataset was randomly divided into 70 % training, 20 % validation, and 10 % testing in a single experiment.

Med Rela.	Train	Dev	Test
Symptom	14 326	3 001	3 087
May treat	12 924	2 664	2 735
Contraindicates	10 593	2 237	2 197
May prevent	2 113	440	460
Causes	1 389	305	354
Total	41.3 k	8.6 k	8.8 k

Table 1. Dataset statistics

Evaluation Metrics: The evaluation metrics often used for relational extraction tasks are Precision, Recall, and F-Measure. Precision is for the extraction result, which means how many of the samples whose extraction result is the relation R are correct. The TP (True Positive) is the number of correct samples, and the FP (False Positive) is the number of incorrect samples. The formula is:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{16}$$

Recall is for the original sample, which indicates how many samples with relation R are correctly extracted. The correct extraction from the sample set with relation R is recorded as TP, and the wrong extraction is recorded as FN (False Negative). Its calculation formula is:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{17}$$

For relational extraction, the two metrics, accuracy, and recall, are sometimes contradictory and complementary. In this way, they need to be considered comprehensively. The most common method is the F value, also known as F_{score}. Its calculation formula is:

$$F_{score} = \frac{(\beta^2 + 1) \times \text{Pr} \times \text{Re}}{\text{Pr} + \text{Re}}, \tag{18}$$

where Pr denotes the precision and Re denotes the recall score, β is used to balance the weight of precision and recall in the calculation of F value.

In relation extraction tasks, β is generally taken as 1, and the two metrics are considered equally important. Therefore, the calculation formula of F1 value is:

$$F1 = 2 \times \frac{\mathrm{Pr} \times \mathrm{Re}}{\mathrm{Pr} + \mathrm{Re}} \tag{19}$$

The models were all evaluated using F1 as the metrics, and the experimental results were averaged over three replicate experiments.

Rela.	UMLS Relations
Symptom of	disease has finding; disease may have found; has associated finding; has manifestation; associated condition of; defining characteristic of
May treat	May treat
Contraindicates	has contraindicated drug
May prevent	may prevent
Causes	cause of; induces; causative agent of
Total	41.3 k

Table 2. Relations correspond to the mapping UMLS semantic relation

4.2 Implementation Details

Experiments adopt the Adam [32] optimization strategies in our method training and use Binary Cross-Entropy (BCE) [33] loss to improve our network performance. For the method to be used on dataset, the initial parameters are set to learning, the rate is 1e−3, batch size is 128. The number of epochs is 200. All training and testing of the methods are implemented on PyTorch 1.7. Repeat the experiment three times and take the average value as the results.

4.3 Comparison Experiments

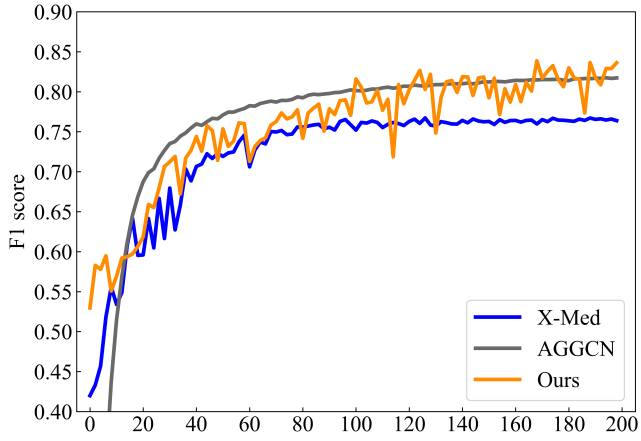
In this section, we compare MGCN with a comprehensive set of relation extraction methods. For predicting the relation between two nodes in a graph, the framework of a neural method usually includes an entity encoder and a relation scoring function. Therefore, different encoders are used to learn entity embeddings and to make a comprehensive comparison. The relation scoring function is unified with RESCAL [34]. The encoders select one word embedding method, Word2vec [35], two graph embedding methods, random-walk based DeepWalk [36], edge-sampling based LINE [37], and one distributional approach REPEL-D [38] for weakly-supervised relation extraction. For graph structure-based relation extraction algorithms, the extended LSTM methods Graph LSTM [39] and bidirectional DAG LSTM [40], attention guided graph convolutional networks (AGGCNs) [26], two newer methods KnowPrompt [15] and PURE [16], and co-occurrence graph-based X-MEDRELA [28] were chosen.

Method	May treat	Symptom	Contrain.	May prevent	Causes	Avg.
Word2vec + RESCAL	0.753	0.764	0.799	0.638	0.650	0.720
DeepWalk + RESCAL	0.701	0.772	0.793	0.623	0.705	0.718
LINE + RESCAL	0.725	0.765	0.800	0.601	0.689	0.716
REPEL-D + RESCAL	0.726	0.769	0.776	0.680	0.707	0.731
Graph LSTM	0.746	0.806	0.743	0.717	0.703	0.743
Bidir DAG LSTM	0.756	0.773	0.769	0.722	0.707	0.745
AGGCN	0.831	0.833	0.801	0.803	0.774	0.828
KnowPrompt	0.836	0.835	0.829	0.814	0.762	0.815
PURE	0.820	0.862	0.833	0.805	0.724	0.809
X-MedRELA	0.805	0.811	0.816	0.676	0.684	0.758
Ours	0.851	0.850	0.832	0.823	0.803	0.831

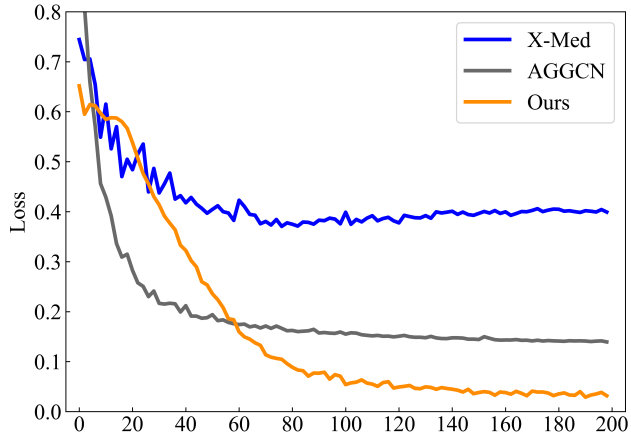
Table 3. Comparison of model predictive performance

Table 3 shows the prediction performance of different methods for F1 scores under each relation prediction task. MGCN obtained a very competitive performance compared to the integrated baseline approach. Specifically, our method achieves substantial improvements in the prediction tasks of "May treat" and "Contraindicates" and performs very competitively in the "Symptom of" and "May prevent" tasks. The poor performance on the "Causes" task may be due to too little training data. This shows that relation extraction based on associations and interactions between entities is effective. Furthermore, compared to those baseline methods that encode graph structures into latent vector representations, MGCN makes full use of co-occurrence graphs, associating context to generate human-understandable rationales. Each stage of our method is interpretable, which can substantially help medical experts.

To demonstrate the effectiveness and convergence of the methods, the F1 and loss curves of X-Med, AGGCN and MGCN at 200 epochs were plotted. As shown in Figure 6 a), the prediction accuracy of all three methods increases rapidly within 40 epochs, and then increases slowly until the best result is achieved at 200 epochs. Among them, the highest F1 value achieved by X-Med is the lowest, and the best result of MGCN is slightly better than that of AGGCN. it can be seen from Figure 6 b) that X-Med and MGCN basically finish converging at 80 epochs, while our method converges approximately at 200 epochs. Comparing the final convergence results, X-Med has the highest loss value at around 0.45; AGGCN also has a poor loss value at around 0.2; while MGCN's loss value has dropped to around 0.04, indicating that the method fits the data well. For deep learning methods, 200 epochs to complete convergence are also a reasonable range. For the performance improvement, the time overhead is worth it. In terms of overall trend, MGCN outperforms X-Med and AGGCN.



a) F1 curve



b) Loss Curve

Figure 6. a) is the FI curve of X-Med vs. MGCN vs. AGGCN and b) is the loss curve of X-Med vs. MGCN vs. AGGCN

4.4 Ablation Study

This section examines the contributions of two main components, namely GCN and Knowledge-aware attention instance encoder. Experiments were conducted on the dataset using the best performing MGCN (w/OWA) method, and the results are shown in Table 4. It can be observed that the introduction of GCN can help the method learn better information aggregation and produce better graph representation, significantly improving the performance of the method. At the same time, adding an attention instance encoder to supervise the inference process of the

method can also further improve the performance. In addition, an ablation study was also carried out for the feedforward layer in the associated entity stage, which confirmed the importance of the feedforward layer in the deep learning method. Without the feedforward layer, the F1 value would drop significantly.

Method	F1
MGCN	0.831
– GCN	0.778
– Attention Instance Encoder (AIE)	0.805
– GCN, AIE	0.758
– Feed-Forward layer(FF)	0.770

Table 4. An ablation study for MGCN model

Performance against Training Data Size. To further test the method performance and explore the effect of different scales of data on the method, a set of experiments were designed. Five training settings (20 %, 40 %, 60 %, 80 %, and 100 % of the training data) were considered in the experiments, and the results are shown in Figure 7. We investigate the performance of MGCN, AGGCN and X-MED on the CTF dataset under different training settings. We investigate the performance of MGCN, AGGCN and X-MED on the CTF dataset under different training settings. At 20 % and 40 % of the training settings, all three methods perform poorly, with MGCN only having a slight advantage, because the performance of deep learning methods relies on large-scale datasets. At 60 % of the training setting, MGCN significantly outperforms AGGCN and X-MED. This means that MGCN has better learning ability when the training data size is average. Under the same amount of training data, MGCN and AGGCN consistently outperform X-MED, and the performance gap becomes more pronounced as the amount of training data increases. When using 100 % training data, the F1 score of MGCN reaches 83.1, which is higher than that of AGGCN of 82.8. These results show that under different scales of data, our method is able to utilize training resources more efficiently and achieve better results.

4.5 Case Studies

This section provides two concrete examples to demonstrate the prediction principles of MGCN to help the reader understand the construction of the method more intuitively.

As shown in Table 5, in order to predict that “cephalosporin” may treat “bacterial infection”, our method will obtain the associated entity “cefuroxime” and “sulbactam” for “cephalosporin”, and the associated entity “viral syndrome’ for “infectious disease” “low grade fever”, “infectious diseases”, and relationships between associated entities. After that, the method will use these five hypothetical principles to predict the relationship between “cephalosporin” and “bacterial infection”,

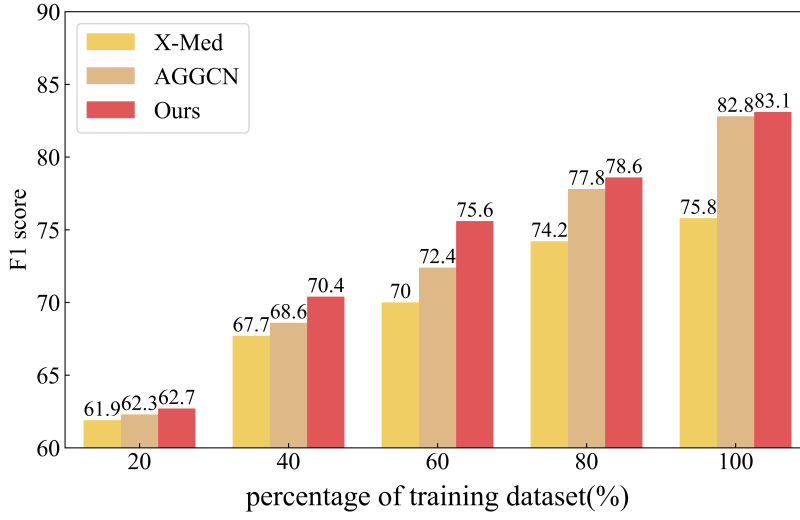


Figure 7. Comparison of MGCN, AGGCN and X-MED against different training data sizes

Subject	Relation	Object
cefuroxime	may treat	viral syndrome
cefuroxime	may treat	low grade fever
cefuroxime	may treat	infectious diseases
cefuroxime	may prevent	low grade fever
sulbactam	may treat	low grade fever
cephalosporins	may treat	bacterial infection

Table 5. Case 1

among which "cefuroxime" may treat "infectious disease" is important to make the final prediction of "possible treatment" theoretical basis. Under the premise of the open-ended hypothesis, doctors may therefore discover new effects of the drug.

Subject	Relation	Object
astepro	may treat	perennial allergic rhinitis
pseudoephedrine	may treat	perennial allergic rhinitis
ciclesonide	may treat	perennial allergic rhinitis
overbite	may treat	perennial allergic rhinitis
diclofenac	may treat	perennial allergic rhinitis
azelastine	may treat	perennial allergic rhinitis

Table 6. Case 2

As shown in Table 6, similarly, the same condition can be treated with different drugs. For the treatment of "perennial allergic rhinitis", the MGCN can give

different medicines (head entities). When one or more of these drugs are known to be effective, doctors can try other drugs to see if they work. Once proven, new drugs can be developed to complement existing drugs. MGCN can make correct predictions based on reasonable principles, providing a theoretical basis to help users understand how the method predictions are performed, and it has an important medical significance.

5 DISCUSSION AND CONCLUSION

Traditional relation extraction methods are all black-box operations, input data sets, output prediction results, and the reasoning process of the method is difficult to visualize. The interpretability of such methods is low and cannot meet the needs of the medical field. For deep learning methods with black-box properties, most of the existing interpretability research use interpretability methods to explain after modeling, such as hidden layer analysis methods, simulation/surrogate methods, sensitivity analysis methods, etc. Different from this kind of research, MGCN itself is an interpretable method. The establishment of the method is based on certain rules, and the decision-making of the method is carried out according to this rule. MGCN performs relation prediction on a given framework, which is set based on the logic of human thinking. For concerned entities, the relevant knowledge is recalled in the first stage, the second stage uses the relevant knowledge to perform relational reasoning, and the third stage gives the prediction result according to the relational reasoning. Under this framework, we can easily understand what each part of the method does and what knowledge is used to make relational predictions. Predicted outcomes for the entity pairs of interest can be traced back to the identified set of associated entities and relational assumptions, as well as the contribution of each assumption to the outcome. In addition to the interpretability of the method itself, each stage of the method can provide a reasoning basis for the results, and has stage interpretability, so MGCN is a method high interpretability. In addition, the method can achieve more accurate and efficient network method tuning and has strong practicability.

This work realizes the relationship extraction of medical information entities and completes the relationship prediction and confidence score of entity pairs in three stages according to different tasks. Unlike existing techniques that rely on multiple different machine learning or deep learning network methods to predict medical entity relationships, MGCN also focuses on the semantic information of the entire corpus while considering OWA. Fully consider the context and spatial structure relationship of the text database, better fit the medical text data characteristics and task characteristics, and finally generate the global optimal prediction theorem. Compared with similar techniques, MGCN is more rational and open in relation prediction, and each stage is interpretable. We believe that MGCN can better assist physicians or practitioners in new medical discoveries and structuring downstream tasks. In the future, this research mainly has the following three exploration direc-

tions. First, consider combining MGCN with state-of-the-art denoising methods to further improve the performance. Secondly, the method is refined on the basis of a small amount of data, so that the method can still achieve better results in the case of scarce data. Finally, MGCN is improved for downstream tasks, enabling it to extract relationships from massive multi-source heterogeneous data and build a medical knowledge graph.

6 DECLARATION OF COMPETING INTEREST

We declare that we have no financial and personal relations with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

Acknowledgements

The research was supported by the Center for Ocean Mega-Science, Chinese Academy of Sciences (KEXUE2019GZ04) and GuangHe Fund of Dawning Information Industry Co., Ltd. (GHFUNd202107021586).

REFERENCES

[1] Aone, C.—Halverson, L.—Hampton, T.—Ramos-Santacruz, M.: SRA: Description of the IE2 System Used for MUC-7. Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29–May 1, 1998, 1998, https://aclanthology.org/M98-1012.

[2] Zheng, S.—Wang, F.—Bao, H.—Hao, Y.—Zhou, P.—Xu, B.: Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2017, pp. 1227–1236, doi: 10.18653/v1/P17-1113.

[3] Eberts, M.—Ulges, A.: Span-Based Joint Entity and Relation Extraction with Transformer Pre-Training. 2019.

[4] Wei, Z.—Su, J.—Wang, Y.—Tian, Y.—Chang, Y.: A Novel Cascade Binary Tagging Framework for Relational Triple Extraction. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020, pp. 1476–1488, doi: 10.18653/v1/2020.acl-main.136.

[5] Appelt, D.—Hobbs, J.—Bear, J.—Israel, D.—Kameyama, M.—Martin, D.—Myers, K.—Tyson, M.: SRI International FASTUS System: MUC-6 Test Results and Analysis. 1995, pp. 237–248.

[6] Miller, S.—Fox, H.—Ramshaw, L.—Weischedel, R.: A Novel Use of Statistical Parsing to Extract Information from Text. 2002.

[7] Zelenko, D.—Aone, C.—Richardella, A.: Kernel Methods for Relation Extraction. Journal of Machine Learning Research, Vol. 3, 2003, pp. 1083–1106, doi: 10.3115/1118693.1118703.

[8] Lample, G.—Ballesteros, M.—Subramanian, S.—Kawakami, K.—Dyer, C.: Neural Architectures for Named Entity Recognition. 2016, pp. 260–270, doi: 10.18653/v1/N16-1030.

[9] Zeng, D.—Liu, K.—Lai, S.—Zhou, G.—Zhao, J.: Relation Classification via Convolutional Deep Neural Network. Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin City University and Association for Computational Linguistics, 2014, pp. 2335–2344, https://aclanthology.org/C14-1220.

[10] Zhang, S.—Zheng, D.—Hu, X.—Yang, M.: Bidirectional Long Short-Term Memory Networks for Relation Classification. Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation, 2015, pp. 73–78, https://aclanthology.org/Y15-1009.

[11] Brin, S.: Extracting Patterns and Relations from the World Wide Web. WebDB, 1998.

[12] Carlson, A.—Betteridge, J.—Kisiel, B.—Settles, B.—Hruschka, E.—Mitchell, T.: Toward an Architecture for Never-Ending Language Learning. Vol. 3, 2010.

[13] Rozenfeld, B.—Feldman, R.: High-Performance Unsupervised Relation Extraction from Large Corpora. 2007, pp. 1032–1037, doi: 10.1109/ICDM.2006.82.

[14] Yao, L.—Riedel, S.—Mccallum, A.: Unsupervised Relation Discovery with Sense Disambiguation. 50th Annual Meeting of the Association for Computational Linguistics, ACL 2012 – Proceedings of the Conference, Vol. 1.

[15] Chen, X.—Zhang, N.—Xie, X.—Deng, S.—Yao, Y.—Tan, C.—Huang, F.—Si, L.—Chen, H.: KnowPrompt: Knowledge-Aware Prompt-Tuning with Synergistic Optimization for Relation Extraction. CoRR, 2021, arXiv: abs/2104.07650.

[16] Zhong, Z.—Chen, D.: A Frustratingly Easy Approach for Entity and Relation Extraction. Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 50–61, doi: 10.18653/v1/2021.naacl-main.5.

[17] Ye, D.—Lin, Y.—Sun, M.: Pack Together: Entity and Relation Extraction with Levitated Marker. CoRR, 2021, arXiv: abs/2109.06067.

[18] Liu, C.—Sun, W.—Chao, W.—Che, W.: Convolution Neural Network for Relation Extraction. Advanced Data Mining and Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 231–242.

[19] Bruna, J.—Zaremba, W.—Szlam, A.—Lecun, Y.: Spectral Networks and Locally Connected Networks on Graphs. 2013.

[20] Kipf, T.—Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks. 2017, arXiv: abs/1609.02907.

[21] Defferrard, M.—Bresson, X.—Vandergheynst, P.: Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. CoRR, 2016, arXiv:

abs/1606.09375.

[22] Liang Yao, C. M.—Luo, Y.: Graph Convolutional Networks for Text Classification. 2019.

[23] Huang, L.—Ma, D.—Li, S.—Zhang, X.—Wang, H.: Text Level Graph Neural Network for Text Classification. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3444–3450, doi: 10.18653/v1/D19-1345.

[24] Gui, T.—Zou, Y.—Zhang, Q.—Peng, M.—Fu, J.—Wei, Z.—Huang, X.: A Lexicon-Based Graph Neural Network for Chinese NER. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, 2019, pp. 1040–1050, doi: 10.18653/v1/D19-1096.

[25] Sui, D.—Chen, Y.—Liu, K.—Zhao, J.—Liu, S.: Leverage Lexical Knowledge for Chinese Named Entity Recognition via Collaborative Graph Network. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, 2019, pp. 3830–3840, doi: 10.18653/v1/D19-1396.

[26] Guo, Z.—Zhang, Y.—Lu, W.: Attention Guided Graph Convolutional Networks for Relation Extraction. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2019, pp. 241–251, doi: 10.18653/v1/P19-1024.

[27] Ceylan, u. u.—Darwiche, A.—Van den Broeck, G.: Open-World Probabilistic Databases. Proceedings of the Fifteenth International Conference on Principles of Knowledge Representation and Reasoning, AAAI Press, 2016.

[28] Wang, Z.—Lee, J.—Lin, S.—Sun, H.: Rationalizing Medical Relation Prediction from Corpus-Level Statistics. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020, pp. 8078–8092, https://aclanthology.org/2020.acl-main.719.

[29] Finlayson, S.—LePendu, P.—Shah, N.: Building the Graph of Medicine from Millions of Clinical Narratives. Scientific Data, Vol. 1, 2014, doi: 10.1038/sdata.2014.32.

[30] Wang, C.—Fan, J.: Medical Relation Extraction with Manifold Models. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (volume 1: Long Papers), Association for Computational Linguistics, 2014, pp. 828–838, doi: 10.3115/v1/P14-1078.

[31] Wang, C.—Cao, L.—Fan, J.: Building Joint Spaces for Relation Extraction. Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, AAAI Press, IJCAI'16, 2016, pp. 2936–2942.

[32] Kingma, D. P.—Ba, J.: Adam: A Method for Stochastic Optimization. 2014.

[33] Boer, P. T.—Kroese, D.—Mannor, S.—Rubinstein, R.: A Tutorial on the Cross-Entropy Method. Annals of Operations Research, Vol. 134, 2005, pp. 19–67,

doi: 10.1007/s10479-005-5724-z.

[34] Nickel, M.—Tresp, V.—Kriegel, H. P.: A Three-Way Model for Collective Learning on Multi-Relational Data. Proceedings of the 28th International Conference on International Conference on Machine Learning, Omnipress, Madison, WI, USA, ICML'11, 2011, pp. 809–816.

[35] Mikolov, T.—Sutskever, I.—Chen, K.—Corrado, G.—Dean, J.: Distributed Representations of Words and Phrases and Their Compositionality. Proceedings of the 26th International Conference on Neural Information Processing Systems – Volume 2, Curran Associates Inc., Red Hook, NY, USA, NIPS'13, 2013, pp. 3111–3119.

[36] Perozzi, B.—Al-Rfou, R.—Skiena, S.: Deepwalk: Online Learning of Social Representations. Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, New York, NY, USA, doi: 10.1145/2623330.2623732.

[37] Tang, J.—Qu, M.—Wang, M.—Zhang, M.—Yan, J.—Mei, Q.: LINE: Large-Scale Information Network Embedding. Proceedings of the 24th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2015, doi: 10.1145/2736277.2741093.

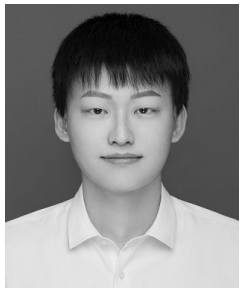
[38] Qu, M.—Ren, X.—Zhang, Y.—Han, J.: Weakly-Supervised Relation Extraction by Pattern-Enhanced Embedding Learning. Proceedings of the 2018 World Wide Web Conference, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2018, doi: 10.1145/3178876.3186024.

[39] Peng, N.—Poon, H.—Quirk, C.—Toutanova, K.—Yih, W. t.: Cross-Sentence N-Ary Relation Extraction with Graph LSTMs. Transactions of the Association for Computational Linguistics, Vol. 5, 2017, pp. 101–115, doi: 10.1162/tacl_a_00049.

[40] Song, L.—Zhang, Y.—Wang, Z.—Gildea, D.: N-Ary Relation Extraction Using Graph-State LSTM. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2018, pp. 2226–2235, doi: 10.18653/v1/D18-1246.

Yongpan Wang is currently pursuing her Master's degree at the School of Information Science and Technology, Qingdao University of Science and Technology. Her research interests include information extraction and knowledge graphs.

Yong Liu graduated from the Ocean University of China in 2011, majoring in computer application technology. She is dedicated to the research of intelligent identification and quantitative analysis of marine organisms, medical knowledge graph and intelligent medical big data. So far, she has obtained one national invention patent, seven software copyrights, and published more than 20 research papers (SCI/EI). She has published two textbooks.



Jianyi Zhang is currently pursuing his Master's degree at the School of Information Science and Technology, Qingdao University of Science and Technology. His research interests include object detection, medical image segmentation, and natural language processing.

Computing and Informatics, Vol. 42, 2023, 436–456, doi: 10.31577/cai_2023_2_436

CORRELATION COEFFICIENT MEASURE OF INTUITIONISTIC FUZZY GRAPHS WITH APPLICATION IN MONEY INVESTING SCHEMES

Naveen Kumar Akula, Sharief Basha Shaik*

Department of Mathematics
School of Advanced Sciences, VIT University
Vellore 632 014
Tamil Nadu, India
e-mail: {naveenkumar.akula, shariefbasha.s}@vit.ac.in

Abstract. Intuitionistic fuzzy graphs are extensions of fuzzy graphs that preserve the dualism characteristics of fuzzy graphs and have a stronger capacity to describe ambiguity in actual decision-making issues than fuzzy graphs. In this research paper, the Laplacian energy and correlation coefficient of intuitionistic fuzzy graphs are computed for finding group decision-making problems that are supported by intuitionistic fuzzy preference relations. We propose a novel method for calculating establishments' comparative position loads by manipulating the undecided corroboration of IFPR and the correlation coefficient of one personality IFPR to the other items. As a result, we comprehend a large number of establishments in the detailed IFPR and devise a correlation coefficient process to investigate the significance of alternatives and the best of the alternatives. Finally, we present a collaborative decision-making technique in a money-investing scheme, and that idea may be devised in disparate beneficial investing schemes.

Keywords: Intuitionistic fuzzy preference relation, intuitionistic fuzzy adjacency matrix, intuitionistic fuzzy laplacian matrix, intuitionistic fuzzy graph, Laplacian energy, correlation coefficient, group decision-making problem

Mathematics Subject Classification 2010: 03E72, 03B52

* Corresponding author

1 INTRODUCTION

FS	Fuzzy sets
FG	Fuzzy graph
IFS	Intuitionistic fuzzy set
IFG	Intuitionistic fuzzy graph
IFPR	Intuitionistic fuzzy preference relation
IFAM	Intuitionistic fuzzy adjacency matrix
IFLM	Intuitionistic fuzzy Laplacian matrix
LE	Laplacian energy
CC	Correlation coefficient
GDMP	Group decision making problem
FMF	Fuzzy membership function
FNMF	Fuzzy non-membership function
MVs	Membership values
NMVs	Non-membership values

Table 1. Nomenclature

Zadeh [1] proposed the notion of fuzzy sets. The range of truth value of the membership relation is the interval $[0, 1]$, which is a property of FS. To address the ambivalence and doubt regarding the membership degree, Atanassov [2] added a new degree, termed as degree of non-membership, to the FS concept in 1986. In a fuzzy set, one excluding the degree of membership functions is known as the indecision degree or non-membership degree of a particular component, and it is thus totally stable. However, in authentic or many instances, there is a degree of ambivalence seen between membership functions, and thus they are independent. Zadeh [3] presented the idea of a fuzzy graph relation, which has been used to analyse cluster patterns. Kaufmann [4], created the concept of FG based on Zadeh's hazy relations. Rosenfeld [5] proposed the notion and construction of the FG. Gutman [6] and Balakrishnan [7] defined graph energy in chemistry, as well as its importance to the total π-electron energy of specific compounds, and identified superior and inferior graph energy limits. In [8] Anjali and Mathew investigated the energy of a FG. The LE of a FG was presented by Sharbaf and Fayazi [9]. The idea of a FG was expanded by Parvathi and Karunambigai [10] to include an IFG. The familiarity with the LE of a FG was applied to the LE of an IFG by Basha and Kartheek in [11]. IFG is one of the most popular and unrivalled extensions of IFS perception. Recently, Falehi [12, 13, 14] has successfully performed IFPRs and their executions using a variety of methodologies. Many novel notions about extended architectures of fuzzy graphs were proposed by Akram et al. [15, 16, 17, 18, 19, 20, 21], and their related implications in decision-making. Also, to choose the optimum alliance partner, Ramesh et al. [22] used a GDM procedure that connected the TOPSIS method with IFG.

