frontiers | Research Topics

Explainable, trustworthy and responsive intelligent processing of biological resources integrating data, information, knowledge, and wisdom – volume II

Edited by
Yucong Duan and Yungang Xu

Published in
Frontiers in Genetics

ISSN 1664-8714
ISBN 978-2-83251-329-3
DOI 10.3389/978-2-83251-329-3

About Frontiers

Frontiers is more than just an open access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers journal series

The Frontiers journal series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the *Frontiers journal series* operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews. Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the *Frontiers journals series*: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area.

Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers editorial office: frontiersin.org/about/contact

Explainable, trustworthy and responsive intelligent processing of biological resources integrating data, information, knowledge, and wisdom – volume II

Topic editors

Yucong Duan — Hainan University, China

Yungang Xu — Xi'an Jiaotong University, China

Citation

Duan, Y., Xu, Y., eds. (2023). *Explainable, trustworthy and responsive intelligent processing of biological resources integrating data, information, knowledge, and wisdom – volume II*. Lausanne: Frontiers Media SA.

doi: 10.3389/978-2-83251-329-3

Table of contents

frontiers | Frontiers in Genetics

TYPE Editorial
PUBLISHED 04 January 2023
DOI 10.3389/fgene.2022.1114441

OPEN ACCESS

EDITED AND REVIEWED BY
Richard D. Emes,
University of Nottingham,
United Kingdom

*CORRESPONDENCE
Yucong Duan,
duanyucong@hotmail.com
Yungang Xu,
yungang.xu@xjtu.edu.cn

SPECIALTY SECTION
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

RECEIVED 02 December 2022
ACCEPTED 05 December 2022
PUBLISHED 04 January 2023

CITATION
Duan Y and Xu Y (2023), Editorial: Explainable, trustworthy and responsive intelligent processing of biological resources integrating data, information, knowledge, and wisdom—Volume II. *Front. Genet.* 13:1114441. doi: 10.3389/fgene.2022.1114441

Editorial: Explainable, trustworthy and responsive intelligent processing of biological resources integrating data, information, knowledge, and wisdom—Volume II

Yucong Duan[1]* and Yungang Xu[2]*

[1]College of Computer Science and Technology, Hainan University, Haikou, China, [2]School of Basic Medical Science, Xi'an Jiao Tong University, Xi'an, China

KEYWORDS

DIKW, DIKW graph, explainability and interpretability, trustworthy AI, responsive ability, knowledge, information retreival, data comprehension

Editorial on the Research Topic
Explainable, trustworthy and responsive intelligent processing of biological resources integrating data, information, knowledge, and wisdom-volume II

The increasing practice of Artificial Intelligence (AI) in biological and biomedical resources faces challenges of the explainable, trustworthy, responsive AI processing of multi-modal, intertwined, interactive biological and biomedical data, which requires the integration of data, information, knowledge, wisdom and purpose (DIKWP) across objective content and subjective cognition/purpose. Transformations among data, information, knowledge and wisdom open possibilities to comply with uncertainties originating in the incompleteness of data samples, insufficiency of information, vulnerability of invalid knowledge and imbalanced wisdom strategies, towards achieving more precise, robust, reproducibility and less repeated operations of data Research Topic and information synthesis, and more comprehensive knowledge reproducibility through multiple sources reasoning and abstraction. Moreover, alongside the COVID emergency, more and more attention is focused on balancing social welfare, cultural moralities, and the biological practices involving privacy-preserving data Research Topic and legal information usage, under rapid iterations of international political and technical negotiations, towards a responsible AI-enabled AI governance implementing justice, transparency and fairness. This Research Topic aimed to collect the latest research efforts devoted to building capabilities of integration and transformation of multi-modal data, information, knowledge and wisdom in an integrated semantic understanding space unifying subjective purposes and objective

formalism, to validate data, retrieve information, abstraction on information to attain knowledge hypotheses, and balanced optimization. In total, nine articles including one review article were published in Frontiers in Genetics.

In the review article Wang et al. proposed a systemic construction towards the mutual incentive among the "social-biological-technological triangle" interaction in hope of interpreting the success and lessons of AI participation in the prevention and treatment of COVID-19.