In an intuitionistic ambiguous scenario, focusing on the variance and covariance of the IFS, Xuan [23] devised a method for determining the correlation coefficient, the value of which is in $[-1, 1]$. Ye [24] proposed a technique in GDMP based on weighted correlation coefficients using LE is presented for particular situations when the knowledge about criterion weights for alternatives is totally unknown. Also, several statistical methods have been executed by Akula and Sharief Basha [25], Zeng and Li [26], Mitchell [27], Huang ad Guo [28], Szmidt and Kacprzyk [29]. Garg and Rani [30], Khaleie and Fasanghari [31], etc. offered several statistical methods for handling decision-making circumstances by using intuitionistic fuzzy sets to represent the quality of the substitutes and fuzzy values to express the weight of each criterion.

According to intuitionistic fuzzy set research, it is crucial to consider this extension concept. It motivates us to think about IFGs and their applications. In this paper, we provide a strategy for solving GDM issues when the weights (loads) of the criteria are completely unknown and the alternatives are solely determined by the IFG. To address ambiguous information criteria, we use the LE measure to calculate the relative weights based on each decision matrix. To satisfy the total weight vector requirement, we combine each LE weight that was received. The correlation coefficient metric is used to evaluate IFG alternatives, and the best ones are then chosen by calculating the correlation degree for each ranking of the alternatives.

The remainder of this article is structured as follows: The essential principles, covariance, and correlation coefficient measures of IFG are presented in Section 2. Group decision-making is presented in Section 3, utilising IFG's Laplacian energy and correlation coefficient technique. The appropriate application is found in Section 4. Ultimately, the conclusion of the article is presented in Section 5.

2 PRELIMINARIES

Definition 1. An IFG $G_i = (V, E, \mu, \nu)$ is defined as a FG with the nodes set V and the paths set E, where μ is a FMF specified on $V \times V$ and ν is a FNMF, then we specify $\mu(v_i, v_j)$ by μ_{ij} and $\nu(v_i, v_j)$ by ν_{ij} so as that

- $0 \leq \mu_{ij} + \nu_{ij} \leq 1$,
- $0 \leq \mu_{ij}, \nu_{ij}, \pi_{ij} \leq 1$,

where $\pi_{ij} = 1 - (\mu_{ij} + \nu_{ij})$.

Definition 2. An IFAM is well-defined for an IFG $G = (V, E, \mu, \nu)$ by $A(G_i) = [a_{ij}]$, where $a_{ij} = (\mu_{ij}, \nu_{ij})$. It is worth noting that μ_{ij} denotes the strength of the membership bond between v_i and v_j and ν_{ij} denotes the strength of the non-membership bond among both v_i and v_j.

Definition 3. An IFAM can be represented by two matrices, one carrying MVs as well as the other carrying NMVs. So that we represent this matrix as

$$A(G_i) = [(A_\mu(G_i)), (A_\nu(G_i))],$$

where $A_\mu(G_i)$ is the intuitionistic fuzzy membership matrix and $A_\nu(G_i)$ is the intuitionistic fuzzy non-membership matrix.

Definition 4. The Eigen roots of an IFAM are described as (Y, Z), where Y represents the set of latent roots of $A_\mu(G_i)$ and Z represents the set of latent roots of $A_\nu(G_i)$.

Definition 5. Permit $A(G_i)$ as an IFAM and $D(G_i)$ specified by $[d_{ij}]$ as the degree matrix of an IFG. Then IFLM of IFG is defined as

$$L(G_i) = D(G_i) - A(G_i).$$

An IFG's Laplacian matrix can be represented as two matrices, one with MV elements and the other with NMV elements i.e.

$$L(G_i) = [(L(\mu_{ij})), (L(\nu_{ij}))].$$

Definition 6. Consider an IFG $G_i = (V, E, \mu, \nu)$ and λ_i, θ_i are the latent roots of Intuitionistic fuzzy adjacency matrix $A(G_i)$. Then the LE of IFG is described as follows:

$$LE(G_i) = [LE(A_\mu(G_i)), LE(A_\nu(G_i))],$$

where $A_\mu(G_i)$ and $A_\nu(G_i)$ are the membership matrix and non-membership matrix of $A(G_i)$ of an IFG, and λ_i, θ_i are the latent roots of $A_\mu(G_i)$ and $A_\nu(G_i)$. Also, $LE(A_\mu(G_i))$ and $LE(A_\nu(G_i))$ gives the Laplacian energies of membership matrix $A_\mu(G_i)$ and non-membership matrix $A_\nu(G_i)$ of IFG. The LE of $(A_\mu(G_i))$ and $(A_\nu(G_i))$ of an IFG is given by the euations:

$$LE(A_\mu(G_i)) = \sum_{i=1}^{n} \left| \lambda_i - \frac{2\sum_{1\le i\le j\le n} \mu(v_i, v_j)}{n} \right|,$$

$$LE(A_\nu(G_i)) = \sum_{i=1}^{n} \left| \theta_i - \frac{2\sum_{1\le i\le j\le n} \nu(v_i, v_j)}{n} \right|.$$

Definition 7. [Correlation coefficient of IFGs] The Intuitionistic energies of two Intuitionistic Fuzzy Graphs G_1 and G_2 are described as

$$E_{IFG}(G_1) = \sum_{i=1}^{n} \left[\mu^2_{G_1}(x_i) + \nu^2_{G_1}(x_i)\right] = \sum_{j=1}^{n} \lambda_j^2(G_1)$$

and

$$E_{IFG}(G_2) = \sum_{i=1}^{n} \left[\mu^2_{G_2}(x_i) + \nu^2_{G_2}(x_i)\right] = \sum_{j=1}^{n} \lambda_j^2(G_2).$$

The covariance of the IFGs G_1 and G_2 is defined as

$$C_{IFG}(G_1, G_2) = \sum_{i=1}^{n} \left[\mu_{G_1}(x_i)\mu_{G_2}(x_i) + \nu_{G_1}(x_i)\nu_{G_2}(x_i)\right].$$

Therefore, the correlation coefficient measure of IFGs G_1 and G_2 are given by the equation

$$K_{IFG}(G_1, G_2) = \frac{C_{IFG}(G_1, G_2)}{\sqrt{E_{IFG}(G_1)E_{IFG}(G_2)}} = \frac{\sum_{i=1}^{n}\left[\mu_{G_1}(x_i)\mu_{G_2}(x_i) + \nu_{G_1}(x_i)\nu_{G_2}(x_i)\right]}{\sqrt{\sum_{i=1}^{n}\left[\mu_{G_1}^2(x_i) + \nu_{G_1}^2(x_i)\right]}\sqrt{\sum_{i=1}^{n}\left[\mu_{G_2}^2(x_i) + \nu_{G_2}^2(x_i)\right]}}$$

Alternately, Xu et al., developed an alternate version of the CC of IFGs C and D, so the same form can be converted on IFGs G_1 and G_2 as follows.

$$K_{IFG}(G_1, G_2) = \frac{\sum_{i=1}^{n}\left[\mu_{G_1}(x_i)\mu_{G_2}(x_i) + \nu_{G_1}(x_i)\nu_{G_2}(x_i)\right]}{Max\left\{\left[\sum_{i=1}^{n}\left[\mu_{G_1}^2(x_i) + \nu_{G_1}(x_i)\right]\right]^{\frac{1}{2}}, \left[\sum_{i=1}^{n}\left[\mu_{G_2}^2(x_i) + \nu_{G_2}^2(x_i)\right]\right]^{\frac{1}{2}}\right\}}$$

or

$$K_{IFG}(G_1, G_2) = \frac{\sum_{i=1}^{n}\left[\mu_{G_1}(x_i)\mu_{G_2}(x_i) + \nu_{G_1}(x_i)\nu_{G_2}(x_i) + \pi_{G_1}(x_i)\pi_{G_2}(x_i)\right]}{Max\left\{\left[\sum_{i=1}^{n}\left[u_{G_1}^2(x_i) + \nu_{G_1}^2(x_i) + \pi_{G_1}^2(x_i)\right]\right]^{\frac{1}{2}}, \left[\sum_{i=1}^{n}\left[\mu_{G_2}^2(x_i) + \nu_{G_2}^2(x_i) + \pi_{G_2}^2(x_i)\right]\right]^{\frac{1}{2}}\right\}}$$

or

$$K_{IFG}(G_1, G_2) = \frac{\sum_{i=1}^{n}\left[\mu_{G_1}(x_i)\mu_{G_1}(x_i) + \nu_{G_1}(x_i)\nu_{G_2}(x_i) + \pi_{G_1}(x_i)\pi_{G_2}(x_i)\right]}{\left\{\sqrt{\sum_{i=1}^{n}\left[\mu_{G_1}^2(x_i) + \nu_{G_1}^2(x_i) + \pi_{G_1}^2(x_i)\right]}\sqrt{\sum_{i=1}^{n}\left[\mu_{G_2}^2(x_i) + \nu_{G_2}^2(x_i) + \pi_{G_2}^2(x_i)\right]}\right\}}.$$

The function K_{IFG} satisfies the following conditions

- (P_1): $0 \leq K_{IFG}(G_1, G_2) \leq 1$,
- (P_2): $K_{IFG}(G_1, G_2) = K_{IFG}(G_1, G_2)$,
- (P_3): $K_{IFG}(G_1, G_2) = 1$, if $G_1 = G_2$.

3 GROUP DECISION-MAKING BASED ON INTUITIONISTIC FUZZY GRAPHS LAPLACIAN ENERGY AND CORRELATION COEFFICIENT

3.1 Algorithm

For the purpose of finding GDMP based on IFPR, let $\omega = (\omega_1, \omega_2, \ldots, \omega_m)$ be a subjective loading vector of authorities, where $\omega_k > 0,\ k = 1, 2, \ldots, m\ with\ \sum_{i=1}^{m}\ \omega_i = 1$.

Step (i). Calculate the $LE(G_i)$ using the following equations.

$$\begin{aligned} LE(G_i) &= \sum_{i=1}^{n} \left| \lambda_i - \frac{2\sum_{1\le i\le j\le n} \mu(v_i, v_j)}{n} \right|, \\ LE(G_i) &= \sum_{i=1}^{n} \left| \theta_i - \frac{2\sum_{1\le i\le j\le n} \nu(v_i, v_j)}{n} \right|. \end{aligned} \tag{1}$$

Step (ii). Calculate the weight ω_k^a by using Laplacian energy of the authorities e_k using the equation

$$\omega_k^a = ((\omega_\mu)_k, (\omega_\nu)_k) = \left[\frac{LE((G_\mu)_k)}{\sum_{i=1}^{m} LE((G_\mu)_i)}, \frac{LE((G_\nu)_k)}{\sum_{i=1}^{m} LE((G_\nu)_i)} \right]. \tag{2}$$

Step (iii). Calculate the Karl Pearson's correlation coefficient $K(G_s, G_l)$ between G_s and G_l for $s \neq l$, using the equation

$$K_{IFG}(G_s, G_l) = \frac{\sum_{i=1}^{n} \left[\mu_{G_s}(x_i)\mu_{G_l}(x_i) + \nu_{G_s}(x_i)\nu_{G_l}(x_i)\right]}{\sqrt{\sum_{i=1}^{n} \left[\mu_{G_s}^2(x_i) + \nu_{G_s}^2(x_i)\right]} \sqrt{\sum_{i=1}^{n} \left[\mu_{G_l}^2(x_i) + \nu_{G_l}^2(x_i)\right]}}. \tag{3}$$

Compute the average correlation coefficient degree $K(G_s)$ to the others by using the equation

$$K(G_s) = \frac{1}{m-1} \sum_{l=1, s\neq l}^{m} K(G_s, G_l), \quad s = 1, 2, \ldots, m. \tag{4}$$

Step (iv). Compute the weight ω_s^b determined by $K(G_s)$ of the authority e_k, using the equation

$$\omega_s^b = \frac{K(G_s)}{\sum_{i=1}^{m} K(G_i)}, \quad s = 1, 2, \ldots, m. \tag{5}$$

Step (v). Calculate the authority $e_k's$ objective weight ω_s^2 using the following equation

$$\omega_s^2 = \eta\, \omega_s^a + (1 - \eta)\, \omega_s^b, \quad \eta \in [0, 1], \quad s = 1, 2, \ldots, m. \tag{6}$$

Step (vi). Incorporate the weight ω_s with authority e_k subjective weight ω_s^a and objective weight ω_s^2 using the equation

$$\omega_s = \gamma\omega_s^1 + (1-\gamma)\omega_s^2, \quad \gamma \in [0,1], \quad s = 1,2,\ldots,m. \tag{7}$$

3.2 Procedure – I

Step (vii). Use the equation

$$\tau_i^{(s)} = \frac{1}{n}\sum_{j=1}^{n} \tau_{ij}^{(s)}, \tag{8}$$

where $i = 1,2,\ldots,m$, to obtain the aggregate intuitionistic ambiguity value of the option $\tau_i^{(s)}$ across all alternatives.

Step (viii). Use the equation

$$\tau_i = \sum_{i=1}^{m} \omega_s \tau_i^{(s)}, \quad \forall i = 1,2,\ldots,m \tag{9}$$

to make a total intuitionistic ambiguity value of the alternative τ_i over other choices by summing all $\tau_i^{(s)}$ $(s = 1,2,\ldots,n)$, corresponding to n-authorities.

Step (ix). Calculate the rank function from the equation

$$K(\tau_i) = \mu_i - \nu_i \tag{10}$$

of τ_i if the better value of $K(\tau_i)$ is the finer alternate τ_i, then the alternates must be ranked in groups.

3.3 Procedure – II

Step (i). Determine the supportive IFPR as $M = (\tau_{ij})_{n\times n}$ by the equation

$$\tau_{ij} = (\mu_{ij}, \nu_{ij}) = \left(\sum_{l=1}^{m} \omega_l \mu_{ij}^{(l)}, \sum_{l=1}^{m} \omega_l \nu_{ij}^{(l)}\right), \quad i,j = 1,2,\ldots,n. \tag{11}$$

Step (ii). For every choice x_i, decide the correlation coefficient value $K(M^i, M^+)$ between M^i and M^+ and the correlation coefficient value $K(M^i, M^-)$ between M^i and M^- using the equations

$$K(M^i, M^+) = \frac{1}{n}\sum_{j=1}^{n} \frac{\mu_{ij}(1) + \nu_{ij}(0)}{\sqrt{\mu_{ij}^2 + \nu_{ij}^2}\sqrt{1^2 + 0^2}} = \frac{1}{n}\sum_{j=1}^{n} \frac{\mu_{ij}}{\sqrt{\mu_{ij}^2 + \nu_{ij}^2}} \tag{12}$$

and

$$K(M^i, M^-) = \frac{1}{n}\sum_{j=1}^{n} \frac{\mu_{ij}(0) + \nu_{ij}(1)}{\sqrt{\mu_{ij}^2 + \nu_{ij}^2}\sqrt{0^2 + 1^2}} = \frac{1}{n}\sum_{j=1}^{n} \frac{\nu_{ij}}{\sqrt{\mu_{ij}^2 + \nu_{ij}^2}}. \tag{13}$$

Step (iii). For each choice x_i, ascertain its estimate value by the equation

$$h(x_i) = \frac{K(M^i, M^+)}{K(M^i, M^+) + K(M^i, M^-)}. \tag{14}$$

The two procedures (I and II) listed above are intended for acquiring the included loads and ranking the substitutes. When the value of $h(x_i)$ is greater, the alternative x_i is preferred. The finest ranking of the substitutes is then available for decision-makers.

4 FLOW CHART

The flowchart below illustrates how the suggested technique would work to get the alternate rankings.

5 APPLICATION: FINEST SELECTION OF MONEY-INVESTING SCHEMES

Suppose a man who wants to invest his money in any of the four categories such as Fixed deposit (F_D, x_1), Govt bonds (G_B, x_2), Postal savings (P_S, x_3), and Shares (S_H, x_4) (Wang et al. 2005) [32]. He can only pick one based on three criteria such as Tax benefits (e_1), Risk coverage (e_2) and Rate of interest (e_3). Due to his inadequate expertise, he wanted to seek advice from experts who could offer the finest investment strategy. As a result, the experts will apply IFGs to express their preference ratings in order to find the original ranking information, which is provided in the intuitionistic fuzzy decision matrices. It should be noted that the criteria are classified into two types:

1. Benefit type and
2. Price type.

This should be considered by the experts and client when selecting preference values.

To determine one of the most desired categories, the recommended experts use the appropriate aggregate decision information. In order to choose the best category, they use the correlation coefficient and LE of IGFs based on GDMP as follows.

From Figure 2, the IFAM is defined as

$$A(G_1) = \begin{bmatrix} (0,0) & (0.2,0.4) & (0.5,0.4) & (0.7,0.1) \\ (0.4,0.2) & (0,0) & (0.3,0.5) & (0.4,0.5) \\ (0.4,0.5) & (0.5,0.3) & (0,0) & (0.8,0.2) \\ (0.1,0.7) & (0.5,0.4) & (0.2,0.8) & (0,0) \end{bmatrix}.$$

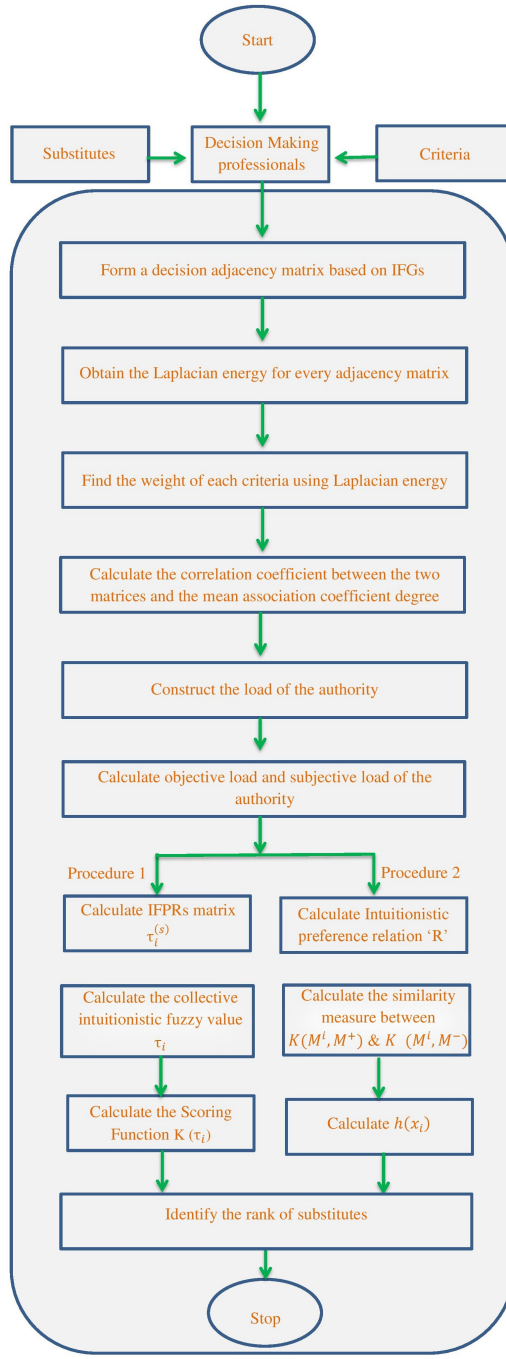


Figure 1. The procedure of ranking the alternatives (substitutes) for GDM assessment

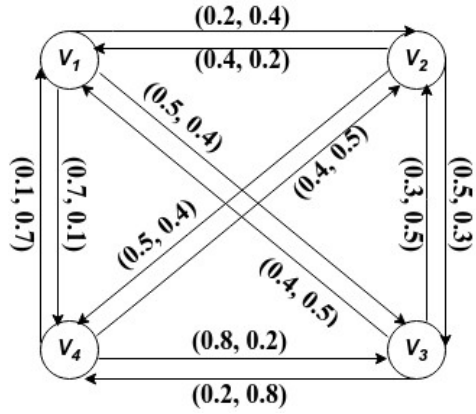


Figure 2. IFG (G_1) related to tax benefits

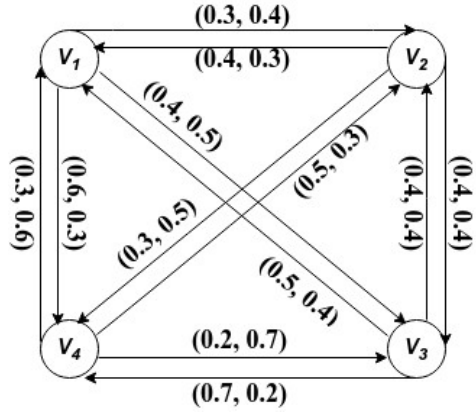


Figure 3. IFG (G_2) related to risk coverage

From Figure 3, the IFAM is defined as

$$A(G_2) = \begin{bmatrix} (0,0) & (0.3,0.4) & (0.4,0.5) & (0.6,0.3) \\ (0.4,0.3) & (0,0) & (0.4,0.4) & (0.5,0.3) \\ (0.5,0.4) & (0.4,0.4) & (0,0) & (0.7,0.2) \\ (0.3,0.6) & (0.3,0.5) & (0.2,0.7) & (0,0) \end{bmatrix}.$$

From Figure 4, the IFAM is defined as

$$A(G_3) = \begin{bmatrix} (0,0) & (0.8,0.1) & (0.3,0.4) & (0.6,0.4) \\ (0.1,0.8) & (0,0) & (0.5,0.3) & (0.4,0.5) \\ (0.4,0.3) & (0.3,0.5) & (0,0) & (0.3,0.7) \\ (0.4,0.6) & (0.5,0.4) & (0.7,0.3) & (0,0) \end{bmatrix}.$$

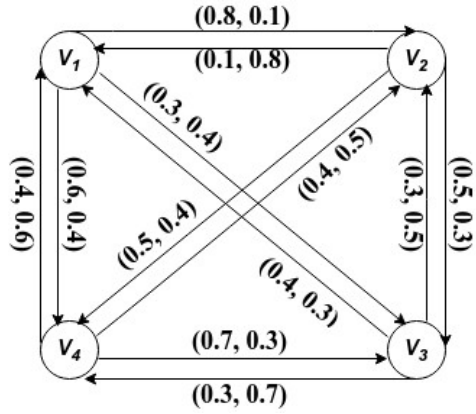


Figure 4. IFG (G_3) related to rate of interest

The Laplacian IFAM $A(G_1)$ of G_1 is given by

$$L(A(G_1)) = D(G_1) - A(G_1),$$

$$L(A(G_1)) = \begin{bmatrix} (1.4, 0.9) & (0,0) & (0,0) & (0,0) \\ (0,0) & (1.1, 1.2) & (0,0) & (0,0) \\ (0,0) & (0,0) & (1.7, 1.0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0.8, 1.9) \end{bmatrix}$$

$$- \begin{bmatrix} (0,0) & (0.2, 0.4) & (0.5, 0.4) & (0.7, 0.1) \\ (0.4, 0.2) & (0,0) & (0.3, 0.5) & (0.4, 0.5) \\ (0.4, 0.5) & (0.5, 0.3) & (0,0) & (0.8, 0.2) \\ (0.1, 0.7) & (0.5, 0.4) & (0.2, 0.8) & (0,0) \end{bmatrix}.$$

The Laplacian IFAM $A(G_2)$ of G_2 is

$$L(A(G_2)) = D(G_2) - A(G_2),$$

$$L(A(G_2)) = \begin{bmatrix} (1.3, 1.2) & (0,0) & (0,0) & (0,0) \\ (0,0) & (1.3, 1.0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (1.6, 1.0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0.8, 1.8) \end{bmatrix}$$

$$- \begin{bmatrix} (0,0) & (0.3, 0.4) & (0.4, 0.5) & (0.6, 0.3) \\ (0.4, 0.3) & (0,0) & (0.4, 0.4) & (0.5, 0.3) \\ (0.5, 0.4) & (0.4, 0.4) & (0,0) & (0.7, 0.2) \\ (0.3, 0.6) & (0.3, 0.5) & (0.2, 0.7) & (0,0) \end{bmatrix}.$$

The Laplacian IFAM $A(G_3)$ of G_3 is

$$L(A(G_3)) = D(G_3) - A(G_3),$$

$$L(A(G_3)) = \begin{bmatrix} (1.7,0.9) & (0,0) & (0,0) & (0,0) \\ (0,0) & (1.0,1.6) & (0,0) & (0,0) \\ (0,0) & (0,0) & (1.0,1.5) & (0,0) \\ (0,0) & (0,0) & (0,0) & (1.6,1.3) \end{bmatrix} - \begin{bmatrix} (0,0) & (0.8,0.1) & (0.3,0.4) & (0.6,0.4) \\ (0.1,0.8) & (0,0) & (0.5,0.3) & (0.4,0.5) \\ (0.4,0.3) & (0.3,0.5) & (0,0) & (0.3,0.7) \\ (0.4,0.6) & (0.5,0.4) & (0.7,0.3) & (0,0) \end{bmatrix}.$$

5.1 Algorithm

Step (i). By formula 1, we calculate the LEs of G_i, $i = 1, 2, 3$.

From Figure 2 and $A(G_1)$ we get

$$LE(G_1) = (2.5796, 2.7298).$$

From Figure 3 and $A(G_2)$ we get

$$LE(G_2) = (2.5000, 2.5000).$$

From Figure 4 and $A(G_3)$ we get

$$LE(G_3) = (2.7425, 2.7047).$$

Step (ii). Using formula 2, we get the weights of G_i determined with LEs as follows:

$$\omega_1^a = (0.3298, 0.3440),$$
$$\omega_2^a = (0.3196, 0.3151)$$

and

$$\omega_3^a = (0.3506, 0.3409).$$

Step (iii). Using 3 formula, we have

$$K(G_1, G_2) = 0.9681,$$
$$K(G_1, G_3) = 0.7794$$

and

$$K(G_2, G_3) = 0.8350.$$

By Equation (4), we get

$$K(G_1) = 0.8738,$$
$$K(G_2) = 0.9016$$

and

$$K(G_3) = 0.8072.$$

Step (iv). By Equation (5), we have $\omega_s^b = \frac{K(G_s)}{\sum_{i=1}^{m} K(G_i)}$, $s = 1, 2, \ldots, m$. then we get

$$\omega_1^b = 0.3383,$$
$$\omega_2^b = 0.3491$$

and

$$\omega_3^b = 0.3126.$$

Step (v). By Equation (6), we have $\omega_s^2 = \eta\omega_s^a + (1-\eta)\omega_s^b$, and taking $\eta = 0.5$ we get

$$\omega_{1,\mu}^2 = 0.3341,$$
$$\omega_{2,\mu}^2 = 0.3344,$$
$$\omega_{3,\mu}^2 = 0.3316$$

and

$$\omega_{1,\nu}^2 = 0.3412,$$
$$\omega_{2,\nu}^2 = 0.3321,$$
$$\omega_{3,\nu}^2 = 0.3268.$$

So, weights of authorities are

$$\omega_1^2 = (0.3341, 0.3412),$$
$$\omega_2^2 = (0.3344, 0.3321)$$

and

$$\omega_3^2 = (0.3316, 0.3268).$$

Step (vi). By Equation (7), we have $\omega_s = \gamma\omega_s^a + (1-\gamma)\omega_s^2$ and taking $\gamma = 0.5$ we get

$$\omega_{1,\mu} = 0.3320,$$
$$\omega_{2,\mu} = 0.3270,$$
$$\omega_{3,\mu} = 0.3411$$

and

$$\omega_{1,\nu} = 0.3426,$$
$$\omega_{2,\nu} = 0.3236,$$
$$\omega_{3,\nu} = 0.3339.$$

So, the impartial weights are

$$\omega_1 = (0.3320, 0.3426),$$
$$\omega_2 = (0.3270, 0.3236)$$

and

$$\omega_3 = (0.3411, 0.3339).$$

5.2 Procedure I

Step (vii). By Equation (8), we have $\tau_i^{(s)} = \frac{1}{n}\sum_{j=1}^{n} \tau_{ij}^{(s)}$, $i = 1, 2, \ldots, m$.

Then from Figure 2 and $A(G_1)$ we get

$$\tau_1^{(1)} = (0.4667, 0.3000),$$
$$\tau_2^{(1)} = (0.3667, 0.4000),$$
$$\tau_3^{(1)} = (0.5667, 0.3334),$$
$$\tau_4^{(1)} = (0.2667, 0.6334).$$

From Figure 3 and $A(G_2)$ we get

$$\tau_1^{(2)} = (0.4334, 0.4000),$$
$$\tau_2^{(2)} = (0.4334, 0.3334),$$
$$\tau_3^{(2)} = (0.5334, 0.3334),$$
$$\tau_4^{(2)} = (0.2667, 0.6000).$$

From Figure 4 and $A(G_3)$ we get

$$\tau_1^{(3)} = (0.5667, 0.3000),$$
$$\tau_2^{(3)} = (0.3334, 0.5334),$$
$$\tau_3^{(3)} = (0.3334, 0.5000),$$
$$\tau_4^{(3)} = (0.5334, 0.4334).$$

Step (viii). By Equation (9), we have $\tau_i = \sum_{s=1}^{m} \omega_s \tau_i^{(s)}$, $i = 1, 2, \ldots, n.$, we get

$$\tau_{1,\mu} = 0.4900, \qquad \tau_{1,\nu} = 0.3324,$$
$$\tau_{2,\mu} = 0.3772, \qquad \tau_{2,\nu} = 0.4230,$$
$$\tau_{3,\mu} = 0.4763, \qquad \tau_{3,\nu} = 0.3891$$

and

$$\tau_{4,\mu} = 0.3577, \qquad \tau_{4,\nu} = 0.5559.$$

Therefore

$$\tau_1 = (0.4900, 0.3324),$$
$$\tau_2 = (0.3772, 0.4230),$$
$$\tau_3 = (0.4763, 0.3891)$$

and

$$\tau_4 = (0.3577, 0.5559).$$

Step (ix). By Equation (10), we have $K(\tau_i) = \mu_i - \nu_i$, we get

$$K(\tau_1) = 0.1576,$$
$$K(\tau_2) = -0.0450,$$
$$K(\tau_3) = 0.0872,$$
$$K(\tau_4) = -0.1982.$$

Therefore $K(\tau_1) > K(\tau_3) > K(\tau_2) > K(\tau_4)$, as a result $\tau_1 > \tau_3 > \tau_2 > \tau_4$.

The resulting ranking order is the same for all the values of γ ($\gamma \in [0, 1]$), not only the one ($\gamma = 0.5$) used in Equation (7).

5.3 Procedure II

Step (i). In this part, we present the position outcome potential using our comparable correlation coefficient approach. By Equation (11) in method II, we form the group IFPR as follows.

From the matrices $A(G_1)$, $A(G_2)$ and $A(G_3)$ we get

$$M = \begin{bmatrix} (0,0) & (0.4376, 0.2999) & (0.3994, 0.4324) & (0.6333, 0.2649) \\ (0.2977, 0.4327) & (0,0) & (0.4010, 0.4009) & (0.4327, 0.4353) \\ (0.4327, 0.4009) & (0.3991, 0.3992) & (0,0) & (0.6309, 0.3670) \\ (0.2677, 0.6343) & (0.4347, 0.4324) & (0.3706, 0.6008) & (0,0) \end{bmatrix}.$$

Step (ii). By using the Equations (12) and (13), we achieve

$$K(M^1, M^+) = 0.6065,$$

$$K(M^2, M^+) = 0.4947,$$

$$K(M^3, M^+) = 0.5762,$$

$$K(M^4, M^+) = 0.4057$$

and

$$K(M^1, M^-) = 0.4215,$$

$$K(M^2, M^-) = 0.5601,$$

$$K(M^3, M^-) = 0.4724,$$

$$K(M^4, M^-) = 0.6194.$$

Step (iii). Next, for each choice x_i, $(i = 1, 2, 3, 4)$, Equation (14) provides the computation standards as

$$h(x_1) = 0.5900,$$

$$h(x_2) = 0.4690,$$

$$h(x_3) = 0.5494,$$

$$h(x_4) = 0.3958.$$

Since $h(x_1) > h(x_3) > h(x_2) > h(x_4)$, as a result $x_1 > x_3 > x_2 > x_4$.

The resulting ranking order is the same for all the values γ, where $\gamma \in [0, 1]$.

According to $Xu's$ algorithm [33] with Procedures I and II, rank wise Fixed deposit (x_1) is at the top position, Shares (x_4) are at the last, and Govt bonds (x_2) and Postal savingas (x_3) are in the middle position. Also,the position ordering of alternatives is the same for both procedures and are shown in the following tables.

After the assessment, the decision-maker concludes that a fixed deposit is the best option for a person looking to invest money among the four categories mentioned. The overall analysis revealed that the two working methods produced the same ranking order. Furthermore, when compared to the method (see [22]), this approach yields slightly faster results.

γ	ω	τ
0.3	$\omega_1 = (0.3328, 0.3420)$	$\tau_1 = (0.4894, 0.3327)$
	$\omega_2 = (0.3298, 0.3270)$	$\tau_2 = (0.3774, 0.4662)$
	$\omega_3 = (0.3373, 0.3310)$	$\tau_3 = (0.4770, 0.3885)$
		$\tau_4 = (0.3566, 0.5568)$
0.5	$\omega_1 = (0.3320, 0.3426)$	$\tau_1 = (0.4900, 0.3324)$
	$\omega_2 = (0.3270, 0.3236)$	$\tau_2 = (0.3772, 0.4230)$
	$\omega_3 = (0.3411, 0.3339)$	$\tau_3 = (0.4763, 0.3891)$
		$\tau_4 = (0.3577, 0.5559)$
0.7	$\omega_1 = (0.3311, 0.3432)$	$\tau_1 = (0.4904, 0.3321)$
	$\omega_2 = (0.3240, 0.3202)$	$\tau_2 = (0.3768, 0.4450)$
	$\omega_3 = (0.3449, 0.3367)$	$\tau_3 = (0.4754, 0.3895)$
		$\tau_4 = (0.3587, 0.5554)$

Table 2. The table values of the alternatives for distinct values of γ using Xu's technique and working procedure I

γ	$\mathbf{K}(\tau_1)$	$\mathbf{K}(\tau_2)$	$\mathbf{K}(\tau_3)$	$\mathbf{K}(\tau_4)$	**Ranking**
0.3	0.1567	−0.0888	0.0885	−0.2002	$\tau_1 > \tau_3 > \tau_2 > \tau_4$
0.5	0.1576	−0.0450	0.0872	−0.1982	$\tau_1 > \tau_3 > \tau_2 > \tau_4$
0.7	0.1583	−0.0682	0.0859	−0.1967	$\tau_1 > \tau_3 > \tau_2 > \tau_4$

Table 3. The ranking order of the alternatices by using Xu's technique and working procedure I

γ	ω	$\mathbf{K}(\mathbf{M^i}, \mathbf{M}^+)$	$\mathbf{K}(\mathbf{M^i}, \mathbf{M}^-)$
0.3	$(0.3328, 0.3420)$	$K(M^1, M^+) = 0.6060$	$K(M^1, M^-) = 0.4222$
	$(0.3298, 0.3270)$	$K(M^2, M^+) = 0.4954$	$K(M^2, M^-) = 0.5596$
	$(0.3373, 0.3310)$	$K(M^3, M^+) = 0.5736$	$K(M^3, M^-) = 0.4769$
		$K(M^4, M^+) = 0.4047$	$K(M^4, M^-) = 0.6201$
0.5	$(0.3320, 0.3426)$	$K(M^1, M^+) = 0.6065$	$K(M^1, M^-) = 0.4215$
	$(0.3270, 0.3236)$	$K(M^2, M^+) = 0.4947$	$K(M^2, M^-) = 0.5601$
	$(0.3411, 0.3339)$	$K(M^3, M^+) = 0.5762$	$K(M^3, M^-) = 0.4724$
		$K(M^4, M^+) = 0.4057$	$K(M^4, M^-) = 0.6194$
0.7	$(0.3311, 0.3432)$	$K(M^1, M^+) = 0.6068$	$K(M^1, M^-) = 0.4210$
	$(0.3240, 0.3202)$	$K(M^2, M^+) = 0.4940$	$K(M^2, M^-) = 0.5606$
	$(0.3449, 0.3367)$	$K(M^3, M^+) = 0.5593$	$K(M^3, M^-) = 0.4785$
		$K(M^4, M^+) = 0.4067$	$K(M^4, M^-) = 0.6188$

Table 4. The table values of the replacements for distinct values of γ using Xu's technique and working procedure II

γ	$\mathbf{K}(\tau_1)$	$\mathbf{K}(\tau_2)$	$\mathbf{K}(\tau_3)$	$\mathbf{K}(\tau_4)$	**Ranking Order**
0.3	0.5894	0.4696	0.5460	0.3949	$x_1 > x_3 > x_2 > x_4$
0.5	0.5900	0.4690	0.5494	0.3958	$x_1 > x_3 > x_2 > x_4$
0.7	0.5904	0.4684	0.5389	0.3966	$x_1 > x_3 > x_2 > x_4$

Table 5. The ranking order of the replacements for distinct values of γ using Xu's technique and working procedure II

6 CONCLUSION

In general, the opinions of the authorities on alternatives might be unclear and divergent when there is a lack of information or expertise concerning an ambiguous situation. The ideal solution to this issue is the intuitionistic fuzzy concept. In this paper, we illustrated how correlation coefficient measures and Laplacian energy can be used to solve GDM problems when the weight of the criterion is completely unknown and the IFG is the main factor that affects the alternatives. The proposed statistical measure has been successfully implemented for money-investing schemes, and its use will aid in ranking the substitutes. This analogous approach can be used to investigate other aspects of various fuzzy graphs and is also applicable to many IFG types, including Hesitancy fuzzy graphs, Complex fuzzy graphs, etc.

REFERENCES

[1] Zadeh, L.: Fuzzy Sets. Information and Control, Vol. 8, 1965, No. 3, pp. 338–353, doi: 10.1016/S0019-9958(65)90241-X.

[2] Atanassov, K. T.: Intuitionistic Fuzzy Sets. Physica, Heidelberg, 1999, doi: 10.1007/978-3-7908-1870-3_1.

[3] Zadeh, L. A.: Similarity Relations and Fuzzy Orderings. Information Sciences, Vol. 3, 1971, No. 2, pp. 177–200, doi: 10.1016/S0020-0255(71)80005-1.

[4] Kaufmann, A.: Introduction À La Théorie Des Sous-Ensembles Flous À L'usage Des Ingénieurs (Fuzzy Sets Theory). Masson, 1973.

[5] Rosenfeld, A.: Fuzzy Graphs. In: Zadeh, L. A., Fu, K. S., Tanaka, K., Shimura, M. (Eds.): Fuzzy Sets and Their Applications to Cognitive and Decision Processes. Elsevier, 1975, pp. 77–95, doi: 10.1016/B978-0-12-775260-0.50008-6.

[6] Gutman, I.: The Energy of a Graph: Old and New Results. In: Betten, A., Kohnert, A., Laue, R., Wassermann, A. (Eds.): Algebraic Combinatorics and Applications. Springer, 2001, pp. 196–211, doi: 10.1007/978-3-642-59448-9_13.

[7] Balakrishnan, R.: The Energy of a Graph. Linear Algebra and Its Applications, Vol. 387, 2004, pp. 287–295, doi: 10.1016/j.laa.2004.02.038.

[8] Anjali, N.—Mathew, S.: Energy of a Fuzzy Graph. Annals of Fuzzy Mathematics and Informatics, Vol. 6, 2013, No. 3, pp. 455–465.

[9] Rahimi Sharbaf, S.—Fayazi, F.: Laplacian Energy of a Fuzzy Graph. Iranian Journal of Mathematical Chemistry, Vol. 5, 2014, No. 1, pp. 1–10.

[10] Parvathi, R.—Karunambigai, M.: Intuitionistic Fuzzy Graphs. In: Reusch, B. (Ed.): Computational Intelligence, Theory and Applications. Springer, Berlin, Heidelberg, Advances in Intelligent and Soft Computing, Vol. 38, 2006, pp. 139–150, doi: 10.1007/3-540-34783-6_15.

[11] Basha, S. S.—Kartheek, E.: Laplacian Energy of an Intuitionistic Fuzzy Graph. Indian Journal of Science and Technology, Vol. 8, 2015, No. 33, pp. 1–7, doi: 10.17485/ijst/2015/v8i33/79899.

[12] Darvish Falehi, A.: Robust and Intelligent Type-2 Fuzzy Fractional-Order Controller-Based Automatic Generation Control to Enhance the Damping Performance of Multi-Machine Power Systems. IETE Journal of Research, Vol. 68, 2022, No. 4, pp. 2548–2559, doi: 10.1080/03772063.2020.1719908.

[13] Falehi, A. D.: MOPSO Based TCSC–ANFIS–POD Technique: Design, Simultaneous Scheme, Power System Oscillations Suppression. Journal of Intelligent and Fuzzy Systems, Vol. 34, 2018, No. 1, pp. 23–34, doi: 10.3233/JIFS-16241.

[14] Darvish Falehi, A.: An Innovative OANF–IPFC Based on MOGWO to Enhance Participation of DFIG-Based Wind Turbine in Interconnected Reconstructed Power System. Soft Computing, Vol. 23, 2019, No. 23, pp. 12911–12927, doi: 10.1007/s00500-019-03848-0.

[15] Akram, M.—Ishfaq, N.—Sayed, S.—Smarandache, F.: Decision-Making Approach Based on Neutrosophic Rough Information. Algorithms, Vol. 11, 2018, No. 5, Art. No. 59, doi: 10.3390/a11050059.

[16] Akram, M.—Zafar, F.: Rough Fuzzy Digraphs with Application. Journal of Applied Mathematics and Computing, Vol. 59, 2019, No. 1-2, pp. 91–127, doi: 10.1007/s12190-018-1171-2.

[17] Akram, M.—Luqman, A.: Certain Networks Models Using Single-Valued Neutrosophic Directed Hypergraphs. Journal of Intelligent and Fuzzy Systems, Vol. 33, 2017, No. 1, pp. 575–588, doi: 10.3233/JIFS-162347.

[18] Akram, M.—Shahzadi, S.—Smarandache, F.: Multi-Attribute Decision-Making Method Based on Neutrosophic Soft Rough Information. Axioms, Vol. 7, 2018, No. 1, Art. No. 19, doi: 10.3390/axioms7010019.

[19] Sarwar, M.—Akram, M.: An Algorithm for Computing Certain Metrics in Intuitionistic Fuzzy Graphs. Journal of Intelligent and Fuzzy Systems, Vol. 30, 2016, No. 4, pp. 2405–2416, doi: 10.3233/IFS-152009.

[20] Shahzadi, S.—Akram, M.: Graphs in an Intuitionistic Fuzzy Soft Environment. Axioms, Vol. 7, 2018, No. 2, Art. No. 20, doi: 10.3390/axioms7020020.

[21] Naz, S.—Akram, M.—Smarandache, F.: Certain Notions of Energy in Single-Valued Neutrosophic Graphs. Axioms, Vol. 7, 2018, No. 3, Art. No. 50, doi: 10.3390/axioms7030050.

[22] Ramesh, O.—Basha, S. S.: Group Decision Making of Selecting Partner Based on Signless Laplacian Energy of an Intuitionistic Fuzzy Graph with Topsis Method: Study on Matlab Programming. Advances in Mathematics: Scientific Journal, Vol. 9, 2020, No. 8, pp. 5849–5859, doi: 10.37418/amsj.9.8.52.

[23] Xuan Thao, N.: A New Correlation Coefficient of the Intuitionistic Fuzzy Sets and Its Application. Journal of Intelligent and Fuzzy Systems, Vol. 35, 2018, No. 2, pp. 1959–1968, doi: 10.3233/JIFS-171589.

[24] Ye, J.: Fuzzy Decision-Making Method Based on the Weighted Correlation Coefficient Under Intuitionistic Fuzzy Environment. European Journal of Operational Research, Vol. 205, 2010, No. 1, pp. 202–204, doi: 10.1016/j.ejor.2010.01.019.

[25] Akula, N. K.—Sharief Basha, S.: Association Coefficient Measure of Intuitionistic Fuzzy Graphs with Application in Selecting Best Electric Scooter for Mar-

keting Executives. Journal of Intelligent and Fuzzy Systems, 2023, pp. 1–10, doi: 10.3233/JIFS-222510 (in press).

[26] Zeng, W.—Li, H.: Correlation Coefficient of Intuitionistic Fuzzy Sets. Journal of Industrial Engineering International, Vol. 3, 2007, No. 5, pp. 33–40.

[27] Mitchell, H.: A Correlation Coefficient for Intuitionistic Fuzzy Sets. International Journal of Intelligent Systems, Vol. 19, 2004, No. 5, pp. 483–490, doi: 10.1002/int.20004.

[28] Huang, H. L.—Guo, Y.: An Improved Correlation Coefficient of Intuitionistic Fuzzy Sets. Journal of Intelligent Systems, Vol. 28, 2019, No. 2, pp. 231–243, doi: 10.1515/jisys-2017-0094.

[29] Szmidt, E.—Kacprzyk, J.: Correlation of Intuitionistic Fuzzy Sets. In: Hüllermeier, E., Kruse, R., Hoffmann, F. (Eds.): Computational Intelligence for Knowledge-Based Systems Design (IPMU 2010). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6178, 2010, pp. 169–177, doi: 10.1007/978-3-642-14049-5_18.

[30] Garg, H.—Rani, D.: A Robust Correlation Coefficient Measure of Complex Intuitionistic Fuzzy Sets and Their Applications in Decision-Making. Applied Intelligence, Vol. 49, 2019, No. 2, pp. 496–512, doi: 10.1007/s10489-018-1290-3.

[31] Khaleie, S.—Fasanghari, M.: An Intuitionistic Fuzzy Group Decision Making Method Using Entropy and Association Coefficient. Soft Computing, Vol. 16, 2012, No. 7, pp. 1197–1211, doi: 10.1007/s00500-012-0806-8.

[32] Wang, Y. M.—Yang, J. B.—Xu, D. L.: Interval Weight Generation Approaches Based on Consistency Test and Interval Comparison Matrices. Applied Mathematics and Computation, Vol. 167, 2005, No. 1, pp. 252–273, doi: 10.1016/j.amc.2004.06.080.

[33] Xu, Z.—Hu, H.: Projection Models for Intuitionistic Fuzzy Multiple Attribute Decision Making. International Journal of Information Technology & Decision Making, Vol. 9, 2010, No. 2, pp. 267–280, doi: 10.1142/S0219622010003816.

Naveen Kumar Akula is a Ph.D. researcher in the Department of Mathematics, School of Advanced Sciences, Vellore Institute of Technology, Vellore, Tamil Nadu, India. He received his M.Sc. degree in mathematics from the Sri Venkateswara University, Tirupati, Andhra Pradesh, India. His research focuses primarily on intuitionistic fuzzy graphs using Laplacian energy and some statistical measures.

Sharief Basha Shaik received his Ph.D. in mathematics from the Sri Venkateswara University, Tirupati, Andhra Pradesh, India in 2009. In 1995 he received his M.Sc. degree in mathematics from the Sri Venkateswara University, Tirupati, Andhra Pradesh, India. Since 1998, he has worked as Assistant Professor, Associate Professor, and Professor in Madina Engineering College, Kadapa, Andhra Pradesh, India. He is presently working as Assistant Professor in the Department of Mathematics, School of Advanced Sciences, Vellore Institute of Technology, Vellore, Tamil Nadu, India. His main research interest is in the area of graph theory, fuzzy graphs, neural networks, and neuro-fuzzy systems.

Computing and Informatics, Vol. 42, 2023, 457–479, doi: 10.31577/cai_2023_2_457

POINTHUMAN: RECONSTRUCTING CLOTHED HUMAN FROM POINT CLOUD OF PARAMETRIC MODEL

Zongguo Mo, Qicong Wang*

Department of Computer Science and Technology
Xiamen University, Xiamen 361000, China
&
Shenzhen Research Institute, Xiamen University
Shenzhen 518000, China
e-mail: 23020201153790@stu.xmu.edu.cn, qcwang@xmu.edu.cn

Hua Shi

School of Optoelectronic and Communication Engineering
Xiamen University of Technology, Xiamen 361024, Fujian, China
e-mail: shihua@xmut.edu.cn

Baobing Zhang*, Wanxin Sui

Department of Electronic and Electrical Engineering, College of Engineering,
Design and Physical Sciences, Brunel University London, Uxbridge UB8 3PH, UK
e-mail: {Baobing.Zhang, cynthia.sui}@brunel.ac.uk

Abstract. It is very difficult to accomplish the 3D reconstruction of the clothed human body from a single RGB image, because the 2D image lacks the representation information of the 3D human body, especially for the clothed human body. In order to solve this problem, we introduced a priority scheme of different body parts spatial information and proposed PointHuman network. PointHuman combines the spatial feature of the parametric model of the human body with the

* Corresponding author

implicit functions without expressive restrictions. In PointHuman reconstruction framework, we use Point Transformer to extract the semantic spatial feature of the parametric model of the human body to regularize the implicit function of the neural network, which extends the generalization ability of the neural network to complex human poses and various styles of clothing. Moreover, considering the ambiguity of depth information, we estimate the depth of the parameterized model after point cloudization, and obtain an offset depth value. The offset depth value improves the consistency between the parameterized model and the neural implicit function, and accuracy of human reconstruction models. Finally, we optimize the restoration of the parametric model from a single image, and propose a depth perception method. This method further improves the estimation accuracy of the parametric model and finally improves the effectiveness of human reconstruction. Our method achieves competitive performance on the THuman dataset.

Keywords: 3D reconstruction, clothed human reconstruction, SMPL estimation

1 INTRODUCTION

By using intelligent devices to describe and represent the real world has always been a hot and difficult research direction in computer vision and computer graphics areas. The research field of 3D vision has also fast developed in recent years. Lots of 3D human reconstruction research results have been applied in real life. Such as virtual fitting, AR, VR, film, television and 3D games, etc. Creating value for the society while it also brings economic effects. For computer to understand human behavior, participate in human life, realize interaction with humans, it is very important for us to obtain the 3D pose and shape of the human body.

Deep learning is a branch of machine learning. Many traditional machine learning algorithms have a limited learning capacity, and therefore cannot learn the total amount of knowledge with increasing amounts of data. However, deep learning systems can improve performance by accessing more data, a machine surrogate for "more experience". Once a machine has gained enough experience through deep learning, it can be used for specific tasks such as driving a car, face recognition, diagnosing a disease, detecting machine malfunctions, etc. Deep learning can provide a variety of solutions in computer vision, natural language processing, and many other applications. In the future metaverse era, deep learning can play an important role, for example in 3D reconstruction, where deep learning can perform such functions.

The current 3D human body reconstruction methods can be classified into three categories. The first category is to use the existing parametric human body model, such as human parametric model [1], which can directly restore the three-dimensional human body model from a single RGB image or video. The difficulty of recovering 3D model directly from RGB image or video lies in the com-

plexity of the human body, clarity, occlusion, clothing, lighting and the inherent ambiguity of 2D inferring 3D poses. This method does not need specific depth sensor and has a low dependence on external. It is widely used. However, the accuracy of the currently constructed model is far from enough, especially for detailed feature with a hand and face are obviously missing, and no clothing details.

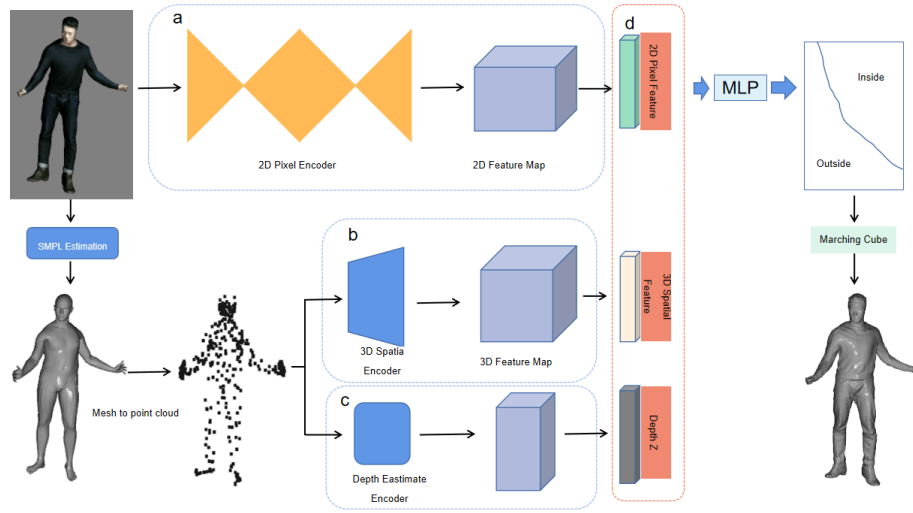


Figure 1. The pipeline of human reconstruction. Given an input image, 2D Pixel Encoder performs pixel feature extraction on the image (a). SMPL estimation is performed on the image to obtain the parametric model, and the parametric model is transformed into a point cloud. 3D Spatial Encoder performs spatial feature extraction on these point clouds (b). Depth Estimation Encoder estimates the offset depth value for these point clouds. The features of a, b and c are fused, and sent to the multi-layer perceptron to predict the distance symbol function value (d), and finally the human body mesh model is obtained.

The second category is the parametric model's deformation. Adding offsets (SMPL + D) to the vertices of the human parametric model to represent a clothed human body is a simple model that is widely used and easy to parameterize. The body geometry of the target pose is obtained by adding the offsets of the vertices under the standard pose of the human parametric model, and then using the skin deformation. There are several previous study [2, 3, 4, 5] to implement. It is difficult to represent SMPL + D for clothes that are not consistent with the SMPL mesh topology, such as open jackets and skirts. Moreover, the binding of clothing to SMPL vertices, especially the binding of mask weights, leads to loose clothing that may be distorted in the mask deformation. And the SMPL + D approach is poorly robust in reconstructing clothing away from the body. It would be better not to adopt a parametric model, such as [6, 7, 8, 9, 10].

The third category is implicit function without using the parameterized model. The pixel alignment implicit function first introduced by PIFu [6] uses MLP to determine the volume occupancy value for a given 3D location. In order to obtain both global and local feature, PIFu [6] uses a deep network to extract the feature of each pixel, and combine this feature together with the depth information of the corresponding 3D point as the input of the MLP to obtain high-fidelity 3D clothed human body reconstruction. Based on PIFu [6], PIFuHD [11] utilizes higher-precision feature and predictes normal information to obtain clothed human reconstructions with more geometric details. Hong et al. [12] use the stereoscopic sense of binocular camera to introduce voxel features to the human body reconstruction and get better results. Summary, the 3D reconstruction of the clothed human body reconstructed from a single RGB image still has the following problems. First, the complexity of the action pose of the person, the ever-changing and different actions of the same person. Second, the self-occlusion of the person, whether the occluded part or the occluded part will lose the integrity Information. Last, RGB images taken by ordinary cameras lack depth information, resulting in depth ambiguity.

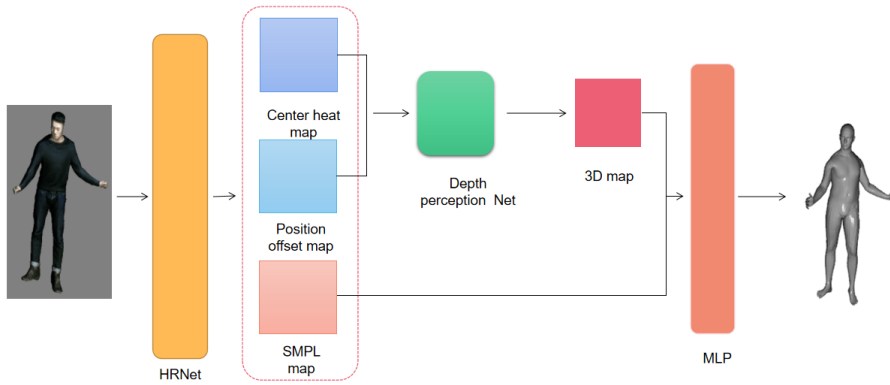


Figure 2. SMPL estimation frame diagram. Inputting an RGB image, HRNet obtains three feature maps: center heat map, position offset map and SMPL map, center heat map and position offset map past depth perception information, and then carry out with the SMPL map Fusion, the SMPL parameters are regressed by the multilayer perceptron

Our three technical contributions are:

- We extract spatial information from the parameterized model, and give the reconstruction network prior knowledge, constrain its spatial expression, meanwhile, impose restrictions on the estimated shape of the human reconstruction. It improves the generalization ability of the neural network to complex human poses and various styles of clothing.
- In order to solve the problem of depth ambiguity, the parametric model contains the relevant coordinate information of each limb of the human body after

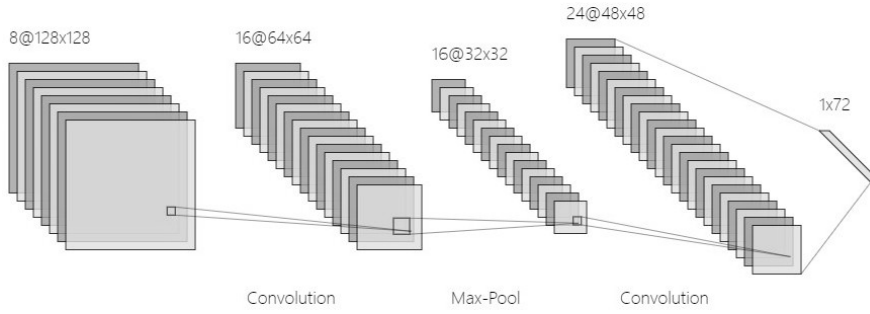


Figure 3. Depth perception Net Frame

point cloud, and the depth can be estimated by the depth network to generate the offset depth value. The offset depth value can use the human body prior information of the predicts depth to guide the occupied space.

- We propose a depth perception method for parameterized model estimation, which reduces the problem of depth ambiguity and restores a more accurate parameterized model.