The Research Topic published eight original research papers that cover a wide range of efforts in applying AI technology in multiple biological and biomedical data sources. Three papers focus on explainable intelligence crossing data graph, information graph and knowledge graph, led by Geng et al., Zhao et al. and Diao et al., respectively. In the article towards addressing the information overloaded problem for personalized recommendation/prescription, Geng et al. proposed a compliment method for integrating subjective sentimental information in the information graph form and objective feature representation in knowledge graph based on representational learning *via* triple-autoencoder. In the article towards leveraging current data intensive or statistical based data graphs into logically explainable knowledge graph in medical industry, Zhao et al. proposed a multi-layers entity extraction architecture to extract object-level entities with "object-attribute" dependencies in the data graph for construction of logic in high-quality medical knowledge graphs based real electronic clinical records. In the article towards constructing an error-avoiding and effort-saving solution in discovering bioinformatics workflow fragments and leveraging historical usages of related activities/services, Diao et al. proposed a workflow Knowledge Graph to unifying common types of data entities and data structural relationship in the data graph of service invoking network, and the implicit information of the information graph in both individual user's requirements and service communities.

Two article focus on hybrid intelligence resource merging mechanisms crossing incomplete data, inconsistent information and not validated knowledge, led by Wang et al. and Yu and Duan respectively. In the article towards objectifying the knowledge level inconsistency and redundancy originating in the information subjectivity inputted by various biomedical experts, Wang et al. proposed a data-information-knowledge merging approach for biomedical ontology matching *via* a hybrid graph attention network. In the article towards addressing sparsity of data and the cold start of recommendation in prediction of Quality of Services, Yu and Duan proposed a GRU-GAN based learning uniformity over quality data and user characteristic information.

Additionally, three articles presented a trusted resource scheduling method, a miRNA prediction algorithm, and a biological adaptation mechanism, respectively. In the article towards realizing reliable and credible intelligent processing of biological resources, Yu et al. designed a composite service scheduling model under the containers instance mode hybridizing reservation and on-demand. In the article towards understanding miRNAs' cellular function information and knowledge roles in regulating gene expression, Min et al. proposed to predict essential miRNAs using XGBoost framework with Classification and Regression Trees on various types of sequence-based information features. In the article of towards enhancing the diversity of self-replicating structures, Xu et al. proposed an active self-adaptations in comparison with the passive mechanism through introduction of knowledge rules.

Author contributions

YD and YX are guest associate editors of the Research Topic and wrote this editorial.

Funding

YD is supported by Hainan Province Key R&D Program No.ZDYF2022GXJS007, ZDYF2022GXJS010, Hainan Province Higher Education and Teaching Reform Research Project No.Hnjg2021ZD-3, Natural Science Foundation of Hainan Province No.620RC561 and Hainan Province Key Laboratory of Meteorological Disaster Prevention and Mitigation in the South China Sea No.SCSF202210. YX is supported by the National Natural Science Foundation of China No. 62171365.

Acknowledgments

We thank the authors for their valuable contributions and reviewers for their efforts to guarantee the high quality of this Research Topic, with especially thanks to the editorial board of the journal of Frontiers in Genetics.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

frontiers in Genetics

METHODS
published: 28 March 2022
doi: 10.3389/fgene.2022.877409

XGEM: Predicting Essential miRNAs by the Ensembles of Various Sequence-Based Classifiers With XGBoost Algorithm

Hui Min[1], *Xiao-Hong Xin*[1], *Chu-Qiao Gao*[1], *Likun Wang*[2]* *and Pu-Feng Du*[1]*

[1]College of Intelligence and Computing, Tianjin University, Tianjin, China, [2]Institute of Systems Biomedicine, Department of Pathology, School of Basic Medical Sciences, Beijing Key Laboratory of Tumor Systems Biology, Peking-Tsinghua Center of Life Sciences, Peking University Health Science Center, Beijing, China

MicroRNAs (miRNAs) play vital roles in gene expression regulations. Identification of essential miRNAs is of fundamental importance in understanding their cellular functions. Experimental methods for identifying essential miRNAs are always costly and time-consuming. Therefore, computational methods are considered as alternative approaches. Currently, only a handful of studies are focused on predicting essential miRNAs. In this work, we proposed to predict essential miRNAs using the XGBoost framework with CART (Classification and Regression Trees) on various types of sequence-based features. We named this method as XGEM (XGBoost for essential miRNAs). The prediction performance of XGEM is promising. In comparison with other state-of-the-art methods, XGEM performed the best, indicating its potential in identifying essential miRNAs.