2 RELATED WORK

2.1 Human Reconstruction Based on Parametric Model

Parametric model capable of changing its parameters to represent the shape of the human body. When human changing its action, the parametric model of the human body will change its parameters to describe the height, short, fat and thin of the human body. Lassner et al. [13] extract 72 joint points of the human body and use random forests to regress SMPL pose and shape parameters. Pavlakos et al. [14] regress SMPL parameters by relying on a smaller number of key points and body contours, further adopt a similar method, then use a segmentation map of human body parts as an intermediate representation. HMR [15] tries to use a weakly supervised method, relying on two-dimensional joint point reprojection penalty and a pre-learned human pose discrimination network, directly using neural network for singe image. Kolotour et al. [5] propose a self-supervised method to solve the same problem. Güler et al. [16] rely on weaker body contour supervision. Rockwell et al. [17] Consider showing only severe occlusion of hand or torso images, to predict the matching SMPL human body. In order to recover more geometric information beyond the body from individual images, such as hand movement and facial expression, Choutas et al. [7] use a body-driven attention technique for extracting high-resolution hand and face from image. A close-up of the part that helps the network predict matching SMPL parameters. Zhang et al. [18] considered how to predict a SMPL human body that matches a 3D scene. For video stream, there are

Figure 4. Our results on a single RGB image. From left to right: the first column is the input image, the second (front), third (side) and fourth (back) columns are the reconstruction results, and the fifth column is the texture inference results, the results show that our method is able to reconstruct high-quality models with robust performance for handling various human poses.

also methods that introduce temporal information to predict SMPL. Among them, Arnab et al. [19] shows that Internet video annotated with SMPLify incorporating temporal continuity can be used to fine-tune HMR results to achieve better results. Kanazaw et al. [20] learn human motion by predicting past and future frames. Sun et al. [15] proposed a temporal model based on a transform network can be used to further improve the effectiveness. VIBE [21] guides action prediction based on priors learned from human sequence motion data. These works focus on using the SMPL parameter space as a homotropic objective. Although the human body reconstruction based on parametric model can capture the movement of the human body and reconstruct the general shape of the human body, it lacks clothing details and is not vivid enough.

2.2 Human Reconstruction Based on Parametric Model Deformation

Adding offsets (SMPL + D) to the vertices of a parametric model of the human body to represent a clothed human body is a simple approach that is widely used and easy to parameterize. By adding the offset of the vertices under the standard pose of the parametric model of the human body, and then using the skin deformation, we obtain the clothing body geometry of the target pose. ClothCap [8] use this representation to separate and reconstruct human clothing for 4D high-quality scan sequences. Zhang et al. [22] use this representation to optimize the shape of naked body that best fit the scan sequences of people. Loop Reg [23] create a self-supervised loop, through end-to-end training, register the scan data of the clothed human body on the SMPL + D representation. Alldieck et al. [2] extract the contour of a rotation sequence of a person roughly in the A pose, and optimize the clothing based on this The SMPL + D representation of the human body. They propose a neural network that uses a few color images and some semantic information to directly return the target SMPL + D representation, greatly increasing the computational speed [24]. Move the texture map space defined in SMP to achieve a higher-resolution SMPL + D representation, which can represent small clothing wrinkles. MGN [4] segmentes the SMPL vertex for different clothing types, so that the reconstructed SMPL + D representation can better express the boundary of the clothing. Bhatnag et al. [25] parameterize the clothing vertex offset as SMPL parameters with the graph convolution representation of clothing parameters, and a generative model of SMPL+D is learned, which supports a small number of clothing types. Inspired by the SMPL + D representation, Sun et al. [28] use hierarchical free-form 3D deformation techniques to improve the predicted body geometry and capture image-compliant details. Weng et al. [26] deform the SMPL model from the normal estimated from a single image to obtain a drivable clothed human body. SMPL + D is simple and compact, but has some limitions. First, there are limited types of clothing that can be expressed.

For clothing that is inconsistent with the SMPL mesh topology, such as open coats, skirts, etc, SMPL + D is difficult to represent. Secondly, due to the binding of the garment to the top of the SMPL, especially the binding of skin weight, resulting in loose clothing and possible skin deformation distortion. SMPL + D method is less robust to garment reconstruction away from the body.

2.3 Human Reconstruction Based on Non-Parametric Model

In order to get rid of the constraints of parametric representation on the complex geometry of the clothed human body, some implicit representations are used for geometric reconstruction. By implicit representation, we mean that a continuous three-dimensional spatial scalar-valued function is defined, and some of its equivalent surfaces are defined as geometric surfaces. The most common implicit representations are the occupancy field (OF) and the signed distance field (SDF). The scalar value of OF is usually a binary value of whether the spatial point is inside

the represented object, while the scalar value of SDF represents the signed distance of the spatial point relative to the represented surface. In the computer, in order to regularize the representation, the spatial implicit function is often discretized with three-dimensional lattice points. More recently, more compact neural representations, capable of efficiently modeling continuous functions, have also become popular in geometric reconstruction. The discrete occupancy field is a lattice discrete representation of a spatially continuous occupancy field. And BodyNet [16] is one of the early works that introduced this representation to human reconstruction. Voxel regression network (VPN) [27] uses an end-to-end convolutional neural network to directly perform voxel regression on 3D human geometry based on various inputs. DeepHuman [9] integrates multi-scale image features into 3D voxel features, solving the problem of poor voxel regression details. Based on the voxel field representation, since voxels reflect occupancy information unlike SDF fields, which have richer geometric information.

The triple memory consumption limits the improvement of resolution and the results are often coarse. The truncated signed distance field is a discretized representation of the SDF field based on three-dimensional lattice points, and at the same time, truncation is performed for larger distance values. This representation is widely used in fusion-based methods using RGB-D inputs.

The pixel-aligned implicit function first introduced by PIFu [6] uses MLP to determine the volume occupancy value for a given 3D location. In order to obtain global and local feature at the same time, PIFu uses a deep network to extract the feature of each pixel, and uses the feature together with the depth information of the corresponding 3D point as the input of MLP, thus obtaining a high-fidelity 3D clothed human body reconstruction. Stereo-PIFu [12] adds voxel-aligned features to pixel-aligned PIFu features to binocular images. And using the predicted voxel for guiding MLP predictions to high accuracy depths that can effectively combat depth blurring, with the recovered geometry details has richer information. Based on PIFu [6], PIFuHD [11] obtained a clothed human body reconstruction with more geometric details by utilizing higher-precision features and predicted normal information. Huang et al. [28] propose a novel multi-scale surface localization algorithm and a direct rendering method without explicit extraction of surface meshes, and for the first time demonstrated real-time reconstruction of the occupancy field of a clothed human body from monocular video and rendering a new perspective. ARCH [28] and ARCH++ [29] try to solve the problem by converting the problem from the pose space to the normative space, but this conversion depends firstly on the pose estimation (HPS) accuracy. Moreover, since the conversion depends on the mask weight attached to SMPL, this weight is hard-coded and defined on the bare body. And forced application it to a clothed person, driven by the action less natural details of the clothes. ICON [30] Predicts SMPL body from image, rendering front and back body normal, and merging it with the original image. Through a normal prediction network, get the positive and negative through normals, apply the normal map to the SMPL. For particularly complex poses, ICON rebuilds as well, but can't do much with looser clothes.

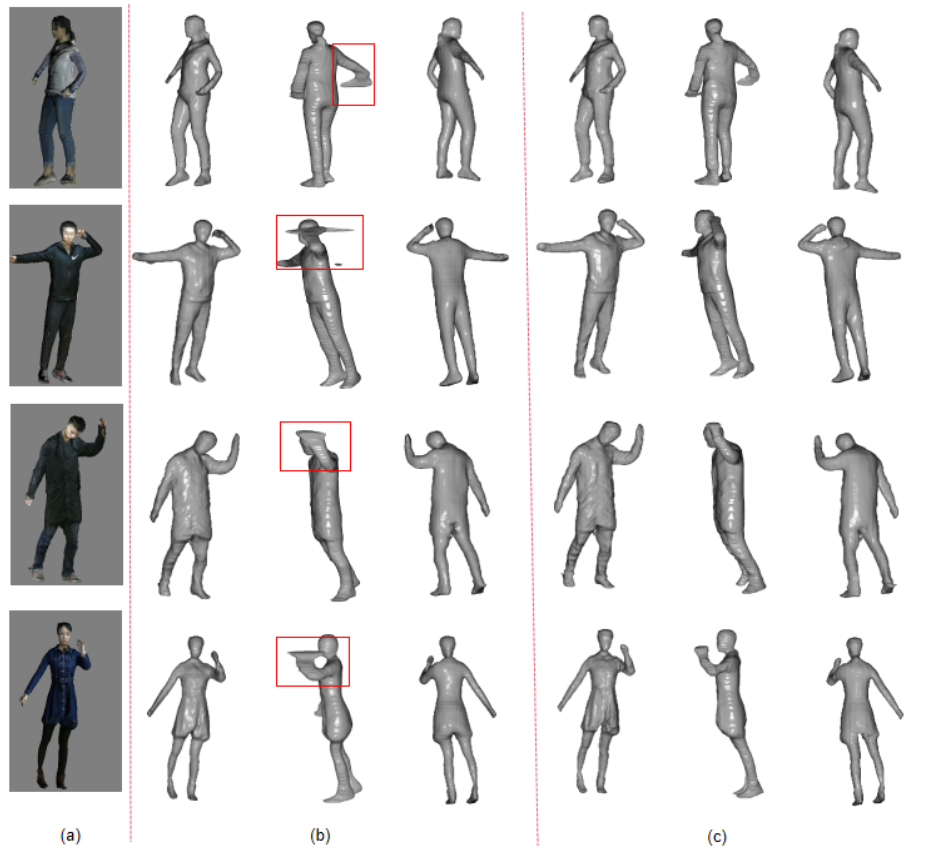


Figure 5. Qualitative comparison against current methods for single-image human model reconstruction: (a) input images, (b) results by PIFu and (c) results by ours

3 METHOD

Our reconstruction of a human body with clothing from a single RGB image is shown in Figure 1. Given a 2D RGB image containing a person, we first estimate its parametric model. The hourglass network performs feature extraction on the image to obtain pixel feature. The parametric model is a grid structure, which consists of vertices and faces. The parametric model is converted into point cloud from mesh. Every point cloud has x, y, and z coordinates. Point Transformer performs 3D spatial feature extraction on every point cloud. After obtaining the spatial information, ResNet extracts the depth information from the point cloud to obtain the depth offset value. The three features are fused as input of the multi-layer perceptron to predict the SDF value. In Figure 1, PointHuman takes a color

Algorithm 1 Training for PointHuman

Input: set of data D. number of optimization steps K and batch size B.
Initialization: randomly initialize g, h, z and fv.

$x \leftarrow 1$
while $x \leq K$ **do**
  $\mathcal{B} \leftarrow \{s_i \in \mathcal{D}\}_{i=1}^{N}$
  for $x_i \in \mathcal{B}$ **do**
    $F(x) = g(I(x))$
    $S(x) = h(I(x))$
    $z(X) = d(I(x))$
    $fv = f((F(x), S(x), z(X))$
    $\mathcal{L}_V = \frac{1}{n}\sum_{i=1}^{n} \left| f_v\left(F_V\left(x_i\right), S\left(x_i\right), z\left(X_i\right)\right) - f_v^*\left(X_i\right)\right|^2$
  end for
  update g, h, z and fv by back-propagation
end while

Output: $s \in \{0, 1\}$

image:

$$f(F(x), S(x), z(X)) = s : s \in \mathbb{R}, \tag{1}$$

where for a 3D point X, $x = \varpi(X)$ is its 2D projection, $S(x) = h(I(x))$ is spatial feature of its parametric model at x. $z(X) = d(I(x))$ is the offset depth value in the camera coordinate space, $F(x) = g(I(x))$ is the image feature at x. For surface reconstruction, we represent the ground truth surface as a 0.5 level-set of a continuous 3D occupancy field:

$$f_v^*(X) = \begin{cases} 1, & \text{if } X \text{ is inside mesh surface,} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The total loss function of our network can be formulated as:

$$\mathcal{L}_V = \frac{1}{n}\sum_{i=1}^{n} \left| f_v\left(F_V\left(x_i\right), S\left(x_i\right), z\left(X_i\right)\right) - f_v^*\left(X_i\right)\right|^2, \tag{3}$$

where $X_i \in \mathbb{R}^3$, $F_V(x) = g(I(x))$ is the image feature from the image encoder g at $x = \varpi(X)$ and n is the number of sampled points. Given the input image, the corresponding parameterized model and the corresponding mesh, the parameters of the image encoder, 3D spatial encoder, depth estimation encoder and fv are updated jointly by minimization so that they are consistent with the input image. The parameters of the image encoder, 3D spatial encoder, depth estimation encoder and fv are updated jointly by minimizing Equation (3). The Algorithm 1 provides the training procedure of our proposed framework.

3.1 Spatial Information Extraction

Spatial shape information is one of the characteristics of a 3D object, which contains the representation information of the object and it is an important input for 3D reconstruction. The mesh structure is one of the manifestations of a 3D object, which reflects the size and shape of the object itself. The mesh consists of multiple triangles on one side and multiple discrete points are used to represent continuous faces in the real world. The point cloud of the mesh is ignore the lines between the vertices, take only the vertices, and use all points on the grid, preserving their spatial shape information. Combining them together is the point cloud of the mesh. The point cloud is a collection of three-dimensional data. The point cloud of the grid still retains its size and shape structure, i.e. spatial information. In order to obtain the spatial geometric information of the parametric model, we perform feature extraction on its point cloud. Transformer has achieved impressive results in the NLP domain and 2D image analysis. Compared with language or image processing, transformer may be more suitable for point cloud processing, because the point cloud is essentially a collection of embedded metric spaces, and the core self-attention of Transformer is a collection operator. In addition to this conceptual matching, Transformer has actually achieved good results in the field of point cloud data processing. Therefore, in this paper, we use Point Transformer [31] to extract geometric information from the point cloud of the parametrized model. Point Transformer adopts a network structure similar to U-net [32]. The first half is down-sampling, The second half has the application of trilinear interpolation to obtain the surface information. The first half and the second half are connected to the information, and the network can then extract the deep spatial information of the parameterized model. Point Transformer uses the subtraction relation and add a position encoding δ to both the attention vector γ and the transformed features α:

$$\mathbf{y}_i = \sum_{\mathbf{x}_j \in \mathcal{X}(i)} \rho\left(\gamma\left(\varphi\left(\mathbf{x}_i\right) - \psi\left(\mathbf{x}_j\right) + \delta\right)\right) \odot \left(\alpha\left(\mathbf{x}_j\right) + \delta\right), \tag{4}$$

where $\mathcal{X}(i) \subseteq \mathcal{X}$ is a set of points in a local neighborhood (specifically, k nearest neighbors) of $\mathbf{x}_i$.

3.2 Estimation of Depth Information

Point cloud of the parametric model has 6 890 vertices, i.e. 6 890 3D coordinates, which contain the relevant depth information of the body. To solve the depth ambiguity problem, point cloud of the parametric model contains relative coordinate information of each human limb, which can be used by the network to estimate depth and generate offset depth values. In addition, the offset depth value can be used to guide occupancy prediction using priority information of the predicted depth. Specifically, the offset depth value makes the network easier to train and allows us to produce good surface detail, reducing the occurrence of limb breakage and breakage. Thus, the offset depth value actually acts

as a bridge between predicted depth and occupancy prediction. For some cases, such as the hand in front of the torso, there will be some discontinuous areas in the predicted depth map. In these cases, the back side of the obscured query point will change discontinuously, leading to unnaturally distorted reconstruction results.

We use ResNet [33] for depth estimation, the z-coordinate values of 6 890 vertices are used as input to obtain the offset depth difference of the body torso of the parameterized model. So the input information is vector of size

$$\boldsymbol{R} \in \mathbb{R}^{6\,890\times 1}.$$

That is the coordinates of the point cloud. The last layer of output is vector of size

$$\boldsymbol{R} \in \mathbb{R}^{256\times 5\,000}.$$

The offset depth value and the depth value of the camera are stitched together to get fused depth value. Fused depth value are added together and fed into the multilayer perceptron.

3.3 Estimating Parametric Model

ROMP [34] aims to recover 3D human body from a single image, but due to the lack of depth information, the correct human body cannot be recovered for self-occlusion. Based on this, we propose a depth perception method to solve this problem. We use the HRnet [35] network to process the image, output the center heat map, the position offset map and the SMPL feature map.

The center heat map and the position offset map are fused and fed into the depth perception network to obtain a 3D feature map. 3D feature map and SMPL feature map get parameters of parametric model through multilayer perceptron regression.

The flow chart is shown in Figure 2. The layout of the depth perception network is convolutional layer–pooling layer–convolutional layer–activation function–output layer, the output is the feature vector of size

$$\boldsymbol{R} \in \mathbb{R}^{72\times 1}.$$

Depth perception network structure is shown in Figure 3. And the SMPL feature map is the feature vector of size

$$\boldsymbol{R} \in \mathbb{R}^{82\times 1}.$$

Center heatmap: The front view center heatmap of size

$$\boldsymbol{M_F} \in \mathbb{R}^{1\times H\times W}.$$

It is aligned in pixel space and uses a Gaussian kernel to represent the likelihood of an object being in 2D. We are adding a second 2D heatmap of size

$$\boldsymbol{M_t} \in \mathbb{R}^{1\times D\times W},$$

which represents an unseen top view. This heatmap represents the likelihood that a person is at a certain depth point. However, this map does not represent metric depths. We synthesize and refine these two maps into a 3D center heatmap

$$\boldsymbol{M_o} \in \mathbb{R}^{1\times D\times H\times W},$$

which uses a 3D Gaussian kernel to represent the 3D position of the detected body center.

Position Offset Map: The discretized center heatmap roughly localizes the body, but we expect the network to produce more precise estimates. Likewise, the position offset map includes a front view and a top view. To improve the granularity of 3D localization, we use additional feature map to refine coarse detections by adding estimated offset vectors at each location. Front view offset feature maps of size

$$\boldsymbol{R_f} \in \mathbb{R}^{1\times H\times W}$$

contain 3D offset vectors. The top view offset map of size

$$\boldsymbol{R_t} \in \mathbb{R}^{1\times D\times W}$$

contains a 1-dimensional offset vector for depth correction.

$$\boldsymbol{R_o} \in \mathbb{R}^{1\times D\times H\times W}$$

corresponds to a 3D center map and contains a 3D offset vector.

SMPL map:

$$\boldsymbol{R} \in \mathbb{R}^{128\times H\times W}$$

contains a 128 grid feature vector at each 2D location. These features are aligned with the input 2D image at the pixel level. After feature fusion with 3D feature map, the SMPL parameters are regressed using a multilayer perceptron.

The front view and top view must work together to estimate the position and depth of the person image. We take the concatenation of the front view map and the backbone feature map as input. We unroll and synthesize 2D maps from front view and top view to generate 3D feature map. The 3D feature map and the SMPL feature map are fused, and the parameters of the parameterized model are regressed through the multilayer perceptron.

4 EXPERIMENTS

In this section we evaluate our approach. Details about the implementation are given in Section 4.1. Our ablation experiment in Section 4.3 and Section 4.4. In

Section 4.2 we demonstrate that our method is able to reconstruct human models with challenging poses. We then compare our method to others methods in Section 4.5. The quantitative evaluation results are given in Table 3.

4.1 Implementation Details

Network Architecture. For image feature extraction, we adapt the Hourglass Stack same encoders in PIFu, take an image of 512×512 as input and outputs a 256-channel feature map with size of 128×128. For spatial feature extraction, we use Point Transformer. Its input resolution is $6\,890 \times 3$, and its output is a 64-channel feature volume with a resolution of $64 \times 128 \times 128$.

For depth formation extraction, we make use of ResNet, its input is the depth value of point cloud of Parametric Model resolution is $6\,890 \times 1$, and its output is a 1-channel vector with a resolution of $1 \times 5\,000$.

Training Data. We use THuman dataset, and it contains 6 795 human meshes with various clothes, shape and poses. We split the dataset into a training set of 5 436 meshes and a testing of 1 359 meshes. THuman dataset is more challenging to learn and less likely to cause over-fitting on upstanding human poses and horizontal camera angles than the dataset used in PIFu. The downside of the dataset is that it lacks high quality texture map for photo-realistic rendering, which might hurt model generalization on in-the-wild natural images.

Network Training. We use Adam optimizer for network training with the learning rate of 1×10^{-3}, the batch size is 8, the number of epochs is 45, and the number of sampled points is 5 000 per subject. The learning rate is decayed by the factor of 0.1 at every $10\,000^{\text{th}}$ iteration. It takes 204 hours for a 3 090 graphics card to complete a training session.

Network Infering. A single RGB image as input, and the improved ROMP predicts its corresponding parametrized model. The parametric model is converted into a point cloud through OPEN3D, which is input to the network together with the image, and the final network outputs the parametric model and the textured reconstructed surface.

4.2 Results

We present the results of our method for 3D human reconstruction from a single RGB image in Figure 4. The input image in Figure 4 contains a variety of complex body poses. The results demonstrate that the ability of our method to reconstruct high-quality 3D human models, as well as its strong ability to handle a variety of human poses. Figures 6, 7 and 8 show the training error, IOU and precision for baseline and different fusing methods. In Figure 4, we can see that after we introduce the parametric model, the reconstructed human body achieves good results, the results show that our method is able to reconstruct high-quality models with robust performance for handling various human poses. Compared with PIFu with only

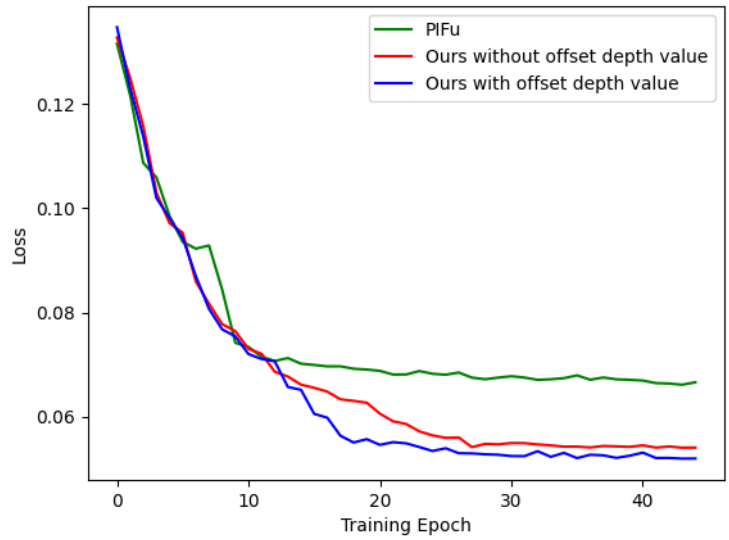


Figure 6. Evaluation of training error. Green line represents PIFu, red line represent our method without offset depth value, and blue line represents our method with offset depth value.

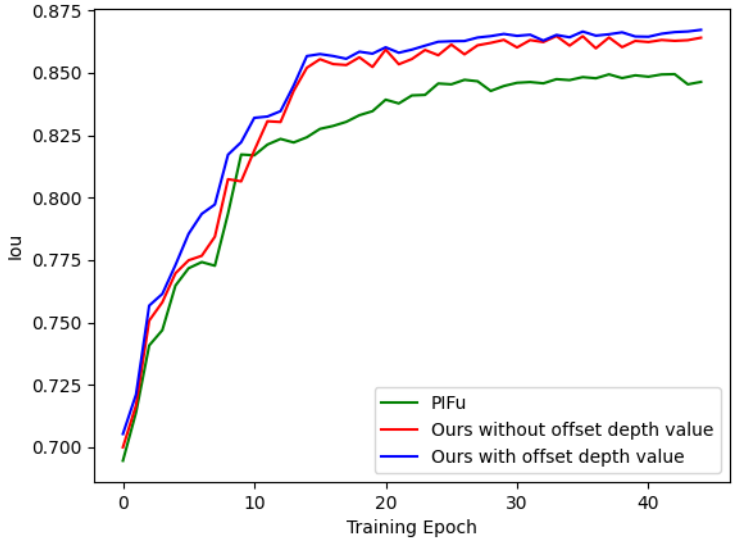


Figure 7. Evaluation of IOU. Green line represents PIFu, red line represents our method without offset depth value, and blue line represents our method with offset depth value.

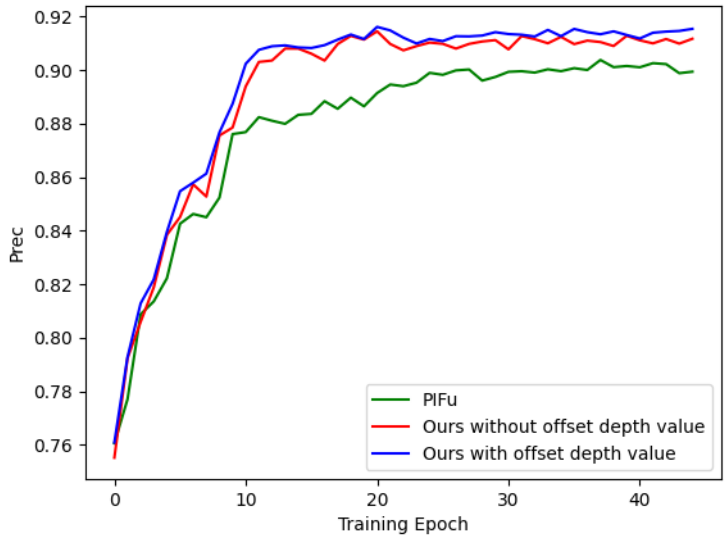


Figure 8. Evaluation of precision. Green line represents PIFu, red line represents our method without offset depth value, and blue line represents our method with offset depth value.

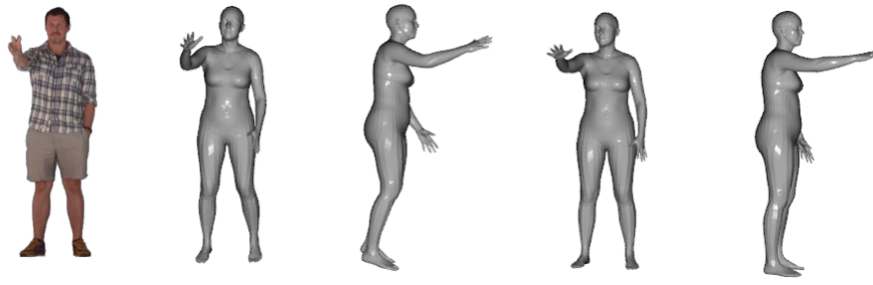


Figure 9. Visualization of the body optimization process. The leftmost column: the input image. The 2nd to 3rd columns: the reconstruction results before reference body optimization. The 4th to 5th columns: the reconstruction results after optimization.

pixel features, after we extract the spatial information of the parameterized model, the Loss, IOU and Prec have made great progress. In addition, after the depth estimation of the parameterized model, the surface details of the reconstructed human body are richer. These indicator is further optimized.

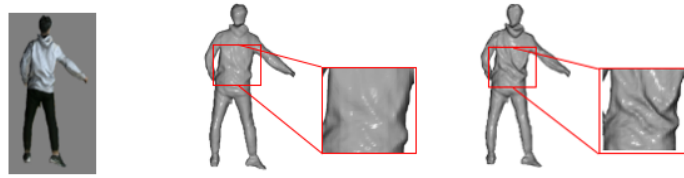


Figure 10. Offset depth values can reconstruct richer details. The leftmost column: the input image. The 2nd to 3rd columns: the reconstruction results without offset depth value and the results with offset depth value.

4.3 Offset Depth Value

We conduct ablation experiments to demonstrate the importance of the inputs in our designed equations for occupancy inference and high-fidelity reconstruction. As shown in Figure 10, PIFu [6] fails to reconstruct reasonable human body geometry using pixel-aligned feature and absolute z-coordinates from a single image due to the complexity and different spatial locations. In contrast, a variant of our PointHuman successfully learns human priors from the same dataset by taking pixel-aligned features, space-aligned features, and the offset depth value as input. Experiments show that our space-aligned feature indeed encode the depth-scale information of query points and further enhance the expressive power of previous work. Table 1 also shows that by replacing absolute z-values with offset depth value, geometric detail can be better recovered. The increase in PointHuman may come from the reconstruction process of the occupancy field, which verifies that our offset depth value indeed effectively utilizes human priors from predicted depth map to guide occupancy inference.

	PSD (cm)	Chamfer (cm)	Normal (cm)
w/o offset depth	2.197	2.312	0.292
w/ offset depth	2.100	2.288	0.281

Table 1. Numerical ablation study of offset depth

4.4 SMPL Estimation

To evaluate the effectiveness of the improved parametric model, we compare the human fitting results before and after improvement using evaluation image. As shown in Figure 9, the optimization step can further fit the SMPL model to the actual human body, resulting in a more accurate body pose estimation. This is also demonstrated in the quantitative evaluation in Table 2, we can also see that the body mesh model reconstruction is also improved after the reference body is optimized.

	PSD (cm)	Chamfer (cm)	Normal (cm)
w/o SMPL optimization	2.203	2.367	0.291
w/ SMPL optimization	2.175	2.301	0.285
Ours using ground-truth SMPL	2.100	2.288	0.281

Table 2. Numerical ablation study of SMPL optimization

4.5 Comparison

We compare our method with several current methods, DeepHuman and PIFu. Among them, PIFu uses deep implicit functions as geometric representation, DeepHuman combines volume representation with SMPL model. We compare with PIFu in Figure 5, PIFu struggles to reconstruct model in challenging pose, while also suffering from self-occlusion. Unlike these methods, our method is able to perform in challenging body poses. Our method outperforms these methods in terms of surface quality and pose generalization ability.

The results of the comparison are shown in Table 3, and the quantitative comparison shows that our method outperforms the methods of Deephuman and PIFu in terms of surface reconstruction accuracy. Overall, our method is more general, more robust and more accurate than DeepHuman and PIFu.

	PSD (cm)	Chamfer (cm)	Normal (cm)
Deephuman [9]	11.246	11.928	0.464
PIFu [6]	4.026	2.604	0.300
Ours	2.100	2.288	0.281

Table 3. Numerical comparison results

5 CONCLUSION

Accurately and robustly reconstructing a 3D human body from a single RGB image is a challenging problem due to the diversity of body movements, clothing types, and other factors. We propose PointHuman to fuse feature of pixel feature, spatial feature and offset depth values implements single-view human mesh reconstruction. Our construction method addresses both spatial priors and deep blurring. The key idea behind our approach to overcome these challenges is to decompose the pose estimation from the surface reconstruction. To this end, we provide a deep-learning based framework that combines the point cloud form of a parametric SMPL model with a non-parametric deep implicit function for reconstructing a 3D human body model from a single RGB image. Our method performs well in terms of robustness and surface detail. For very complex poses and very loose clothing, our method cannot generate reasonable human bodies. Therefore, although the proposed method has taken a step forward in terms of generalization ability, it still

fails in the case of extremely challenging poses. Point Transformer has limited ability to extract spatial information from parametric models and cannot extract spatial information from extremely complex poses. For the invisible area, our PointHuman can only predict a plausible result while can not guarantee its accuracy. The network for spatial feature extraction can be improved, or multi-view reconstruction can be used, so that the reconstructed human body is better. An important future direction is to alleviate the reliance on ground truth and save costs by exploring large-scale image dataset and video dataset for unsupervised training. Additionally, we can consider combining semantic segmentation for reconstruction to solve the problem of not being able to reconstruct loose clothes.

Acknowledgements

This work was supported by Shenzhen Science and Technology Projects under Grant JCYJ20200109143035495 and Natural Science Foundation of Fujian Province (2022J011275).

REFERENCES

[1] Loper, M.—Mahmood, N.—Romero, J.—Pons-Moll, G.—Black, M. J.: SMPL: A Skinned Multi-Person Linear Model. ACM Transactions on Graphics (ToG), Vol. 34, 2015, No. 6, pp. 1–16, doi: 10.1145/2816795.2818013.

[2] Alldieck, T.—Magnor, M.—Xu, W.—Theobalt, C.—Pons-Moll, G.: Video Based Reconstruction of 3D People Models. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8387–8397, doi: 10.17863/CAM.85609.

[3] Alldieck, T.—Pons-Moll, G.—Theobalt, C.—Magnor, M.: Tex2shape: Detailed Full Human Body Geometry from a Single Image. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2293–2303, doi: 10.1109/ICCV.2019.00238.

[4] Feng, Y.—Choutas, V.—Bolkart, T.—Tzionas, D.—Black, M. J.: Collaborative Regression of Expressive Bodies Using Moderation. 2021 International Conference on 3D Vision (3DV), IEEE, 2021, pp. 792–804, doi: 10.1109/3DV53792.2021.00088.

[5] Kolotouros, N.—Pavlakos, G.—Black, M. J.—Daniilidis, K.: Learning to Reconstruct 3D Human Pose and Shape via Model-Fitting in the Loop. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2252–2261, doi: 10.1109/ICCV.2019.00234.

[6] Saito, S.—Huang, Z.—Natsume, R.—Morishima, S.—Kanazawa, A.—Li, H.: Pifu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2304–2314, doi: 10.1109/ICCV.2019.00239.

[7] Choutas, V.—Pavlakos, G.—Bolkart, T.—Tzionas, D.—Black, M. J.: Monocular Expressive Body Regression Through Body-Driven Attention. European Conference on Computer Vision, Springer, 2020, pp. 20–40, doi: 10.1007/978-3-030-58607-2_2.

[8] Pons-Moll, G.—Pujades, S.—Hu, S.—Black, M. J.: Clothcap: Seamless 4d Clothing Capture and Retargeting. ACM Transactions on Graphics (ToG), Vol. 36, 2017, No. 4, pp. 1–15, doi: 10.1145/3072959.3073711.

[9] Zheng, Z.—Yu, T.—Wei, Y.—Dai, Q.—Liu, Y.: Deephuman: 3D Human Reconstruction from a Single Image. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 7739–7749, doi: 10.1109/ICCV.2019.00783.

[10] Sitzmann, V.—Martel, J.—Bergman, A.—Lindell, D.—Wetzstein, G.: Implicit Neural Representations with Periodic Activation Functions. Advances in Neural Information Processing Systems, Vol. 33, 2020, pp. 7462–7473.

[11] Saito, S.—Simon, T.—Saragih, J.—Joo, H.: Pifuhd: Multi-Level Pixel-Aligned Implicit Function for High-Resolution 3D Human Digitization. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 84–93, doi: 10.1109/CVPR42600.2020.00016.

[12] Hong, Y.—Zhang, J.—Jiang, B.—Guo, Y.—Liu, L.—Bao, H.: Stereopifu: Depth Aware Clothed Human Digitization via Stereo Vision. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 535–545, doi: 10.1109/CVPR46437.2021.00060.

[13] Lassner, C.—Romero, J.—Kiefel, M.—Bogo, F.—Black, M. J.—Gehler, P. V.: Unite the People: Closing the Loop Between 3D and 2D Human Representations. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6050–6059, doi: 10.1109/CVPR.2017.500.

[14] Pavlakos, G.—Zhu, L.—Zhou, X.—Daniilidis, K.: Learning to Estimate 3D Human Pose and Shape from a Single Color Image. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 459–468, doi: 10.1109/CVPR.2018.00055.

[15] Sun, Y.—Ye, Y.—Liu, W.—Gao, W.—Fu, Y.—Mei, T.: Human Mesh Recovery from Monocular Images via a Skeleton-Disentangled Representation. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 5349–5358, doi: 10.1109/ICCV.2019.00545.

[16] Güler, R. A.—Neverova, N.—Kokkinos, I.: Densepose: Dense Human Pose Estimation in the Wild. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7297–7306, doi: 10.1109/CVPR.2018.00762.

[17] Rockwell, C.—Fouhey, D. F.: Full-Body Awareness from Partial Observations. European Conference on Computer Vision, Springer, 2020, pp. 522–539, doi: 10.1007/978-3-030-58520-4_31.

[18] Zhang, Y.—Hassan, M.—Neumann, H.—Black, M. J.—Tang, S.: Generating 3D People in Scenes Without People. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6194–6204, doi: 10.1109/CVPR42600.2020.00623.

[19] Arnab, A.—Doersch, C.—Zisserman, A.: Exploiting Temporal Context for

3D Human Pose Estimation in the Wild. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3395–3404, doi: 10.1109/CVPR.2019.00351.

[20] Kanazawa, A.—Zhang, J. Y.—Felsen, P.—Malik, J.: Learning 3D Human Dynamics from Video. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5614–5623.

[21] Kocabas, M.—Athanasiou, N.—Black, M. J.: Vibe: Video Inference for Human Body Pose and Shape Estimation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5253–5263, doi: 10.1109/CVPR42600.2020.00530.

[22] Zhang, C.—Pujades, S.—Black, M. J.—Pons-Moll, G.: Detailed, Accurate, Human Shape Estimation from Clothed 3D Scan Sequences. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4191–4200, doi: 10.1109/CVPR.2017.582.

[23] Bhatnagar, B. L.—Sminchisescu, C.—Theobalt, C.—Pons-Moll, G.: Loopreg: Self-Supervised Learning of Implicit Surface Correspondences, Pose and Shape for 3D Human Mesh Registration. Advances in Neural Information Processing Systems, Vol. 33, 2020, pp. 12909–12922.

[24] Alldieck, T.—Magnor, M.—Bhatnagar, B. L.—Theobalt, C.—Pons-Moll, G.: Learning to Reconstruct People in Clothing from a Single RGB Camera. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 1175–1186, doi: 10.1109/CVPR.2019.00127.

[25] Bhatnagar, B. L.—Tiwari, G.—Theobalt, C.—Pons-Moll, G.: Multi-Garment Net: Learning to Dress 3D People from Images. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 5420–5430, doi: 10.1109/ICCV.2019.00552.

[26] Weng, C. Y.—Curless, B.—Kemelmacher-Shlizerman, I.: Photo Wake-Up: 3D Character Animation from a Single Photo. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5908–5917, doi: 10.1109/CVPR.2019.00606.

[27] Jackson, A. S.—Manafas, C.—Tzimiropoulos, G.: 3D Human Body Reconstruction from a Single Image via Volumetric Regression. Proceedings of the European Conference on Computer Vision (ECCV) Workshops, 2018, pp. 64–77, doi: 10.1007/978-3-030-11018-5_6.

[28] Huang, Z.—Xu, Y.—Lassner, C.—Li, H.—Tung, T.: Arch: Animatable Reconstruction of Clothed Humans. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 3093–3102, doi: 10.1109/CVPR42600.2020.00316.

[29] He, T.—Xu, Y.—Saito, S.—Soatto, S.—Tung, T.: ARCH++: Animation-Ready Clothed Human Reconstruction Revisited. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 11046–11056, doi: 10.1109/ICCV48922.2021.01086.

[30] Xiu, Y.—Yang, J.—Tzionas, D.—Black, M. J.: Icon: Implicit Clothed Humans Obtained from Normals. Proceedings of the IEEE Conference on

Computer Vision and Pattern Recognition (CVPR), Vol. 2, 2022, doi: 10.1109/CVPR52688.2022.01294.

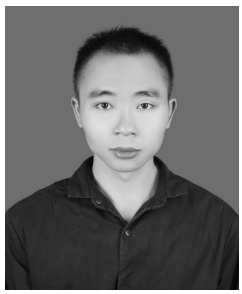
[31] Zhao, H.—Jiang, L.—Jia, J.—Torr, P. H.—Koltun, V.: Point Transformer. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 16259–16268.

[32] Ronneberger, O.—Fischer, P.—Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015, pp. 234–241.

[33] Targ, S.—Almeida, D.—Lyman, K.: Resnet in Resnet: Generalizing Residual Architectures. Arxiv Preprint Arxiv:1603.08029, 2016.

[34] Sun, Y.—Bao, Q.—Liu, W.—Fu, Y.—Michael J., B.—Mei, T.: Monocular, One-Stage, Regression of Multiple 3D People. ICCV, 2021, doi: 10.1109/ICCV48922.2021.01099.

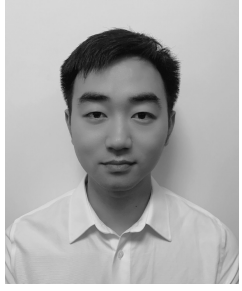
[35] Yu, C.—Xiao, B.—Gao, C.—Yuan, L.—Zhang, L.—Sang, N.—Wang, J.: Lite-Hrnet: A Lightweight High-Resolution Network. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 10440–10450, doi: 10.1109/CVPR46437.2021.01030.



Zongguo Mo is pursuing a graduate degree from the Department of Computer Science and Technology, at Xiamen University, Fujian, China. His research interests include deep learning, human reconstruction and human pose estimation.

Qicong Wang received his Ph.D. in information and communication engineering from Zhejiang University, Hangzhou, China. He is currently Associate Professor at the Department of Computer Science and Technology, Xiamen University, Xiamen, China. His research interests include computer vision, machine learning, and big data analytics.

Hua Shi is a lecturer at the School of Optoelectronic and Communication Engineering, Xiamen University of Technology in China. He received his Ph.D. from the Xiamen University, P.R. China in 2014. His research is in the areas of machine learning, computer vision, and artificial intelligence.



Baobing Zhang received his Ph.D. degree in artificial intelligence from the Brunel University London, UK in 2020. He is currently Post-Doctoral Research Fellow at the Brunel University London. His research interests include deep learning, computer vision, image processing, data privacy, and AI applications.

Wanxin Sui is currently Ph.D. candidate at the Brunel University London, UK. Her research interests are in the areas of data protection and privacy, privacy-preserving AI techniques, and AI applications in higher education.

Computing and Informatics, Vol. 42, 2023, 480–500, doi: 10.31577/cai_2023_2_480

ONTOLOGY FOR BLIND SQL INJECTION

Jean Rosemond Dora, Ladislav Hluchý

Institute of Informatics
Slovak Academy of Sciences
Dúbravská cesta 9
845 07 Bratislava, Slovakia
e-mail: {jeanrosemond.dora, Ladislav.Hluchy}@savba.sk

Karol Nemoga

Institute of Mathematics
Slovak Academy of Sciences
Štefánikova 49
811 04 Bratislava, Slovakia
e-mail: nemoga@mat.savba.sk

Abstract. In cyberspace, there exists a prevalent problem that heavily occurs to web application databases and that is the exploitation of websites by using SQL injection attacks. This kind of attack becomes more difficult when it comes to blind SQL vulnerabilities. In this paper, we will first make use of this vulnerability, and subsequently, we will build an ontology (OBSQL) to address the detection of the blind SQL weakness. Therefore, to achieve the exploitation, we reproduce the attacks against a website in production mode. We primarily detect the presence of the vulnerability, after we use our tools to abuse it. Last but not least, we prove the importance of applying ontology in cybersecurity for this matter. The mitigation techniques in our ontology will be addressed in our future work.

Keywords: SQL injection, blind SQL, vulnerability, weakness, ontology, semantic web, information security, cyber threats, website security, web application vulnerabilities, attack detection

1 INTRODUCTION

The significance of the website at present, and its constant use, make it a niche for evildoers to obtain confidential data of users. According to recent research from `https://www.verizon.com/business/en-gb/resources/reports/dbir/` (2022), web application attacks are involved in 26 % of all breaches, making the second most common attack pattern. Knowing this information is extremely crucial for both the attackers and the IT security engineers, as it involves security concerns.

The detection of the blind SQL injection is a high-level scenario. It involves advanced tools and techniques to be used by the attacker. First and foremost, a web application developer does not mean to be a cyber security analyst; therefore, it may be hard for him to detect vulnerabilities. Moreover, during the programming phase of the web application, the may use some common techniques in PHP, Java, Python, etc. to mitigate SQL vulnerability. However, detecting the blind SQL injection will require a deep analysis of the codes and the use of penetration testing tools. Manually, it is very time-consuming to detect the vulnerability and attack its corresponding database. One of the reasons is that the injection payloads that should be used differ from a database to a database. For example, we need at least two (2) different payloads to abuse SQLite3 and MySQL databases.

Regarding this web application vulnerability, an important factor that requires great attention is when it accepts user input. For example, if it accepts users to be registered, submit, log in, comment, etc. Publishing the website without submitting it to a penetration testing phase, will be presented to attackers like a doorless house for thieves. Additionally, it is also vital to expand the awareness of the need to comprehensively evaluate what kinds of information the web application will reveal as output when a request is made, and which mechanisms we use to break down the request to keep up the whole security framework high.

To approach this idea, we resort to the methodology that necessitates knowing the followings:

- what kind of response,
- who utilizes,
- for what purposes,
- if this, then that.

Hence, a question may be risen: "How can we use semantic languages, (semantic axioms and rules) to help us understand the structure of a possible vulnerability?" Thus, comes the importance of ontology.

Briefly, an ontology is a well-structured diagram consisting of a tree of classes (sub-classes) or simply classes inheritance, attributes, and relationships. The construction of an ontology relies on the establishment of rules.

The rest of this paper is then organized as follows: Section 2 provides the definition and impact of the Blind SQL injection. It also summarizes some related works. Sections 3 and 4 embrace the attack scenario, the detection, and the exploitation of the vulnerability. Section 5 provides information on the usability of ontology in cybersecurity, the semantic web, axioms, and rules implementation. Section 6 provides our future work and concludes the paper.

Note: *For privacy reasons, we are obliged to hide some confidential data in the figures (public address and more), since it is a live website.*

2 DESCRIPTION AND IMPACT OF BLIND SQL INJECTION

Here, we will describe SQL injection from two angles:

1) **SQL vulnerabilities** – It is when a web page suffers from SQL weakness. This kind of vulnerability cannot be detected by simply reading the website source code only, most of the time it has to go through a testing phase.

   SQL injection is a code injection method that may destroy the database of a web application if wrongly used. Imagine a scenario where a bank stores all its clients' data in a specific database. Having access to that database can help the attacker find relevant and sensitive information, such as clients' names, hashed or plaintext passwords, telephone numbers, home addresses, signed contracts, etc. We also have to note that such companies have a lot of databases on their systems. Thus, manually exploiting this vulnerability can be problematic, as the attacker may have no clues about the name of the database management system (DBMS), databases name, tables name, and columns name. Utilizing automatic tools is more appropriate for this particular detection and exploitation of the vulnerability. It helps us reduce the time of the attack since all the requests will produce latency of the target server.

   Usually, this attack works by inserting malicious code in an SQL statement. Wherever there are some parameters, then it is possible to inject any payload for detection purposes. SQL injection is one of the most common web application vulnerabilities on the OWASP checklist.

2) **SQL attacks** – commonly known as SQL injection, it is when the SQL vulnerability of a web page is being exploited by an attacker, or by a penetration tester.

   Therefore, blind SQL injection arises when a web application is vulnerable to SQL injection, but its HTTP responses do not include the results of the relevant SQL query or the information of any database errors.

   With this type of vulnerability, many techniques such as UNION attacks are not efficient, since they rely on being able to see the results of the injected query within the website's responses.

However, it is still possible to exploit the blind SQL injection to access unauthorized data, but different techniques must be applied. From a boolean operation case, it asks the database true or false questions and determines the answer based on the response of the application. This attack scenario is often used when the website is configured to show generic error messages but has not diminished the code that is vulnerable to SQL injection.

From a latency viewpoint, the payload request aims to slow down the time the server takes to respond to the query. That being said, an attacker can double-check how the server reacts in time when a payload is injected.

The impact of exploiting the SQL vulnerabilities is greatly significant since the attacker can steal confidential data from the database of the web application (username, table name, user passwords, etc.). For more information, please the following related works: [1, 2, 3, 4, 5, 6, 7, 8].

In the following section of the practical part (Detection & Exploitation), we had to force the server response to slow down to detect the presence of the blind SQL vulnerability from the target website.

3 DETECTION OF BLIND SQL VULNERABILITIES

Detecting the blind SQL weakness from a website can be difficult using the manual inputs (payloads) method to a query field. Sometimes, it requires to the attacker hundreds of payloads to inject into the user-input field of the web page. Saying so, attempting to inject commands one by one by an individual is drastically not a good practice. Therefore, hackers or penetration testers usually resort to manual tools or some automated tools to achieve this goal.

By injecting some characters into the user input of the web page, we found a table name "X". And the text on the page provides more information about another table name "Y" and its column "Y1". However, some of this data was dummy, fake. We had to find a way somehow to abuse what we have obtained from the response. From the following source (*please see* `https://hackersonlineclub.com/sql-injection-cheatsheet/`), We realized that the present blind SQL is a "Generic Time-Based SQL Injection" by invoking the ASCII char. When we used 500000000/1 (please see below), the server response comes up automatically. But using /2, it takes 2 seconds to pop up.

Next, by modifying the payload, we forced the server to give us a response at the time of our choice.

We have seen clearly how we were able to detect the presence of this vulnerability. The next section will demonstrate how we were able to exploit it.

4 EXPLOITATION OF BLIND SQL VULNERABILITIES

The attempt to exploit a system is usually the subsequent action after detecting the presence of a vulnerability. To proceed with the exploitation, we first double-

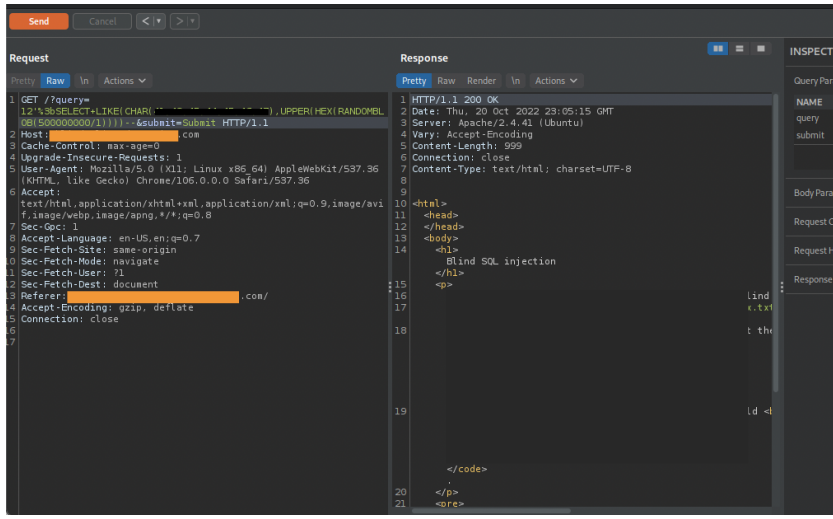


Figure 1. No latency has occurred from the server response

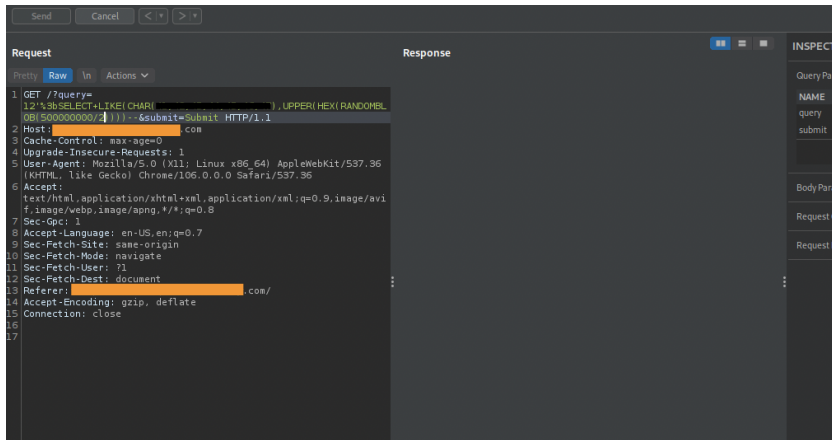


Figure 2. 2s of latency has been occurred from the server response

checked if the target web application is behind a firewall. We also made sure that we had full right to do anything we want to exploit the vulnerability from that target website. As we can see below, the command we used is very aggressive and dangerous as the "–risk 3" and "–level 4" might disrupt or erase the target database.

We realized that the target was behind a firewall, so we had to find a way to bypass this type of protection. To do so, we used "–tamper=space2comment" command.

Figure 3. Bypassing firewall

When everything is set up properly (your full right to attack the target, your tools), then it is time to launch the attack. The attack vigorously sends multiple requests to the website server trying to obtain the exact database management system (DBMS) it uses. As we can see in the following figure, we were able to extract the database name, the tables name, and the rows and columns.

For demonstration purposes, we have chosen our target website for which we already know the number of its databases and which do not contain hundreds of tables. It helps avoid time-consuming attacks.

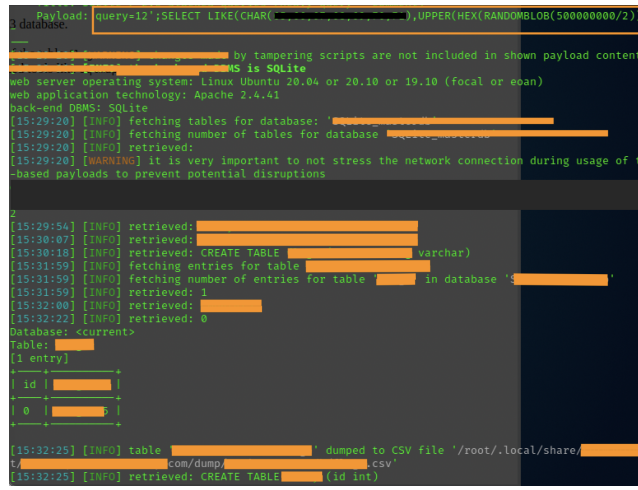


Figure 4. Successful blind SQL attack of the target website database

We have seen how we successfully hacked into a web application and gain information from its database system. Now, let us use the ontology approach to see how it can help when it comes to cybersecurity.

5 NOVEL APPROACH FOR THE DETECTION OF SQL INJECTION ATTACKS (ONTOLOGY)

From the previous sections, we have demonstrated a few examples of how SQL injection (blind) weaknesses can be detected by an attacker (or any individual, penTester for example). We have also seen how he can make use of those vulnerabilities by injecting some payloads to jeopardize the target system. Therefore, is extremely important thus imperative to fight against the adversary by implementing

significant methodologies (approaches and steps) to strengthen security and mitigate the attacks. The term ontology approach is a powerful mechanism which we can start with. For more information about other ontological approaches, please see [9, 10].

Usually, the word ontology can be defined as a formal and explicit specification of a set of concepts in a specific field of interest. The clear specification of those concepts is usually presented in a shape of a well-structured scheme composed of classes inheritance and sub-classes, relationships, and attributes.

5.1 Ontology and Semantic Web

Ontology can be designed to facilitate data to be shared and reused across multiple applications, institutions, organizations, and so on. Based on the field of interest, security experts can use ontology to enhance their systems. In medicine, for instance, IT security engineers can use ontology for pregnancy, covid-19, diabetes, Alzheimer's etc. Please see [11, 12] for some related works.

To apply the concept of ontology in a field, some components should be put into question. The typical ontology components are:

Categories: concepts, i.e., types of objects;

Individuals: situations or things (in this case, individuals are also known as "first-order objects");

Relationships: ways in which individuals and groups can communicate;

Limitations (Constraints): The formal and steady description of what must be true until some inputs are accepted;

Features: classes, properties, aspects, parameters, or instances that objects and categories can contain;

Axioms: assertions, or statements in a logical and understandable form that form together with the perceivable theory that is illustrated and demonstrated by the ontology in their domains.

Before we dive into the construction of our ontological approach, let us first define its importance.

5.1.1 The Reasons of Implementing Ontological Approach in Cyberspace

Ontology is an exciting approach for linking up the description of a data model and the related rules into one application. Ontologies developed in Web Ontology Language (OWL) acquire many benefits afforded by the semantic web stack. The goal of OWL is to represent complex knowledge of entities in a domain through a logic-based language, via a computational, such that the knowledge encapsulated can be ascertained for consistency or utilized as a basis for inferences on that specific knowledge.

- To share a comprehensive structure of data, and information between people. The ontology also allows the reuse of domain knowledge.
- To split domain knowledge from operational knowledge.
- To make domain assumptions obvious.
- To carefully analyze domain knowledge.

  It is good practice to install and configure or use proactive detection tools. Many web-based detection tools are reactive, i.e., they function according to the specific rules set by the administrator.

  The attack can only be prevented if the exact signature of the attack is not only recognized by the scanning tools but also present.

  – It is easy for a malicious entity to launch an attack altering the signature since the majority of the existing techniques are signature-based, which hold the syntax of the attack.
  – Additionally, statistical mechanisms used in Intrusion Detection Systems (IDS) largely provide an attainable solution for the network layer. However, this solution is not efficient at the application layer since it focuses on the character distribution of the input and does not take into account its contextual nature.

5.1.2 Ontology Model – Communication Protocols

The communication protocols, as its name says, allow the transfer of messages from one point to another. It is shaped as semantic networks. The essential part of this activity relies on the "Protocol" concept, which can be classified as the main class of the following sub-classes *FTP*, *SMTP*, *HTTPS*, *HTTP*. This classification subsequently involves three (3) other concepts: *Message*, *Request*, *Response*.

One of the finest benefits of the ontology approach is that it comes up with inference potentiality and the required constructs that enable software systems to reason over the knowledge base.

The following example will produce a response latency of two (2) seconds from the web server response in the attacker's environment, (taken from Figure 2):

query=12';SELECT LIKE(CHAR(22,23,...,28,29),UPPER(HEX(RANDOM BLOB(500000000/2))))–

To illustrate the inference activity and flexibility in semantic rules, the query string carries the detection payloads which forces the web server to respect its request. Instead of inserting a single parameter in the user-input field, (12 for example), we added a SELECT + LIKE of a RANDOMBLOB command there to experience the latency.

The referrer is in line 13, the request does not involve any cookies. All the other lines from the "Request" tab are irrelevant to us. The inference of the ontology

yields all the numerous activities using a general semantic rule. Generally, the rules give a focal point if the malicious payload infects the parameter values. Additionally, the rules describe the inference structure through transitive features.

5.2 Implementation of Rules

By applying the semantic concept, we can use deductive inference rules to reason on a piece of HTTP well-constructed diagram.

Let us describe to which class hierarchy each method and protocol are belonging:

- GET $\sqsubset$ Method,
- POST $\sqsubset$ Method,
- HTTP $\sqsubset$ Protocol,
- SQL injection attacks $\sqsubset$ Attack,
- Request Header $\sqcap$ Response Header $\equiv \perp$,
- POST $\sqcap$ GET $\equiv \perp$.

In our proposed approach, all subsume ($\sqsubset$) relations are *transitive, irreflexive* and *asymmetric*. But the equivalence ($\equiv$) relations are *reflexive, symmetric* and *transitive*. Likewise, no conceptually disjoint ($\sqcap$) relations contravene its properties of *symmetric, reflexive* and *transitive*. We established these rules based on how we were able to detect the SQL injection vulnerabilities, then applied them to our ontology.

Rule 1: *Person(?P)* $\sqcap$ *hasTools(?P, ?Q* $\rightarrow$ *Attacker(?P)*(Transitivity),

Rule 2: *SubClassOf(?P, ?Q)* $\sqcap$ *typeOf(?n,?P)* $\rightarrow$ *typeOf(?n, ?Q)*(Transitivity),

Rule 3: *hasPartOf(?P, ?Q)* $\sqcap$ *hasPartOf(?Q, ?n)* $\rightarrow$ *hasPartOf(?P, ?n)*(Transitivity),

Rule 4: *contains(?P, ?Q)* $\sqcap$ *contains(?Q, ?n)* $\rightarrow$ *contains(?P, ?n)*(Transitivity).

From rules 3 and 4, the 5^{th} becomes:

Rule 5: *hasPartOf(?P, ?Q)* $\sqcap$ *contains(?Q, ?n)* $\rightarrow$ *contains(?P, ?n)*(Transitivity),

Rule 6: *Attacker(?P)* $\sqcap$ *hasInput(?P, ?Q)* $\sqcap$ *hasPartOf(?webAp, ?HTML)* $\sqcap$ *contains(?query, ?method)* $\sqcap$ *contains(?method, ?param)* $\sqcap$ $\exists$*Vulnerability(?webAp, ?v)* $\sqcap$ *is_sentBy(?P, ?a)* $\rightarrow$ *is_detectedBy(?a, ?v)* (Drived),

Rule 7: *IF Rule 6*$\neg$*is_detectedBy(?a, ?v)* $\rightarrow$ *continue(?Q, ?a2)*(Drived).

Rule 7 becomes:
Attacker(?P) $\wedge$ hasInput(?P, ?Q) $\wedge$ hasPartOf(?webAp, ?HTML) $\wedge$ contains(?query, ?method) $\wedge$ contains(?method, ?param) $\wedge$ hasVulnerability(?webAp, ?v) $\wedge$ is_sentBy(?P, ?a) $\wedge$ notFound(?response, ?v) $\wedge$ Payload(?a2) $\rightarrow$ continue(?Q, ?a2)).

Rule 8: *IF Rule 6 is_detectedBy(?a,?v) OR Rule 7 is_detectedBy(?a2, ?v)* $\sqcap$ $\exists$*Vulnerability(?webAp, ?v)* $\sqcap$ *infectedBy(?param, ?a) OR infectedBy(?param, ?a2)* $\rightarrow$ *exploitedBy(?P, ?v)*(Drived).

Interpretation of the rules

- The first rule is a basic rule that states that if someone has some tools (Kali, Metasploit, maliciousPayload, ...), and uses them illegally, then that person is an attacker.
- Rule number 2, indicates that if class P is a sub-class of Q, then each instance of class P also belongs to class Q. For example: if the "Tools" class is a subclass of "Technology", then every instance (browsers, Kali Linux, Metasploit, ...) of the Tool class also belongs to the Technology class. The similar paradigm for the rule 3.
- This Rule 4, basically indicates that if the request contains a malicious string, and that, the malicious string contains a parameter value, then the request also contains that parameter value.
- Rule 5: The HTTP Request has part Referer, and the Referer contains the payload, then the HTTP Request also contains the payload.
- Rule 6: The HTML webpage allows user input. The attacker uses his method built with a parameter of his choice to query the request. If the server responds with a latency defined by the payload, then the vulnerability will be detected by the attacker.
- Rule 7: If the server is not responding (the request does not produce any latency), it does not mean that the web application is not vulnerable. Therefore, continue the attack process.
- Rule 8: If the response produces latency (from rule 6), then through the inference process the vulnerability will be possibly exploited by the attacker using some attack vectors.

5.3 Transformation of SWRL Rules to OWL Axioms

We present a theoretical idea applied to our ontology. Let **E**, **F**, **G** and **H** be some pairwise disjoint, infinite sets of *classes*, *sub-classes*, *properties* (Object and Data), *individuals* and *variables* where $\top$, $\bot \in \mathbf{E}$; the *universal property* U $\in \mathbf{F}$ i.e., **owl:topObjectProperty**. A *class expression* is an element of the following grammar I ::= (I $\sqcap$ I | $\exists$**F**.I | $\exists$.Self | **E** | {a} where $\mathbf{E} \in \mathbf{E}$, $\mathbf{F} \in \mathbf{F}$ and a $\in \mathbf{G}$.

Let us now resort to the definition of what an *axiom* is: it is a formula of the form E $\sqsubseteq$ K or $F_1 \circ \cdots \circ F_n \sqsubseteq F$ with E, K $\in$ I and $E_(i) \in \mathbf{F}$. A *rule* is a first-order logic formula ordinarily of the form $\forall \mathbf{p}(\beta(\mathrm{x}) \rightarrow \eta(\mathbf{q}))$ with β and η conjunctions of atoms; and **p**, **q** are subsequently non-empty sets of terms where $\mathbf{p} \subseteq \mathbf{q}$. *Rules* and *Axioms* expressions are very significant in building an ontology. Furthermore, they are referred to as *logical formulas*. Axioms correspond to OWL2 whereas rules correspond to SWRL.

Consider some terms w and z and a conjunction of atoms β. We say these two terms w and z are directly connected, or joined in β if they occur in the same atom in β. We say w and z are connected in β if there is some sequence of terms w_1, $\ldots$, w_k with $w_1 = w$, $w_k = z$, and w_{i-1} and w_i are directly linked in β for every $i = 2, \ldots, k$.

Additionally, for rules *rules* of the form $\beta \rightarrow \eta$; there exists an interpretation *it* which *entails rules*. Therefore, for every substitution *subst*, we have that *it*, *subst* $\models \beta$ implies *it*, *rules* $\models \eta$. That means, the semantics of rules here follows analogous standard semantics of the first-order predicate logic. From the same perspective, we say that two groupings of logical formulas S and S' are equivalent if and only if each interpretation *it* that calls for S and S' are *equivalent* ($S \equiv S'$) and vice-versa.

On the same current of idea, S' is a conservative extension of S if and only if: Every interpretation that calls for S' also calls for S.

Each interpretation that entails, calls for S' is only expressed for the symbols in S can be extended to an interpretation calling for S' by adding appropriate interpretations for further signature symbols. Normally, all the variables in the body of a rule are connected. If two (2) variables for example (a, b) are not linked with the body of a rule, then we could simply append the atom $\mathrm{U}(a, b)$ to the body of the rule resulting in a semantically $\equiv$ rule.

Using a fundamental example in the ontology can help us explain the transformation of rules into an axiom.

Example 1. Consider the rule Γ = Person(p) $\wedge$ hasChild(c, c') $\wedge$ Female(c') $\rightarrow$ Daughter(c'). The following sequence of rules can be produced as follows:

$(\exists \mathrm{hasChild.Person})(\mathrm{c'}) \wedge \mathrm{Female(c')} \rightarrow \mathrm{Daughter(c')}$
$(\exists \mathrm{hasChild.Person} \sqcap \mathrm{Female})(\mathrm{c'}) \rightarrow \mathrm{Daughter(c')}$

Rule Δ_Γ from the above example can be now transformed into an axiom as stated in the following lemma.

Lemma 1. Consider some rule Γ. If Δ_Γ is of the form $\mathrm{A(p)} \rightarrow \mathrm{B(p)}$, then Γ is equivalent to the axiom $\mathrm{A} \sqsubseteq \mathrm{B}$.

Since the equivalence relation is transitive, the rule Γ is equivalent to the axiom $\exists \mathrm{hasTools.Person} \sqcap \mathrm{Attacks} \sqsubseteq \mathrm{Attacker}$.

Proof. Let r and r' be some rules such that r' results by using some of the transformations (as in the previous example) to r. By definition, we can conclude that there is an equivalency between r and r'. We can also demonstrate through induction that Γ is equivalent to Δ_Γ. Additionally, if δ $(\alpha \rightarrow \gamma)$ is of form $\mathrm{E(p)} \rightarrow \mathrm{F(p)}$, then by the definition of the semantics of rules and axioms, $\mathrm{E} \sqsubseteq \mathrm{F}$ is $\equiv$ to δ $(\alpha \rightarrow \gamma)$. Therefore, since the equivalence $(\equiv)$ relation is transitive, then we can safely say that γ is $\equiv$ to $\mathrm{E} \sqsubseteq \mathrm{F}$. $\square$

Lemma 2. Furthermore, let us consider some rule Gamma (Γ). If Δ_Γ is of the form $\bigwedge_{t=2}^{m}(A_t(x_{t-1})) \wedge R_t(x_{t-1}, x_t) \wedge A_n(x_n) \rightarrow \mathrm{G}(x_1, x_n)$, then the group of axioms $A_t \sqsubseteq \exists R_{A_t}.\mathrm{Self} \mid t = 1, \ldots, m\} \cup \{R_{A_t} \circ R_1 \circ \cdots \circ R_{A_{m-1}} \circ R_m \circ R_{A_m} \sqsubseteq \mathrm{G}\}$ where all R_{A_t} are the properties unique for each class A_t is conservative extension of the rule Γ.

Proof. As illustrated in Lemma 1, rules Γ and Δ_Γ are equivalent. Thus, the lemma which follows the set of rules presented in the statement of the lemma is a conservative extension of Γ. See the following figure to see the preprocessing axiom generated in Protégé in the ROWLTAB plugin. □

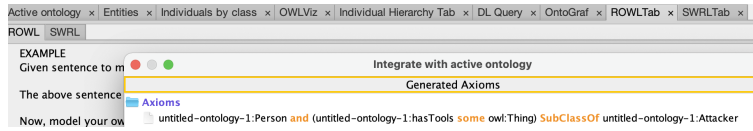


Figure 5. Rule is being converted to OWL axiom

Before implementing these rules, we had to create instances, data properties, object properties, and individuals to cooperate with the classes, and sub-classes; without that, the ontology will not understand your intention. The below listed the main classes and sub-classes. However, there are a lot of sub-classes that are not listed in the figure.

The figure 8 presents the individuals by class, where we can add "data properties assertion, object properties assertion, description types, etc."

For rational numbers xsd:decimal is of the best practice when using SWRL rules because it is the default for SWRL (Figure 7). When SWRL sees a literal such as 2.0 it draws the inference that the datatype is xsd:decimal. For other data types, you need to explicitly define the datatype as the literal. The property is functional because a Process can only have one value for its slack.

After that, we use the Drools rule engine to apply the rules in Section 5.2 to our ontology. If the rules are matched the properties you have established in the software protégé, then running the program using Pellet or HermiT plugins will generate the inferred classes along with their characteristics.

Note that, in the Protégé application, you can install several plugins to suit your needs, and add them to your tabs. After installing, you can simply go to "Window → Tabs" and select your desired one to add to your project. For more information about the software, here is a practical guide to building OWL ontology `https://www.researchgate.net/publication/351037551_A_Practical_Guide_to_Building_OWL_Ontologies_Using_Protege_55_and_Plugins`. The author very well described the concept of "Description Logic Reasoner to check the consistency of the ontology, data properties". He also introduces the Semantic Web Rule Language (SWRL) and a walk-through of creating SWRL and SQWRL rules.

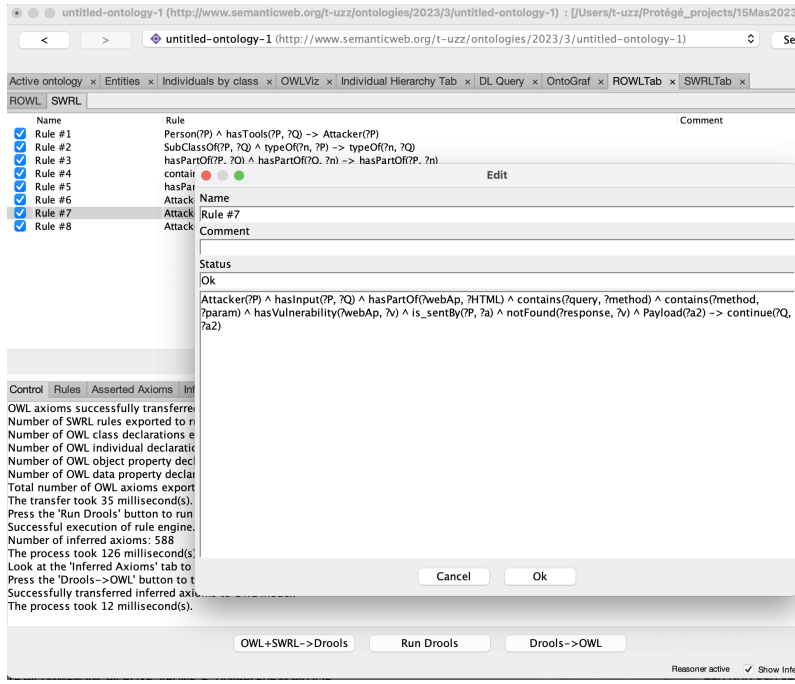


Figure 6. The ROWLTab interface with integrated axioms

We subsequently used the ROWLTab, “ROWL” and “SWRL” tab options to build the rules.

- Clicking on the “OWL + SWRL → Drools” button will transfer SWRL rules and relevant OWL knowledge to the rule engine.
- Likewise, clicking on the “Run Drools” button will run the rule engine.
- Clicking on the “Drools→OWL” button will transfer the inferred rule engine knowledge to OWL knowledge.

The SWRLAPI supports an OWL profile called OWL 2 RL and uses an OWL 2 RL-based reasoner to perform reasoning. An example is given in the following figure.

5.4 Ontology Design

In this section, we define the formalization of the core ontology concepts for SQL injection attacks. First, we introduce the set of terms:

- Term extraction consists of gathering a list of terms together that are relevant for a specific domain of knowledge. This can be done by defining a set of concepts. The properties, relationships, and meaning of concepts should be

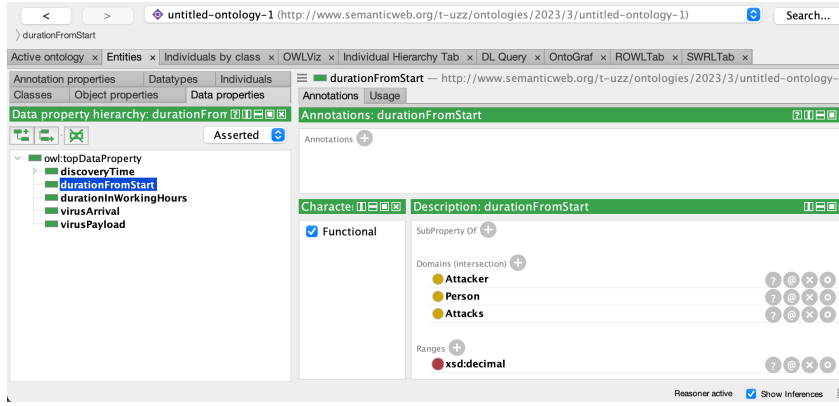


Figure 7. Data properties

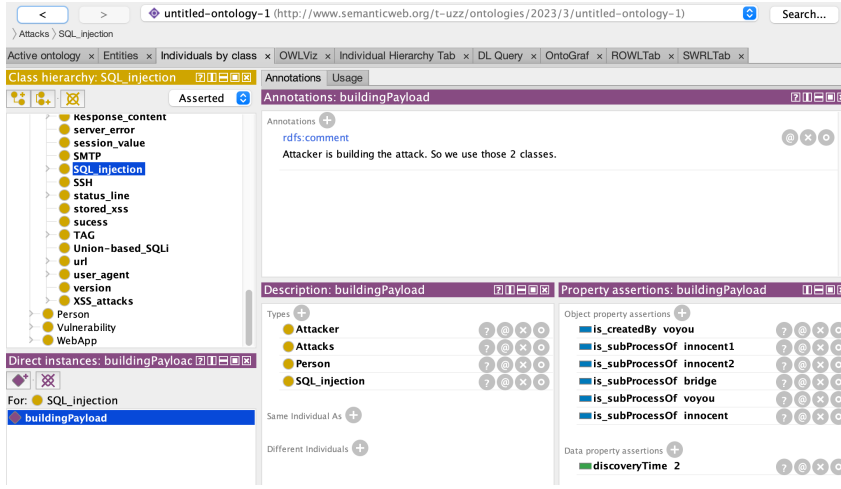


Figure 8. Individuals by class

evaluated before building the class hierarchy. To build our ontology, we make use of the following terms: SQL injection, Blind SQLi, attacks, vulnerability, weakness, attacker, web application, security layer, tools, technology, payloads, victim, exploitation.

- *Modules identification* consists of defining the set of individuals that will comply with the ontology scheme.
- The entities, data properties, object properties, and individuals of the ontology modules are designed using the Description Logics (hence DL) notation.

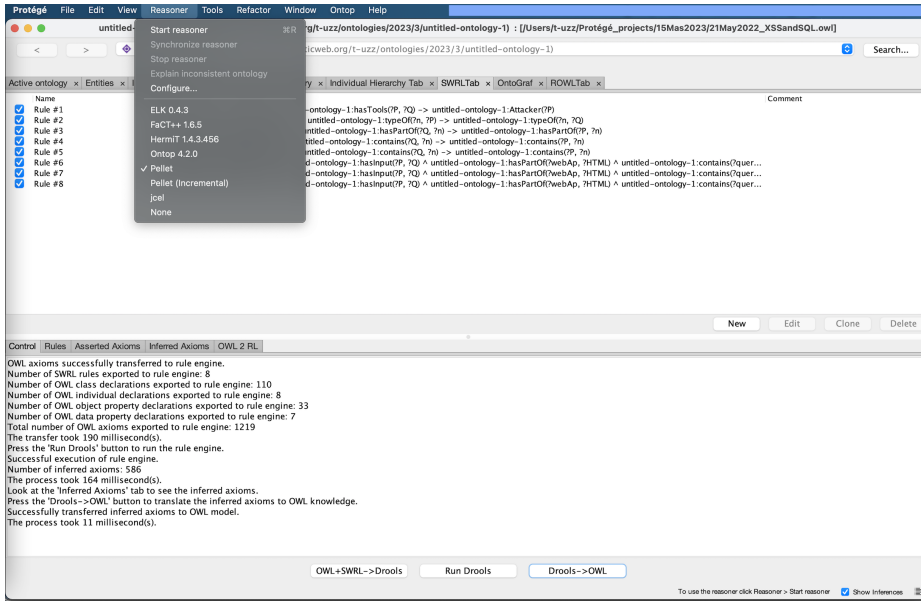


Figure 9. Running Reasoner to establish rules

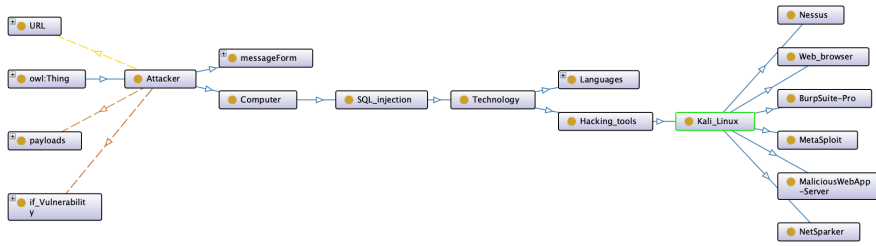


Figure 10. Subclasses of the Attacker ontology

The *Attacker* is a class in our ontology that generates the malicious payload using some technologies to launch the attack against a target victim. This class is further subdivided into several classes.

The following description logic (DL) represents the formal definition of the class *Attacker*.

Our ontology below describes briefly the security layers as a class, but we did not emphasize the mitigation of the SQL attacks. The approach is more related to the detection of the vulnerability. However, to encompass all the important concepts of the attack scenario, we also addressed some mitigation techniques that can be used to reduce these types of attacks.

$$\begin{gathered}\textit{Attacker} \equiv \\ \exists\ \textit{hasTitle.xsd:string} \sqcap \\ \textit{hasTools.xsd:string} \sqcap \textit{typeOf.xsd:string} \sqcap \\ \textit{hasDescription.xsd:string} \sqcap \\ \textit{discoveryTime.xsd:dateTime} \sqcap \\ \textit{discoveryTime.xsd:dateTimeStamp,} \sqcap \textit{isComposedOf(Attacker, Technology),} \\ \forall \textit{isComposedOf.Technology} \sqcap \exists \textit{isVulnerable.xsd:boolean,} \\ \textit{Technology} \subseteq \textit{Attacker,} \\ \textit{Computer} \subseteq \textit{Attacker}\end{gathered}$$

Figure 11. DL of the Attacker ontology

- The sub-class *Security* may include *Firewall*, *IDS*, *IPS* helps the administrator of the website to log and block any malicious-looking activity in the website in real-time such as SQL injections, XSS attacks, etc. The sub-classes *validation_Mechanism* may also include *Filters*, *Sanitization* are meant to be implemented most of the time by the web application developer during the coding process.
- *if_Vulnerability* exists, then a response from the target web server may alert the client (hence, the attacker's web browser). If the alert does not occur with a string response, then mostly it may occur in a form of latency.
- The class *SQLi_Attacks* contains several subclasses and sub-subclasses; it is the main class for the penetration testing phase. This is where all the attempts (SQL malicious payloads) occurred.

6 CONCLUSIONS

From now, our level of thinking about security risks and privacy – specifically about how our confidential data is stored online, should be well-oriented more seriously. In this paper, we briefly talked about the cybersecurity offensive. However, having this kind of knowledge about how users' data can be extracted by attackers abusing SQL injection vulnerabilities leads us into digging into how we can prevent this from happening. We briefly demonstrated through the establishment of semantic rules how we can make use of the ontology to detect vulnerabilities. Due to the required size of this paper, a deeper description will be elaborated on in our future work. Therefore, in our next paper, we will dive into the work performance of our ontology approach for the detection of SQL vulnerabilities and will be more oriented to the mitigation techniques.

Abbreviations and Acronyms

SQLi: SQL injection,

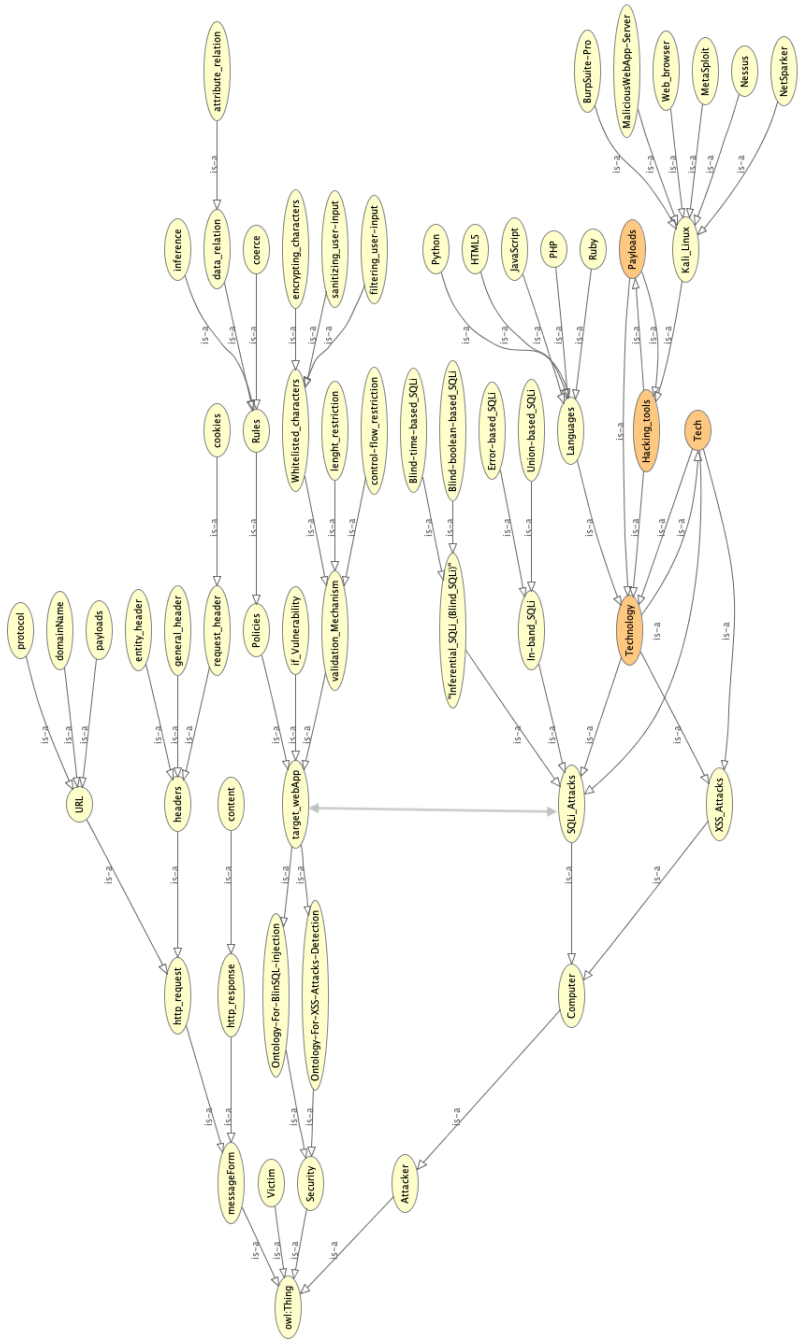


Figure 12. Description of the ontology, generated from OWLViz plugin

DL: Description Logic,

ICS: Industrial Control System,

OWASP: Open Web Application Security Project,

OWL: Web Ontology Language.

Mathematical Symbols

Abstract Syntax	DL Syntax
Class(A partial $C_1 \dots C_n$)	$A \sqsubseteq C_1 \sqcap \dots \sqcap C_n$
Class(A complete $C_1 \dots C_n$)	$A \equiv C_1 \sqcap \dots \sqcap C_n$
EnumeratedClass(A $o_1 \dots o_n$)	$A \equiv \{o_1\} \sqcup \dots \sqcup \{o_n\}$
SubClassOf(C_1 C_2)	$C_1 \sqsubseteq C_2$
EquivalentClasses($C_1 \dots C_n$)	$C_1 \equiv \dots \equiv C_n$
DisjointClasses($C_1 \dots C_n$)	$C_i \sqcap C_j \sqsubseteq \bot, i \neq j$
Datatype(D)	
ObjectProperty(R super(R_1)...super(R_n)	$R \sqsubseteq R_i$
domain(C_1)...domain(C_m)	$\geqslant 1\, R \sqsubseteq C_i$
range(C_1)...range(C_ℓ)	$\top \sqsubseteq \forall R.C_i$
[inverseOf(R_0)]	$R \equiv R_0^-$
[Symmetric]	$R \equiv R^-$
[Functional]	$\top \sqsubseteq \leqslant 1\, R$
[InverseFunctional]	$\top \sqsubseteq \leqslant 1\, R^-$
[Transitive])	$Tr(R)$
SubPropertyOf(R_1 R_2)	$R_1 \sqsubseteq R_2$
EquivalentProperties($R_1 \dots R_n$)	$R_1 \equiv \dots \equiv R_n$
DatatypeProperty(U super(U_1)...super(U_n)	$U \sqsubseteq U_i$
domain(C_1)...domain(C_m)	$\geqslant 1\, U \sqsubseteq C_i$
range(D_1)...range(D_ℓ)	$\top \sqsubseteq \forall U.D_i$
[Functional])	$\top \sqsubseteq \leqslant 1\, U$
SubPropertyOf(U_1 U_2)	$U_1 \sqsubseteq U_2$
EquivalentProperties($U_1 \dots U_n$)	$U_1 \equiv \dots \equiv U_n$
AnnotationProperty(S)	
OntologyProperty(S)	
Individual(o type(C_1)...type(C_n)	$o \in C_i$
value(R_1 o_1)...value(R_n o_n)	$\langle o, o_i \rangle \in R_i$
value(U_1 v_1)...value(U_n v_n))	$\langle o, v_i \rangle \in U_i$
SameIndividual($o_1 \dots o_n$)	$\{o_1\} \equiv \dots \equiv \{o_n\}$
DifferentIndividuals($o_1 \dots o_n$)	$\{o_i\} \sqsubseteq \neg\{o_j\}, i \neq j$

Figure 13. OWL DL axioms and facts [13]

Acknowledgement

This work was supported by the Slovak Research and Development Agency under the Contract No. APVV-20-0548 (ARIEN), also by the Slovak Scientific Grant Agency VEGA 2/0125/20, VEGA 2/0119/23 and APVV 19-0220.

REFERENCES

[1] Gomez-Valades, A.—Martinez-Tomas, R.—Rincon, M.: Integrative Base Ontology for the Research Analysis of Alzheimer's Disease-Related Mild Cognitive Impairment. Frontiers in Neuroinformatics, Vol. 15, 2021, Art. No. 561691, doi: 10.3389/fninf.2021.561691.

[2] Zouri, M.—Ferworn, A.: An Ontology-Based Approach for Curriculum Mapping in Higher Education. 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), 2021, pp. 0141–0147, doi: 10.1109/CCWC51732.2021.9376163.

[3] Karimi, S.—Iordanova, I.—St-Onge, D.: An Ontology-Based Approach to Data Exchanges for Robot Navigation on Construction Sites. 2021, doi: 10.48550/arXiv.2104.10239.

[4] Singels, L.—Biebuyck, C.—Maluleke, L.: A Formal Concept Analysis Driven Ontology for ICS Cyberthreats. In: Gerber, A. J. (Ed.): Proceedings of the First Southern African Conference for Artificial Intelligence Research (SACAIR 2020). 2020, pp. 247–263.

[5] Sattar, A.—Ahmad, M. N.—Surin, E. S. M.—Mahmood, A. K.: An Improved Methodology for Collaborative Construction of Reusable, Localized, and Shareable Ontology. IEEE Access, Vol. 9, 2021, pp. 17463–17484, doi: 10.1109/ACCESS.2021.3054412.

[6] Aguado, E.—Sanz, R.: Using Ontologies in Autonomous Robots Engineering. Robotics Software Design and Engineering, IntechOpen, 2021, doi: 10.5772/intechopen.97357.

[7] Lu, D.—Fei, J.—Liu, L.: A Semantic Learning-Based SQL Injection Attack Detection Technology. Electronics, Vol. 12, 2023, No. 6, 1344 pp., doi: 10.3390/electronics12061344.

[8] Crespo-Martínez, I. S.—Campazas-Vega, A.—Guerrero-Higueras, Á. M.—Riego-DelCastillo, V.—Álvarez-Aparicio, C.—Fernández-Llamas, C.: SQL Injection Attack Detection in Network Flow Data. Computers and Security, Vol. 127, 2023, Art. No. 103093, doi: 10.1016/j.cose.2023.103093.

[9] Nallusamy, S.—Hoo, M. H.—Zulkifle, F. A.: Controlled Experiment for Assessing the Contribution of Ontology Based Software Redocumentation Approach to Support Program Understanding. Computing and Informatics, Vol. 40, 2021, No. 5, pp. 1025–1055, doi: 10.31577/cai_2021_5_1025.

[10] Yakhyaeva, G.—Karmanova, A.—Ershov, A.: Application of the Fuzzy Model Theory for Modeling QA-Systems. Computing and Informatics, Vol. 40, 2021, No. 6, pp. 1197–1216, doi: 10.31577/cai_2021_6_1197.

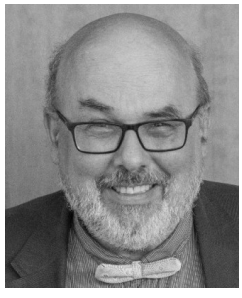
[11] Dora, J. R.—Nemoga, K.: Ontology for Cross-Site-Scripting (XSS) Attack in Cybersecurity. Journal of Cybersecurity and Privacy, Vol. 1, 2021, No. 2, pp. 319–339, doi: 10.3390/jcp1020018.

[12] Dora, J. R.—Nemoga, K.: Clone Node Detection Attacks and Mitigation Mechanisms in Static Wireless Sensor Networks. Journal of Cybersecurity and Privacy, Vol. 1, 2021, No. 4, pp. 553–579, doi: 10.3390/jcp1040028.

[13] Baader, F.—Calvanese, D.—McGuinness, D.—Patel-Schneider, P.—Nardi, D.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, 2003.

Jean Rosemond Dora works in the CAI Editorial Office as Reviewer, as well as in UI SAV. He is a penetration tester, focusing primarily on detecting and exploiting (upon request) vulnerabilities from a given environment. He holds certificates among which, Cybersecurity and Infrastructure Security Agency (CISA) from the U.S. Department of Homeland Security, Industrial Control System (ICS); Certified Ethical Hacker (CEH), Practical Network Penetration Testing (PNPT), Security Awareness Foundations and Training. He holds Ph.D. degree obtained from the Institute of Mathematics, Slovak Academy of Sciences (MUSAV). Has Master's degree from the Faculty of Electronics and Informatics, Slovak University of Technology (FEI-STU) in Bratislava. Holds a second Master's degree in computer science from the Faculty of Education in Ružomberok. He is also employed in internal/external, web applications, and wireless network assessments in penetration testing as an independent contractor. Additionally, He is an online instructor, teaching ethical hacking courses on Udemy, Thinkific, and Teachable platforms.

Ladislav Hluchý is Senior Research Scientist and Manager with more than 20 years of experience in leading national and international research projects and teams of 5 to 20 researchers. He is a competent scientist in the area of high-performance computing, multi-cloud computing, parallel and distributed information processing, and knowledge management. His research also focuses on data flow management through abstract language mechanisms. In the past, Ladislav Hluchý, Associate Professor, has participated in several cooperations with industry, which is beneficial for the transfer of the project results into practice.



Karol Nemoga Director of the Institute of Mathematics Slovak Academy of Sciences. He graduated from the Faculty of Mathematics and Physics of Charles University in Prague. In 1976, he joined the Institute of Mathematics Slovak Academy of Sciences. He has been its director since 2015. He works as University Teacher. He specializes in cryptology, computational number theory, and coding theory. He published about 30 scientific articles and about ten teaching texts. He is also working in the Association for Computing Machinery, and the Institute of Electrical and Electronics Engineers (IEEE). He is a member of the Union of Slovak Mathematicians and Physicists, the Slovak Gas Society, and the International Association for Cryptology.

Computing and Informatics, Vol. 42, 2023, 501–524, doi: 10.31577/cai_2023_2_501

MODELING AND ANALYZING USER BEHAVIOR RISKS IN ONLINE SHOPPING PROCESSES BASED ON DATA-DRIVEN AND PETRI-NET METHODS

Wangyang Yu

The Key Laboratory of Modern Teaching Technology
Ministry of Education, Xi'an, China
& School of Computer Science
Shaanxi Normal University, Xi'an, China

Zhuojing Ma

School of Computer Science
Shaanxi Normal University, Xi'an, China

Xiaojun Zhai*, Yuke Zhou*

School of Computer Science and Electronic Engineering
University of Essex, UK
e-mail: xzhai@essex.ac.uk

Weiwei Zhou

Business School, Shandong Yingcai University China

Yuan Liu

School of Computer Science
Shaanxi Normal University, Xi'an, China

* Corresponding author

Abstract. With the rapid spread of e-commerce and e-payment, the increasing number of people choose online shopping instead of traditional buying way. However, the malicious user behaviors have a significant influence on the security of users' accounts and property. In order to guarantee the security of shopping environment, a method based on Complex Event Process (CEP) and Colored Petri nets (CPN) is proposed in this paper. CEP is a data-driven technology that can correlate and process a large amount of data according to Event Patterns, and CPN is a formal model that can simulate and verify the specifications of the online shopping processes. In this work, we first define the modeling scheme to depict the user behaviors and Event Patterns of online shopping processes based on CPN. The Event Patterns can be constructed and verified by formal methods, which guarantees the correctness of Event Patterns. After that, the Event Patterns are translated into Event Pattern Language (EPL) according to the corresponding algorithms. Finally, the EPLs can be inserted into the complex event processing engine to analyze the users' behavior flows in real-time. In this paper, we validate the effectiveness of the proposed method through case studies.

Keywords: Petri net, data analysis, user behavior

1 INTRODUCTION

In recent years, with the rapid development of Internet, online shopping has become a well-known way. As a novel mode, online shopping has a great impact on peoples' lifestyle and economic development. However, due to the virtuality, dynamic and open environment, the inherent defects of the software systems and network risks pose a great threat to the security of consumers' accounts and funds [1, 2, 3].

In order to improve the security of online shopping environment, some researches focus on authentication as the core security method, among which digital certificate [4], authentication technology [5] and dynamic verification code [6] are the most common method. Mining of user behavior data has been increasingly applied to the construction and analysis of user behavior patterns. Gull and Pervaiz builds user behavior patterns through data mining to analyze users' actual purchasing behaviors [7]. Some identity authentication methods are proposed by monitoring the behaviors of mobile devices [8]. As a distributed Web application, online shopping systems are loosely coupled and interactively complex. Despite the third party service providers bridging the gap of trustiness between merchants and users, their involvement complicates the logic flow in the checkout process [9]. Logic flows of online shopping systems allow malicious users to carry out malicious behaviors under legal identities, e.g., purchase products using fabricated payments [10, 11]. A user can abuse legitimate application-specific functionality against developers' intentions [12]. As a result, vulnerable servers are exposed to malicious users who can potentially implement the behaviors such as alternate the control and data flows through concurrent interactions [13].

With the development of data science, the use of machine learning to conduct real-time analysis of user behaviors has gradually gained more attention. Jiang et al. proposed the online detection methods for credit card fraud based on machine learning [14, 15, 16]. Credit card payment is an important component of the entire online shopping process. The online shopping process includes not only the payment, but also the place order, notification, confirmation, update information, and many other operations [17]. Guaranteeing the real-time security of the entire online shopping processes is the key of avoiding the frequent risks. Today's e-commerce businesses have become increasingly hybrid, with their program logic being distributed across multi-participants, including the servers and their clients, along with various third party API service providers [18]. Their respective business processes construct the entire transaction process. This integration introduces new security challenges due to complex interaction behaviors among multi-participants [19].

Therefore, the real-time identification of behavior risks in online shopping processes is imminent. In the process of risk prevention and control, we should fully consider the relationships among multiple events in the online shopping process, and dynamically identify users' risky behaviors in real-time. Real-time monitoring of the users' shopping data streams and intelligent identification of user behaviors can effectively improve the security of online shopping processes.

Complex Event Processing (CEP) is an an emerging reference framework and standard for building and managing event-driven information systems [20, 21]. The goal is to get the meaningful complex events by reasoning and analyzing the event data flow, and respond in real-time. The CEP framework includes Event Pattern construction and recognition, event association and abstraction, event-driven processing, etc. CEP does not depend on specific methods and technologies, and many new theories, methods and technologies are needed to research and design specific CEP systems in a certain field. At present, it is widely used in the fields of business process analysis [22], financial analysis [23], RFID [24] and wireless sensor network [25].

In addition, the Event Patterns of most current researches on CEP are based on SQL-like statements and non-formal rules [26]. It is not enough to accurately describe the online shopping process which is distributed, complex, concurrent and loosely coupled. Specially, the correctness of Event Patterns and EPLs should be validated and guaranteed. Petri nets are a formal model that is suitable for portraying distributed systems that can accurately describe the concurrency and event relationships [27, 28], and widely used in Workflow [29], Web services [30], control systems [31, 32]. Compared with non-formal rules, Petri nets have a wealth of analytical techniques and other derived advanced models (e.g., CPN), and can be applied to validate and analyze the Event Patterns formally and effectively, thus ensuring the correctness of the Event Patterns. CPN is a kind of high-level Petri nets with powerful graphical modeling ability for depicting discrete events systems. Meanwhile, it can effectively describe the complex structures in dynamic systems, such as sequence, concurrency, and selection. On the other hand, CPN has a mature visual modeling tool (CPN Tools). Ref. [33] has proposed a meaningful complex

event processing model by Prioritized Colored Petri Net based on MEdit4CEP platform. Thus, CPN is an ideal model for depicting the users' risky behaviors of online shopping processes.

Therefore, in this paper, we coalesce the formal model (CPN) and data-driven framework (CEP) to construct the methodology for identifying user behavior risks of online shopping processes in real-time. The contributions of this paper mainly include:

- For accurately depicting and validating the Event Pattern of user behavior risks, this paper defines the formal modeling and validating scheme based on CPN.
- This paper proposes the algorithms for transforming the formal Event Pattern to EPL.
- This paper constructs the risk identification mechanism based on CPN and CEP to cope with the user behavior risks in online shopping processes.

The remainder of this paper is organized as follows. Section 2 introduces the related methods used in this paper. Section 3 illustrates the modeling principle and analyzing process of Event Patterns based on CPN. Section 4 introduces the risk identification mechanism based on CPN and CEP, and the demo system of above methodology is implemented. Section 5 concludes the paper.

2 RELATED METHODS

This section mainly introduces the related concepts and methods involved in the paper including CPN and CEP.

2.1 Colored Petri Nets (CPN)

A CPN is a directed graph that combines the Petri net and the StandML (functional programming language). In CPN, a specific color set (data type) is provided for each place, the data types of the colored set mainly include int, boolean, string, list and record. We can also set the guard function and priority on the transition, and set the expressions on the arc in the process of constructing a CPN model. When the variables in the colored set satisfy the conditions of the input arc and the settings of the transition, the corresponding transition can be fired. More details on CPN can be seen in [27, 34].

Definition 1 ([34])**.** A Colored Petri Net is a nine-tuple $CPN = (P, T, A, \Sigma, V, C, G, E, I)$, where

1. P is a finite set of places.
2. T is a finite set of transitions such that $P \cap T = \emptyset$.
3. $A \subseteq P \times T \cup T \times P$ is a set of directed arcs.
4. Σ is a finite set of non-empty color sets.

5. V is a finite set of typed variables such that $Type[v] \in \Sigma$ for all variables $v \in V$.
6. $C : P \to \Sigma$ is a color set function that assigns a color set to each place.
7. $G : T \to EXPR_V$ is a guard function that assigns a guard to each transition t such that $Type[G(t)] = Bool$.
8. $E : A \to EXPR_V$ is an arc expression function that assigns an arc expression to each arc a such that $Type[E(a)] = C(p)_{MS}$, where p is the place connected to the arc a.
9. $I : P \to EXPR_\emptyset$ is an initialization function that assigns an initialization expression to each place p such that $Type[I(p)] = C(p)_{MS}$.

Definition 2 ([34])**.** A binding element $(t, b) \in BE$ is enabled in a marking M if and only if the following two properties are satisfied:

1. $G(t)\,\langle b \rangle$.
2. $\forall p \in P : E(p, t)\,\langle b \rangle \ll= M(p)$.
3. When (t, b) is enabled in M, it may occur, leading to the marking M' defined by: $\forall p \in P : M'(p) = (M(p) - -E(p, t)\,\langle b \rangle) + +E(t, p)\,\langle b \rangle$.

2.2 Complex Event Process (CEP)

CEP is a real-time data processing framework, which is mainly used to research on how to efficiently extract valuable events from a large number of simple event streams, and can be abstracted and aggregated into complex events. It can quickly find the abnormal situation from the real-time data streams, which is suitable for the scene of abnormal detection [20, 23]. First, the acquired data flow (event stream) is captured by using filtering, association and aggregation; second, based on the temporal relation and aggregation relation among events, by developing the EPL, the valuable events (complex events) are continuously excavated from the event stream at different level, and then they can be abstracted and aggregated into high-level complex events; final, the highest-level complex events are responded by notifying the system, software, or device when a particular situation has been detected [26].

The so-called event means the meaningful state change in actual systems, usually divided into atomic events and complex events. Atomic events refer to the most basic information generated at a certain point-in-time in the process of system execution, which contains limited information and cannot be separated. Complex events means value ones those are generated by pattern matching of atomic events, as shown in Figure 1. A complex event usually includes the time of occurrence, event attribute value and the event name. The Event Pattern is a template that is used to match the set of eligible event stream and accurately describes the causal, time and logical relationships among events. Event Pattern is mainly implemented by EPL, which is a SQL-like language with a rich set of advanced processing expressions. It provides a lot of times and pattern operators to define patterns of interest.

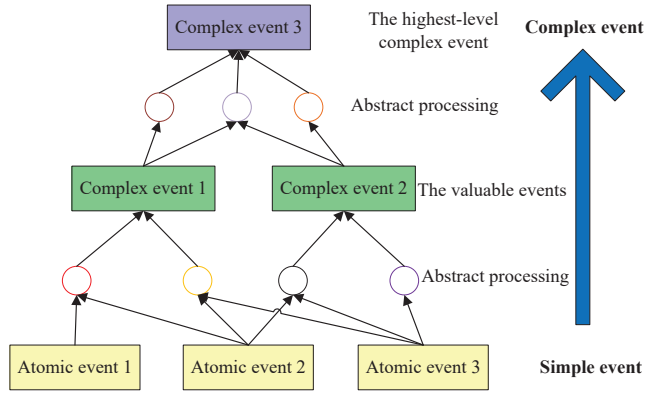


Figure 1. The abstract process of complex events

3 THE EVENT PATTERN MODELS OF BEHAVIOR RISK IDENTIFICATION

E-commerce business interaction is a typical distributed and concurrent system in the open Internet, the recent developments of which has opened a range of security challenges. A major reason is that the distributed system possesses concurrency and the execution may proceed in many different ways. A typical online distributed system handles process concurrency in a number of fashion. It is easy for a malicious user to implement behavior interactions during this process concurrency, which might lead to logical vulnerabilities in the system execution.

The identification models are the basis of analyzing the user behavior risks in online shopping processes. In this paper, a formal modeling method based on CPN is established to depict the risk behaviors of online shopping users, and the models can be validated to guarantee the correctness. In this section, the behavior risk identification models focus on the single-user and multi-user scenes. The risky behaviors includes but it is not limited to: the user's account address or payment method is abnormal in a short time; The abnormality of the payment amount is mainly reflected in two aspects: on the one hand, the user continuously places orders and the purchase amount continues to increase; on the other hand, the user's payment amount is greater than the average payment amount of the user over a period of time. The main colored sets and the variable declarations in Event Pattern models are shown in Table 1.

3.1 Modeling Principles

In general, the operations of identifying user behavior risks mainly include input, filtering, aggregation, and analysis. However, the order of transition executions has a great influence on the result. In order to reduce the adverse effects, it is necessary

Colored Sets	**Variable Declarations**	**Implications**
colset Num = int	var n	The number of user behavior stream
colset Usern = string	var usern	The user name
colset Order = int	var order	The order number
colset Gross = int	var gross	The payment amount
colset Address = string	var place, address	User account address
colset Way = string	var way	The payment method
colset State = bool	var state	The payment result
colset Timee = int	var timee, timee1	The payment time

Table 1. Main type definitions of the models

to set the transition priority in the process of modeling. According to the structure that can trigger transitions under a certain identification state and the region where it is located, the relevant principles are settled for the priority of transitions.

The meaning of the transition in CPN is usually a executable action, and the priority of the transition should meet the scenario setting of the online shopping process. Meanwhile, the transition structures in a certain marking of Event Pattern models are generally divided into the sequential, selection and concurrent structures, as shown in Figure 2. The transition, which is marked red, indicates that it can be triggered in the current state.

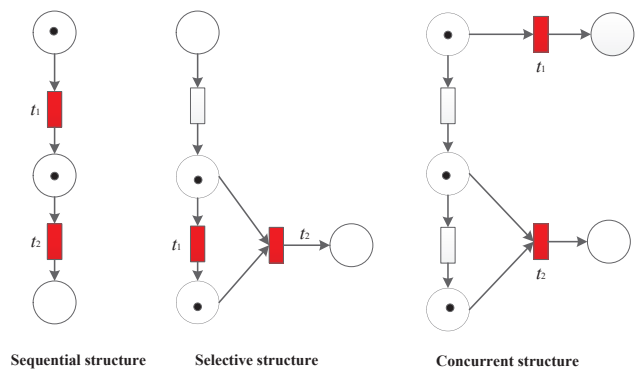


Figure 2. The transition structures

When we depict the model, if there is a sequence structure between two transitions that can be fired under a certain marking, setting different priorities for transitions will not affect the final result. If there is a selection structure between two transitions, the priority setting of the transitions has different effects on the final result. Therefore, in this scene, we should set the priority according to actual scene. If there is a concurrence structure between two transitions, the priority of the transitions does not affect the final result. The modeling scheme of Event Pattern models based on CPN is listed as follows:

- Online shopping user behaviors should be numbered in order, and ensure that transitions are triggered in order;
- The priority should be set according to the structure and region of the transition under a certain state;
- The setting of the guard function and arc expression need to satisfy the identification conditions of user behavior risks on the corresponding transition;
- When using the sliding or fixed window to analyze user behaviors, it is needed to ensure the size of the window in real-time;
- When analyzing the user behavior risks, it is necessary to discharge the risk-free event streams (tokens) in real-time.

Case 1 – The Event Pattern model for the anomaly detection of the user's account address or the way of payment

This section illustrates the formal model of Event Pattern for abnormal detection of user account addresses, which mainly includes the filtering and identifying operations of online shopping behaviors. The model is shown in Figure 3.

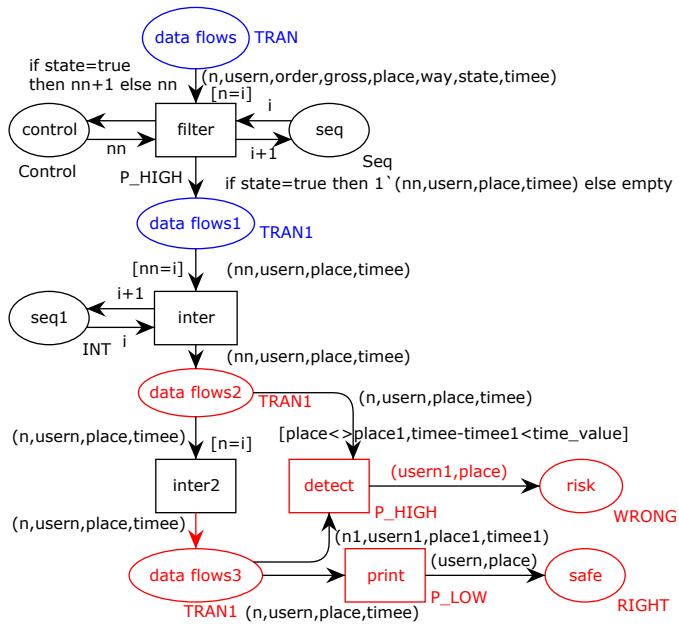


Figure 3. The Event Pattern model for the anomaly detection of user's account address

The filtering operation is mainly used to obtain the behavior streams whose statuses are paid in the entire online shopping behaviors. In the model, the behavior streams are represented by *colset* $TRAN1 = product\ INT * STRING *$

STRING∗ INT. The filtering conditions are represented by the output arc functions of "filter" and the transition's guard function $n = i$. The tokens of "data flows1" are used to identify the users' abnormal account addresses. The analysis operation of the next user behavior stream is controlled by the arc function $i+1$ of the place "seq".

In the identification phase, we define risky behaviors as *colset WRONG = product Usern ∗ Address*, and risk-free behaviors as *colset RIGHT = product Usern ∗ Address*. The guard function *place <> place1, timee-timee1 < time_value* of the transition "detect" indicates that the user account address is abnormal in a short time. Since the priority of "detect" is higher than "inter2", if the tokens of "data flow2" and "data flow3" satisfy the guard function, "detect" can be fired, and the tokens of the place "risk" are risky behaviors. Otherwise, the transition "print" is fired, the tokens of the place "safe" are risk-free behaviors, indicating that the user's account is safe.

The anomaly detection of the user's account address mainly focuses on the address and time in the user's shopping data. If the user's physical address changes in a short time, we define the captured data stream as a risk behavior. The model structure of abnormal detection of user's payment way is the same as that of abnormal detection of user's account address, and it only needs to change the variables of the colored set and arc variable in the specific process, so it is not necessary to be introduced in detail in this paper.

Case 2 – The Event Pattern model for the anomaly detection of the user's payment amount

In this case, the abnormal payment amount means that the user places orders continuously and the purchase amount continues to increase. The model is shown in Figure 4. The model is divided into three parts: the behavior filtering, the behavior anomaly judgment, and the judgment of payment amount continuously increasing. The blue area represents the behavior filtering phase, which is the same as Figure 3. The only difference is that variable m is added to the color set of the place "data flow". The number of identification of user behaviors is determined by m, whose default value is 1. The red area represents the abnormal judgment of behaviors. The purple area is used to determine whether the payment amount continues to increase. In the model, we set the default number of identifications as 4.

The tokens in the place "data flow" is used to simulate the data of user behaviors. The guard functions on the transitions ("filter", "inter", and "inter1") are used to control the firing sequence. When all the prepositive places of "filter" have tokens, it is enabled. The arc function *if state = true then nn+1 else nn* is used for renaming the behavior flow of whose payment state is already paid, so as to avoid the confusion of the identification order of the behavior flows, guarantee the accuracy of the identification result, and complete the filtering operation. Since the priority of transition "detect" is higher than that of "inter2", once the

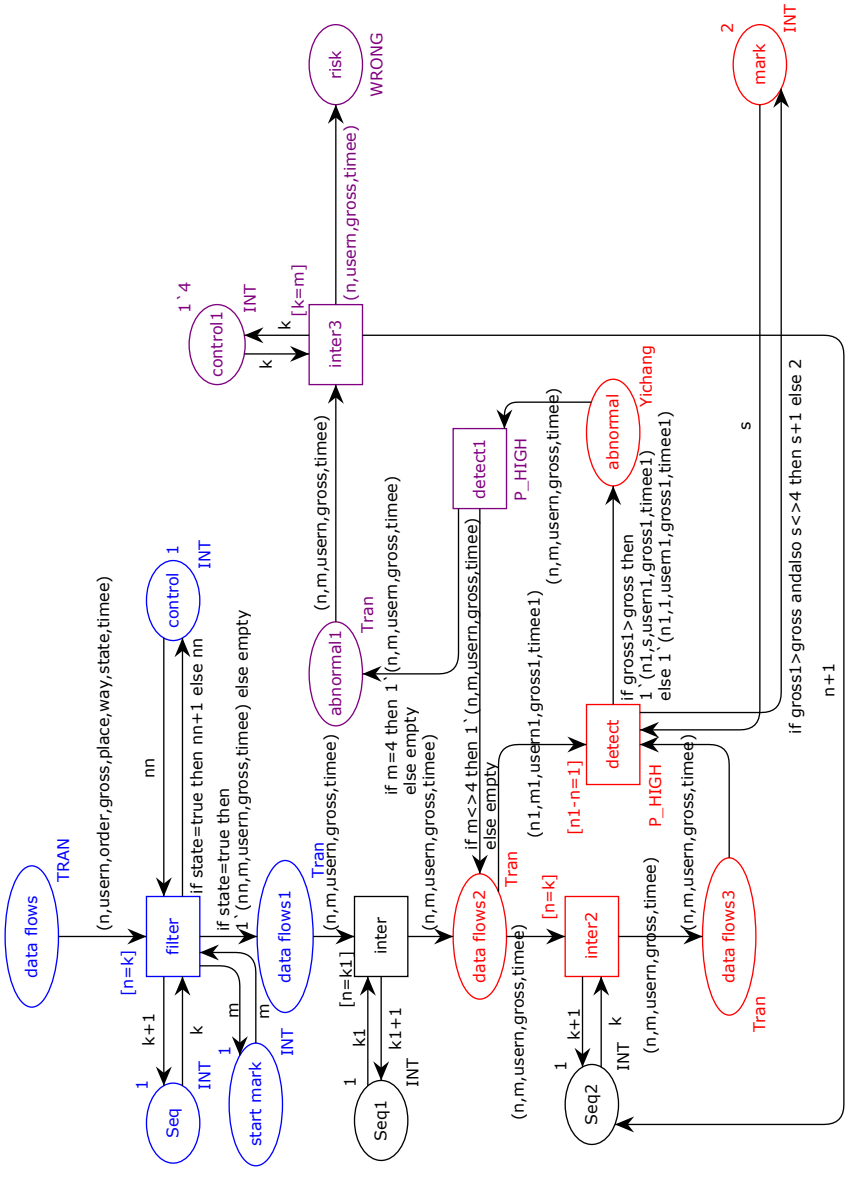


Figure 4. The Event Pattern model for the anomaly detection of the continuous increase of user payment amount

places "data flow" and "data flow2" have token, which satisfies the conditions of the identification operation, the transition "detect" can be fired. At this point, the place "mark" will increase the number of times of identifications by 1. Once the identification is completed, the lowest-numbered token in "data flow" will enter in "data flow2" by firing the transition, and then a new round of identification will start. If the identification times of one user satisfy the marking of "control1", "inter3" can be fired, and the token in "risk" indicates that the user account is at risk.

Case 3 – The Event Pattern model for the anomaly detection of the multi-accounts

The risky behaviors of the multi-account user refer to that the user has two or more accounts to place orders. Order characteristics include: the time interval among different orders are very short, and the order numbers are very similar, the payment statuses are different. Once the above characteristics are satisfied, the user's behavior is judged as risk. The specific model is shown in Figure 5. In the model, the place "timeshold" represents the time threshold. The characteristics of the user's risk behaviors are mainly reflected in the guard function of the transition "detect", and the arc function between the transition "detect" and the place "risk". We can use the model to identify the illegal behaviors.

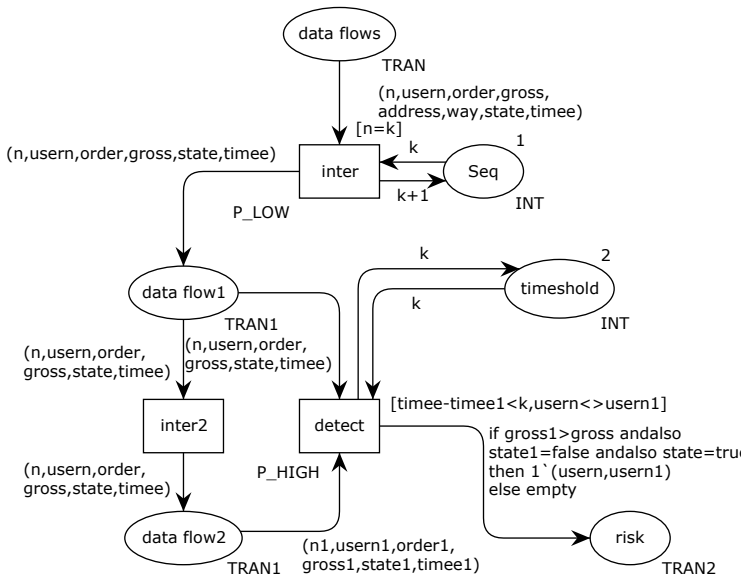


Figure 5. The Event Pattern model of the anomaly detection of the multi-accounts

3.2 Model Validation

This section illustrates the verifying process by Algorithm 1. In Algorithm 1, the variable "Result" depicts whether Event Pattern model is correct. The value of "Result" is obtained by generating a state space diagram or state space report in CPN Tools. The state space diagram is a directed graph. Each reachable mark has a node. The state space report contains the bounded properties of the places. When we consider the reachable markings, the bounded properties can get the specific information.

Algorithm 1: The validation process

Input: (CPN, M_0), where $CPN = (P, T, A, \Sigma, V, C, G, E, I)$ is a Event Pattern model and M_0 a marking of it.
Output: Result.
Result = 'Correct';
Generate a state space diagram or state space report;
if *The number of leaf nodes in the state space diagram is* > 1 *and there are many different markings* **then**
  Result = 'False';
else
  return Result;
end
if *Lower in the state space report* $!= 0$ **then**
  Result = 'False';
else
  return Result;
end

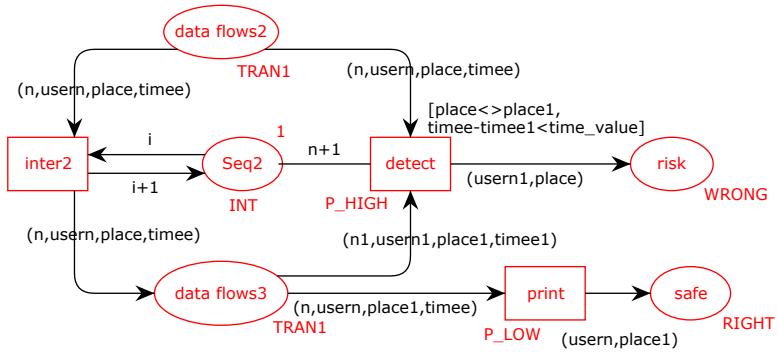


Figure 6. The correct model structure

By Algorithm 1, only the Event Pattern model of the user's account address has errors. The reason is that when the place ("data flows1") has a token or some tokens, the transition "inter2" can be fired, which results in the change of the order of the user shopping data flows. To solve this problem, we need to modify the transition's guard functions, meanwhile ensuring that the user data flows are executed sequentially. We modify the model as shown in Figure 6.

3.3 Model to EPL

CEP refers to the real-time processing of all input event streams according to predefined event processing rules or Event Patterns. Once the input event streams meet the event processing rules, complex events will be generated. In this paper, the event processing rules are provided by Esper and described in the Event Pattern language, which is a like_SQL language [35, 36].

After the modeling and validating of the Event Pattern model, the correctness is guaranteed. Then, the CPN models should be transformed to EPL according to the follow steps.

Algorithm 2: The generation of EPL

Input: An Event Pattern model $CPN = (P, T, A, \Sigma, V, C, G, E, I)$.
Output: An EPL.

1. Translating CPN to a simplified model $PN = (P, T, A)$;
2. Make PN as a directed graph, and the node access sequence $\phi = \{p_1, p_2, \ldots, p_m\}$ is generated according to the depth traversal algorithm, where m is the number of places;
3. Generate the input matrix $A1_{m\times n}$ and output matrix $A2_{m\times n}$ of PN, n is the number of the transitions;
4. Using ϕ, $A1_{m\times n}$ and $A2_{m\times n}$ to achieve structure matching, and generate the keywords: 'select', 'having', 'where', 'group by', 'from', etc.;
5. Variables are generated by the color set of the places, input and output variables of transitions, and guard functions;
6. Generate expressions of clause based on the corresponding places, transitions, keywords;
7. Obtain the EPL.

In the process of identifying the risky behaviors, the event type of the users' shopping behavior is defined as "TranEvent". The event attributes and the meanings are shown in Table 2. We have modeled the cases of possible risky Even Patterns in above section, and then we convert them into EPLs by Algorithm 2. The corresponding Event Patterns and EPLs as shown in Table 3.

Variable Name	Attribute	Meaning
userID	int	ID number of a user
gross	double	The amount paid
orderID	int	Order number
tradeway	string	The payment way
tradeplace	string	The payment place
tradetime	timestamp	The payment time
tradestate	bool	The Payment status

Table 2. The event type of users' shopping behavior

4 USER BEHAVIOR RISK IDENTIFICATION SYSTEM BASED ON ESPER

Nowadays, engines for CEP include Esper [1], Apache Flink [2], Oracle CEP [3], etc. Considering the open source characteristics of Esper and its EPL is more in line with the research in this paper, Esper is chosen to be the engine of user behavior risk identification system.

Esper supports real-time analysis and processing of massive event streams, which is done primarily through the JAVA [37]. The modular design of user behavior risk identification system is shown in Figure 7, which mainly includes:

- Capture the behavior flow of online shopping users: serve as the data source of CEP;
- Input adapter is used to convert the captured data source into an event source through byte filtering, aggregation, etc.;
- Historical access database provides an interface to access the database. While processing the data in the window mechanism, the engine directly call the user behavior flow in the historical access;
- Esper engine is the core of CEP technology including acquiring the configurations, defining the events, defining the EPLs, defining listeners and binding listeners, etc.;
- Output adapter is used to send messages that the Esper engine is listening to external systems.

Among them, Esper engine is mainly composed of the following steps:

- Get the configuration of the Esper engine:

```
Configuration config = new Configuration();
config.addEventType("TranEvent", TranEvent.class.getName());
```

[1] http://www.espertech.com/esper/
[2] https://flink.apache.org/
[3] https://docs.oracle.com/

No	Event Pattern	EPL
1	The user's account address changes within a short period of time (10 s).	select * from pattern [every temp1 = TranEvent → temp2 = TranEvent (temp2.tradeplace != temp1.tradeplace, temp1.userID = temp2.userID, temp1.tradeway = temp2.tradeway, (temp2.tradetime.getTime() - temp1.tradetime.getTime())/1000 < 10, temp1.tradestate = true, temp1.tradestate =temp2.tradestate)]
2	The user's shopping way changes within a short period of time (10 s).	select * from pattern [every temp1 = TranEvent → temp2 = TranEvent (temp2.tradeway != temp1.tradeway, temp1.tradeplace = temp2.tradeplace, temp1.userID = temp2.userID, (temp2.tradetime.getTime() - temp1.tradetime.getTime())/1000 < 10, temp1.tradestate=true,temp1.tradestate = temp2.tradestate)]
3	The user places orders continuously and the payment status of the orders are paid, the order amount presents an increasing trend and the last order amount is much lager than the first order amount.	select * from TranEvent match_recognize (measures A as temp1, B as temp2, C as temp3, D as temp4 pattern (A B C D) define A as A.gross IS NOT NULL, B as (A.gross < B.gross), C as (B.gross < C.gross), D as (C.gross < D.gross) and D.gross > 4*A.gross) having temp1.userID = temp2.userID and temp2.userID = temp3.userID and temp3.userID = temp4.userID and temp1.tradestate = 1 and temp2.tradestate = 1 and temp3.tradestate = 1 and temp4.tradestate = 1
4	The people uses different accounts to purchase the same goods, and the former pays much more than the latter, it is worth noting that the former state of payment is unpaid, the latter state of payment is paid.	select * from pattern[every temp1 = TranEvent → temp2 = TranEvent (temp2.tradeway = temp1.tradeway, temp1.tradeplace = temp2.traeplace, temp1.userID != temp2.userID, (tmep2.tradetime.getTime() - temp1.tradetime.getTime())/1000 < 5, temp1.tradestate!=temp2.tradestate)]

Table 3. The corresponding event patterns and EPLs

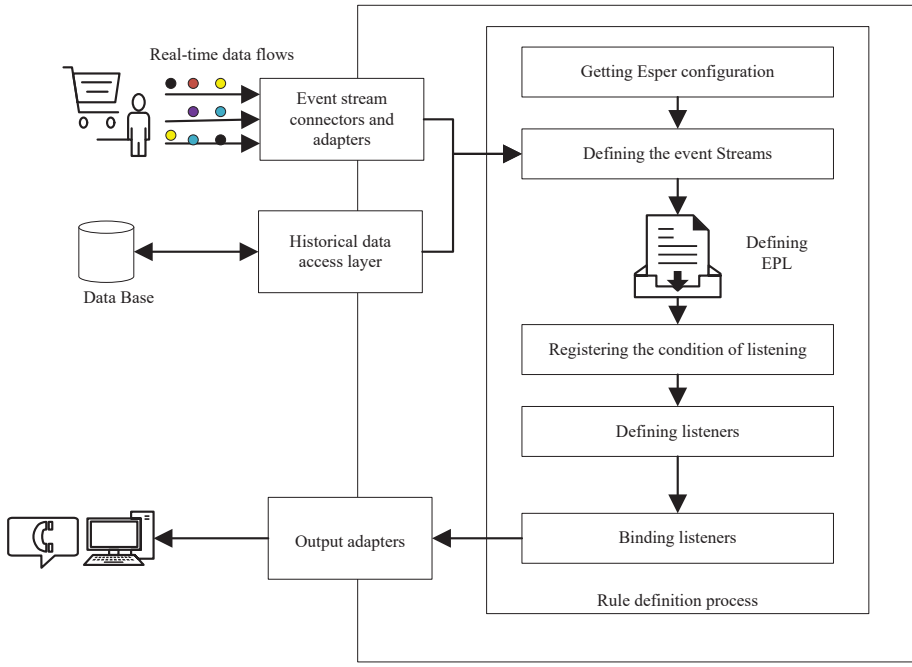


Figure 7. The system architecture

EPServiceProvider cep = EPServiceProviderManager.
getProvider("myCEPEngine", config);

- Defining event types:

  When defining the event types, they usually includes JavaBean, Map, and XML. In order to facilitate analysis, we use JavaBean to define user behavior events for online shopping (TranEvent).

- Add EPL:

  String epl = "select * from TranEvent.win:time(30 sec) group by usern";

- Register the condition of listening:

  EPAdministrator cepAm = cep.getEPAdministrator();
  EPStatement statement = cepAm.creaeEPL(epl);

- Defining listeners:

  The listener is an interface provided by Esper to listen for predefined rules in the engine. The listener is notified as soon as the event satisfies the EPL, because the interface contains the method *update*(), which involves two parameters, newEvents, and oldEventEvents, for receiving the events. At the same time, both parameters are an array of event beans, and the method to get the field values in the EPL is eventbean.get("usern");

- Binding listeners:

  MyListener listener=new MyListener();
  Statement.addListener(listener).

As we cannot get the real trading data, we develop a demo online shopping platform (Figure 8) to produce the simulation data. We produce the order information of simulation users from the specified date 2019.12.01 to 2019.12.30 on the demo platform. For example, the shopping information of the user (id = 37983443) in this period is shown in Table 4. It can be clearly seen from the table that the payment address of orders No. 3 and No. 4 had changed within a short time, indicating that the user's account was at risk. We get all the order information of the user (id = 38975436) in the period as shown in Table 5. During the specified period from 2019.12.23 13:30:23 to 2019.12.23 13:45:00, the user had placed orders continuously and the order amount had continued to increase, and the amount of the last order had been much larger than the amount of the first order, indicating that the user account was at risk. The order information of all users from 2019.12.24 08:00:00 to 2019.12.24 09:00:00 is shown in Table 6. We can see that:

1. The payment time between "18976512" and "19001416" is relatively short;
2. The products purchased by "18976512" and "19001416" are of the same type;
3. The order payment status of "18976512" is unpaid, while the order payment status of "19001416" is paid;
4. The payment amount of "18976512" is much larger than the latter.

These characters indicate that a user uses different accounts to place orders, which may lead to the risk of order replacement attack, so that the user account is at risk. Above examples matches cases 1–3. We input the shopping data streams into the Esper, and can seen from the Figures 9 a), 9 b), 9 c) that the system can successfully capture the risky behaviors.

No.	Gross	Order Number	Payment Way	Payment Place	Payment Time	Payment Status
1	123	1	TPP1	Beijing	2019.12.12 19:30:23	true
2	56.8	2	TPP1	Beijing	2019.12.12 19:45:23	true
3	196	1	TPP1	Beijing	2019.12.17 08:30:23	true
4	444	2	TPP1	Xi'an	2019.12.17 08:30:28	true
5	156	1	TPP2	Beijing	2019.12.25 13:13:56	true

Table 4. Online shopping user behavior flow (id = 37983443)

5 CONCLUSION

Nowadays, online shopping is an indispensable consumption way for people, which has a great impact on people's life. However, there are some risks in the process of

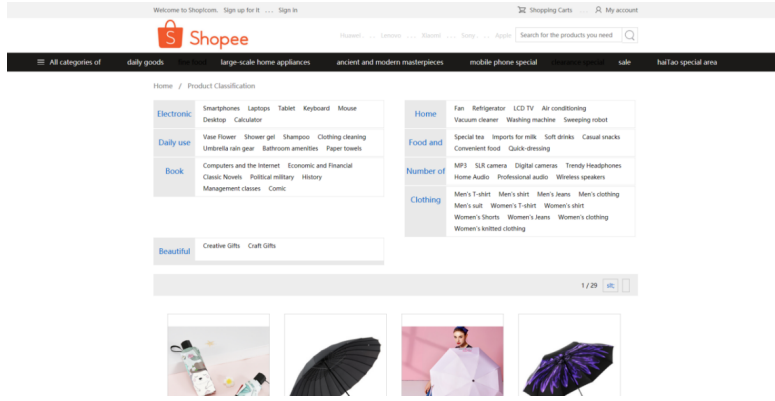


Figure 8. The demo online shopping platform

No.	Gross	Order Number	Payment Way	Payment Place	Payment Time	Payment Status
1	456	1	TPP2	Xi'an	2019.12.01 19:30:23	true
2	325	1	TPP3	Xi'an	2019.12.17 15:30:28	false
3	356	1	TPP1	Beijing	2019.12.23 13:30:23	true
4	778	2	TPP1	Beijing	2019.12.23 13:35:11	true
5	779	3	TPP1	Xi'an	2019.12.23 13:39:45	true
6	19	4	TPP2	Xi'an	2019.12.23 13:43:09	false
7	1 467	5	TPP1	Xi'an	2019.12.23 13:45:09	true

Table 5. Online shopping user behavior flow (id = 38975436)

online shopping. Real-time identification of online shopping user behaviors based on CEP can effectively avoid the generation of risk behaviors. In this work, we propose an Event Pattern modeling method based on CPN, which can depict and validate Event Patterns of user behavior risk effectively. Then, we convert the model to EPL according to the specific steps. Combining CEP with CPN to identify user behavior risk can improve the security of online shopping. In the future work, we will model risk behaviors of online shopping processes from the perspectives of consumers, sellers, and the third parties.

User Account	Gross	Payment Way	Payment Place	Payment Time	Payment Status	Product Types
14 537 817	456	TPP3	Chongqing	2019.12.24 08:01:12	true	Home appliances
18 976 512	7 980	TPP1	Beijing	2019.12.24 08:11:12	false	Gold ware
19 001 416	70	TPP1	Beijing	2019.12.24 08:11:36	true	Gold ware
34 516 537	779	TPP1	Nanjing	2019.12.24 08:30:54	true	Clothing

Table 6. User behavior flows in specified period

```
Sending TranEvent:
Event Flows—UsernID: 91004582 gross: 573.0 way: TPP2 address: Wuhan ordern: 1 time: 2019-12-16 23:19:57.0 status: true
Sending TranEvent:
Event Flows—UsernID: 37983443 gross: 196.0 way: TPP1 address: Beijing ordern: 1 time: 2019-12-17 08:30:23.0 status: true
Sending TranEvent:
Event Flows—UsernID: 76321750 gross: 345.0 way: TPP2 address: Tianjin ordern: 1 time: 2019-12-17 08:30:27.0 status: true
Sending TranEvent:
Event Flows—UsernID: 37983443 gross: 444.0 way: TPP1 address: Xian ordern: 2 time: 2019-12-17 08:30:28.0 status: true
--------------------------------------------------
UserID:37983443 Account address changes within a short period of time
This phenomenon indicates that the user account is at risk
--------------------------------------------------
Sending TranEvent:
Event Flows—UsernID: 38975436 gross: 325.0 way: TPP3 address: Xian ordern: 1 time: 2019-12-17 15:30:28.0 status: false
```

a) The system identification result of '37983443'

```
Sending TranEvent:
Event Flows—UsernID: 38975436 gross: 325.0 way: TPP3 address: Xian ordern: 1 time: 2019-12-17 15:30:28.0 status: false
Sending TranEvent:
Event Flows—UsernID: 38975436 gross: 356.0 way: TPP1 address: Beijing ordern: 1 time: 2019-12-23 13:30:23.0 status: true
Sending TranEvent:
Event Flows—UsernID: 38975436 gross: 778.0 way: TPP1 address: Beijing ordern: 2 time: 2019-12-23 13:35:11.0 status: true
Sending TranEvent:
Event Flows—UsernID: 38975436 gross: 779.0 way: TPP1 address: Beijing ordern: 3 time: 2019-12-23 13:39:45.0 status: true
Sending TranEvent:
Event Flows—UsernID: 38975436 gross: 19.0 way: TPP2 address: Beijing ordern: 4 time: 2019-12-23 13:43:09.0 status: false
Sending TranEvent:
Event Flows—UsernID: 38975436 gross: 1467.0 way: TPP1 address: Beijing ordern: 5 time: 2019-12-23 13:45:09.0 status: true
------------------------------------------------------------------------
UserID:38975436 payment amount continues to increase
This phenomenon indicates that the user account is at risk
------------------------------------------------------------------------
Sending TranEvent:
```

b) The system identification result of '38975436'

```
Sending TranEvent:
Event Flows—UsernID: 74389126 gross: 456.0 way: TPP2 address: Beijing ordern: 2 time: 2020-06-21 22:41:33.0 status: true
Sending TranEvent:
Event Flows—UsernID: 14537817 gross: 456.0 way: TPP3 address: Chongqing ordern: 4 time: 2020-06-22 19:07:53.0 status: true
Sending TranEvent:
Event Flows—UsernID: 18976512 gross: 7980.0 way: TPP1 address: Beijing ordern: 3 time: 2020-06-22 19:07:58.0 status: false
Sending TranEvent:
Event Flows—UsernID: 19001416 gross: 70.0 way: TPP1 address: Beijing ordern: 1 time: 2020-06-22 19:01:38.0 status: true
--------------------------------------------------
Two user accounts (the same person) place two orders in succession
--------------------------------------------------
--------------------------------------------------
Urgent Message-UserID:18976512 and UserID:19001416exit the order replacement risk
--------------------------------------------------
```

c) The system identification result of 'multi-account'

Figure 9. The system identification results

Acknowledgement

This work was supported by the Natural Science Foundation of Shaanxi Province under (No. 2021JM-205) and Open Research Fund of Anhui Province Engineering Laboratory for Big Data Analysis and Early Warning Technology of Coal Mine Safety (No. CSBD2022-ZD05).

REFERENCES

[1] Li, Z.—Huang, M.—Liu, G.—Jiang, C.: A Hybrid Method with Dynamic Weighted Entropy for Handling the Problem of Class Imbalance with Overlap in

Credit Card Fraud Detection. Expert Systems with Applications, Vol. 175, 2021, Art. No. 114750.

[2] Chen, E. Y.—Chen, S.—Qadeer, S.—Wang, R.: Securing Multiparty Online Services via Certification of Symbolic Transactions. Security and Privacy (SP), 2015 IEEE Symposium on, IEEE, 2015, pp. 833–849.

[3] Xie, Y.—Liu, G.—Yan, C.—Jiang, C.—Zhou, M.: Time-Aware Attention-Based Gated Network for Credit Card Fraud Detection by Extracting Transactional Behaviors. IEEE Transactions on Computational Social Systems, 2022.

[4] Sadikin, M. A.—Wardhani, R. W.: Implementation of RSA 2048-Bit and AES 256-Bit with Digital Signature for Secure Electronic Health Record Application. 2016 International Seminar on Intelligent Technology and Its Applications (ISITIA), 2016, pp. 387–392, doi: 10.1109/ISITIA.2016.7828691.

[5] Wu, L.—Chen, T.—Qiao, C.—Li, Z.: Authentication Technology of Mobile Internet of Things Based on the Dynamic Password. 2018 IEEE International Conference of Safety Produce Informatization (IICSPI), 2018, pp. 204–208, doi: 10.1109/IICSPI.2018.8690463.

[6] Muñoz, A.—Toutouh, J.—Jaime, F.: A Review of Dynamic Verification of Security and Dependability Properties. 2019, pp. 162–187, doi: 10.4018/978-1-5225-7353-1.ch007.

[7] Gull, M.—Pervaiz, A.: Customer Behavior Analysis Towards Online Shopping Using Data Mining. International Multi-Topic ICT Conference.

[8] Phillips, M. E.—Stepp, N. D.—Cruz-Albrecht, J.—Sapio, V. D.—Sritapan, V.: Neuromorphic and Early Warning Behavior-Based Authentication for Mobile Devices. 2016 IEEE Symposium on Technologies for Homeland Security (HST), 2016.

[9] Wen, S.—Xue, Y.—Xu, J.—Yuan, L. Y.—Song, W. L.—Yang, H. J.—Si, G. N.: Lom: Discovering Logic Flaws Within MongoDB-Based Web Applications. International Journal of Automation and Computing, Vol. 14, 2017, No. 1, pp. 106–118.

[10] Sun, F.—Xu, L.—Su, Z.: Detecting Logic Vulnerabilities in E-Commerce Applications. NDSS, 2014.

[11] Wang, R.—Chen, S.—Wang, X.—Qadeer, S.: How to Shop for Free Online–Security Analysis of Cashier-as-a-Service Based Web Stores. Security and Privacy (SP), 2011 IEEE Symposium on, IEEE, 2011, pp. 465–480.

[12] Enumeration, C. W.: CWE-840 Business Logic Errors. The MITRE Corporation, Jul, 2014.

[13] Cwe, C.: 472: External Control of Assumed-Immutable Web Parameter.

[14] Jiang, C.—Song, J.—Liu, G.—Zheng, L.—Luan, W.: Credit Card Fraud Detection: A Novel Approach Using Aggregation Strategy and Feedback Mechanism. IEEE Internet of Things Journal, Vol. 5, 2018, No. 5, pp. 3637–3647, doi: 10.1109/JIOT.2018.2816007.

[15] Zheng, L.—Liu, G.—Yan, C.—Jiang, C.—Zhou, M.—Li, M.: Improved Tradaboost and Its Application to Transaction Fraud Detection. IEEE Trans-

actions on Computational Social Systems, Vol. 7, 2020, pp. 1304–1316, doi: 10.1109/TCSS.2020.3017013.

[16] Xie, Y.—Liu, G.—Yan, C.—Jiang, C.—Zhou, M.—Li, M.: Learning Transactional Behavioral Representations for Credit Card Fraud Detection. IEEE Transactions on Neural Networks and Learning Systems, 2022.

[17] Yu, W.—Ding, Z.—Liu, L.—Wang, X.—Crossley, R. D.: Petri Net-Based Methods for Analyzing Structural Security in E-Commerce Business Processes. Future Generation Computer Systems, Vol. 109, 2020, pp. 611–620, doi: 10.1016/j.future.2018.04.090.

[18] Xing, L.—Chen, Y.—Wang, X.—Chen, S.: Integuard: Toward Automatic Protection of Third-Party Web Service Integrations. NDSS, 2013.

[19] Yu, W.—Yan, C.—Ding, Z.—Jiang, C.—Zhou, M.: Analyzing E-Commerce Business Process Nets via Incidence Matrix and Reduction. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 48, 2018, No. 1, pp. 130–141.

[20] Cugola, G.—Margara, A.: Processing Flows of Information: From Data Stream to Complex Event Processing. ACM Computing Surveys, Vol. 44, 2012, No. 3.

[21] Bonino, D.—De Russis, L.: Complex Event Processing for City Officers: A Filter and Pipe Visual Approach. IEEE Internet of Things Journal, Vol. 5, 2018, No. 2, pp. 775–783, doi: 10.1109/JIOT.2017.2728089.

[22] Weidlich, M.—Ziekow, H.—Gal, A.—Mendling, J.—Weske, M.: Optimizing Event Pattern Matching Using Business Process Models. IEEE Transactions on Knowledge and Data Engineering, Vol. 26, 2014, No. 11, pp. 2759–2773, doi: 10.1109/TKDE.2014.2302306.

[23] Milosevic, Z.—Berry, A.—Chen, W.—Rabhi, F. A.: An Event-Based Model to Support Distributed Real-Time Analytics: Finance Case Study. 2015 IEEE 19th International Enterprise Distributed Object Computing Conference, 2015, pp. 122–127, doi: 10.1109/EDOC.2015.26.

[24] Liu, Y.—Wang, D.: Complex Event Processing Engine for Large Volume of RFID Data. 2010 Second International Workshop on Education Technology and Computer Science, Vol. 1, 2010, pp. 429–432, doi: 10.1109/ETCS.2010.214.

[25] Bhargavi, R.—Vaidehi, V.—Bhuvaneswari, P. T. V.—Balamurali, P.—Chandra, G.: Complex Event Processing for Object Tracking in Wireless Sensor Networks. 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Vol. 3, 2010, pp. 211–214, doi: 10.1109/WI-IAT.2010.70.

[26] Ma, Z.—Yu, W.—Zhai, X.—Jia, M.: A Complex Event Processing-Based Online Shopping User Risk Identification System. IEEE Access, Vol. 7, 2019, pp. 172088–172096, doi: 10.1109/ACCESS.2019.2955466.

[27] Jensen, K.: Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Springer Science & Business Media, 2013.

[28] Yu, W.—Jia, M.—Fang, X.—Lu, Y.—Xu, J.: Modeling and Analysis of Medical Resource Allocation Based on Timed Colored Petri Net. Future Generation Computer Systems, Vol. 111, 2020, pp. 368–374, doi: 10.1016/j.future.2020.05.010.

[29] van der Aalst, W. M.—Lohmann, N.—La Rosa, M.: Ensuring Correctness During Process Configuration via Partner Synthesis. Information Systems, Vol. 37, 2012, No. 6, pp. 574–592.

[30] Du, Y.—Li, X.—Xiong, P.: A Petri Net Approach to Mediation-Aided Composition of Web Services. IEEE Transactions on Automation Science and Engineering, Vol. 9, 2012, No. 2, pp. 429–435.

[31] Hu, H.—Liu, Y.: Supervisor Simplification for AMS Based on Petri Nets and Inequality Analysis. IEEE Transactions on Automation Science and Engineering, Vol. 11, 2014, No. 1, pp. 66–77.

[32] Wang, S.—Wang, C.—Zhou, M.: Design of Optimal Monitor-Based Supervisors for a Class of Petri Nets with Uncontrollable Transitions. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 43, 2013, No. 5, pp. 1248–1255.

[33] Macià, H.—Valero, V.—Díaz, G.—Boubeta-Puig, J.—Ortiz, G.: Complex Event Processing Modeling by Prioritized Colored Petri Nets. IEEE Access, Vol. 4, 2016, pp. 7425–7439, doi: 10.1109/ACCESS.2016.2621718.

[34] Jensen, K.—Kristensen, L. M.: Coloured Petri Nets: Modelling and Validation of Concurrent Systems. Springer Science & Business Media, 2009.

[35] Ding, W.—Wang, H.—Nan, P.—Xiao, Y.—Liu, Z.: Stock Technical Analysis System Based on Real-Time Stream Processing. 2017 10th International Symposium on Computational Intelligence and Design (ISCID), 2017.

[36] Sta, S.—Lindeberg, M.—Goebel, V.: Online Analysis of Myocardial Ischemia from Medical Sensor Data Streams with Esper. Proceedings of the First International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL 2008), 2008.

[37] Mathew, A.: Benchmarking of Complex Event Processing Engine-Esper. Dept. Comput. Sci. Eng., Indian Inst. Technol. Bombay, Maharashtra, India, Tech. Rep. IITB/CSE/2014/April/61, 2014.

Wangyang Yu received his Ph.D. degree from the Tongji University, Shanghai, China, in 2014. He is Associate Professor at the School of Computer Science, Shaanxi Normal University, Xi'an, China. His research interests include the theory of Petri nets, formal methods in software engineering, and artificial intelligence.

Zhuojing Ma received her Master's degree from the School of Computer Science, Shaanxi Normal University, Xi'an, China. Her research interests include the Petri nets theory, formal modelling of online transactions, and complex event processing.

Xiaojun Zhai is Senior Lecturer in the Embedded Intelligent Systems Laboratory at the University of Essex. He has authored/co-authored over 120 scientific papers in international journals and conference proceedings. His research interests mainly include designing and implementing digital image and signal processing algorithms, custom computing using FPGAs, embedded systems, and hardware/software co-design. He is a BCS, IEEE Senior member and HEA Fellow.

Yuke Zhou received his Ph.D. degree in geographic information systems from the Institute of Geographical Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing, China, in 2013. He is currently Associate Researcher whose research interests include high-performance computing, distributed systems, and remote sensing of ecosystem environments.

Weiwei Zhou received her B.Sc. degree in business administration from the Shandong University of Science and Technology, Qingdao, China, in 2021. She is currently Lecturer at the School of Business at Shandong Yingcai University, Jinan, China. Her current research interests include e-commerce, innovation, and entrepreneurship management, etc.

Yuan Liu is pursuing his Master's at the School of Computer Science, Shanxi Normal University, Xi'an, China. His research interests include the theory of Petri nets, process mining, machine learning, and CEP.