Keywords: essential miRNA, CART, XGBoost, sequence features, ensemble classifier

OPEN ACCESS

Edited by:
Yungang Xu,
Xi'an Jiaotong University, China

Reviewed by:
Pengmian Feng,
North China University of Science and Technology, China
Zhuhong You,
Northwestern Polytechnical University, China

***Correspondence:**
Likun Wang
wanglk@bjmu.edu.cn
Pu-Feng Du
pdu@tju.edu.cn

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *16 February 2022*
Accepted: *07 March 2022*
Published: *28 March 2022*

Citation:
Min H, Xin X-H Gao C-Q Wang L and Du P-F (2022) XGEM: Predicting Essential miRNAs by the Ensembles of Various Sequence-Based Classifiers With XGBoost Algorithm. Front. Genet. 13:877409. doi: 10.3389/fgene.2022.877409

INTRODUCTION

MicroRNAs (miRNAs) are functional non-coding RNAs of ~22 nt in length. miRNAs are involved in regulating gene expressions (He and Hannon, 2004) in animals and plants. They have diverse expression patterns and regulate many biological processes, including cell proliferation (Cao et al., 2022), cell differentiation (Martin et al., 2016), cell apoptosis (Zhang et al., 2019), fat metabolism (Nematbakhsh et al., 2021), and development of animals and plants (Zhang et al., 2018). They are also related to many complex diseases (Wojciechowska et al., 2017), including many types of tumors (Zhang et al., 2007; Lee and Dutta, 2009; Fridrichova and Zmetakova, 2019).

lin-4 (Lee et al., 1993) was the first miRNA to be discovered, followed by *let*-7 (Reinhart et al., 2000). The regulatory roles of miRNAs have been widely studied (Bartel, 2004, 2018). Although miRNAs are small in length, their cellular role is important. Knocking out or knocking down some miRNA genes will result in lethal or infertile phenotypes (Bartel, 2018). These miRNAs genes are thought to be essential for the organism to live or develop. With the progress of miRNA gene annotations, many computational methods were developed to find miRNA genes in the genome (Wang et al., 2019). However, this resulted in many annotated miRNA genes in the database with little or no functional understanding (Bartel, 2018; Ru et al., 2019). As a basis toward the understanding of gene cellular functions, a gene should be determined if it is essential or not (Zeng et al., 2018; Campos et al., 2019).

In the context of miRNA genes, there are two categories of methods for identifying essential miRNAs: experimental methods and computational predictions. The experimental methods usually perform gene knockout or gene expression knockdown experiments on animal or plant models. By observing the phenotypes, the essentiality of the gene in question will be determined (Larrimore and Rancati, 2019). For example, Ahmed et al. (2017) reported that the *miR*-7a-2 is an essential miRNA gene by knocking out the *miR*-7a-2 gene in the mouse genome to observe the result that it caused infertility. Since the experimental methods are inevitably time-consuming and labor-intensive, computational predictions are always considered as alternative approaches or, at least, beneficial supplements. Computational prediction methods usually combine machine learning algorithms with statistical features of genomic sequences and structures to construct classifiers. Currently, there is no genome-wide clear set of essential miRNA genes. Therefore, constructing such machine learning-based predictors for essential miRNA genes is still a challenging task. As far as we know, only a handful of studies tried to predict essential miRNAs.

Ru et al. (2019) carried out a study in computationally predicting essential miRNAs. They collected 85 essential miRNAs from the literature (Bartel, 2018). By compensating 88 non-essential miRNAs from their own random selection, they presented a benchmarking dataset for computationally predicting essential miRNAs. They achieved a promising result by applying a simple voting scheme in the ensemble of multiple classifiers. Song et al. (2019) collected 77 essential miRNAs from the same literature (Bartel, 2018). They proposed the miES method based on the logistic regression algorithm. Yan et al. (2020) developed a third method based on the same 77 essential miRNAs, namely, PSEM, for the prediction of essential miRNAs in the mouse genome.

In this study, we applied the XGBoost (extreme gradient boosting) method (Chen and Guestrin, 2016) with classification trees to construct our predictor on various sequences and structural features. By optimizing features and parameters, we achieved better prediction performances than existing studies. We named our method as XGEM (XGBoost for essential miRNAs). We provided genome-wide prediction results in mice as a supplemental annotation to the mouse genome.

MATERIALS AND METHODS

Experimental Data

We considered the dataset from Ru's work (Ru et al., 2019), which contains 85 essential and 88 non-essential pre-miRNA sequences. We also obtained the dataset of miES (Song et al., 2019) and PESM (Yan et al., 2020) work, which contains 77 essential miRNAs. To compose a working dataset, we randomly picked up 77 non-essential miRNAs as negative samples for the miES and PESM dataset. We noted the former dataset as Ru's dataset and the latter dataset as the miES-PESM dataset. Ru's dataset was used for training and testing the XGEM method, while the miES-PESM dataset was used only for performance comparison.

Feature Extraction Methods

Five sequence feature extraction methods were incorporated in our work. They are *k*-mer frequencies, sequence mismatch features, subsequence features, PseDSSPC (pseudo-distance structure status pair composition), and triplet compositions. BioSeq-Analysis 2.0 (Liu et al., 2019) and repRNA (Liu et al., 2016b) were used to generate these features. Although the algorithms for generating these features have been elaborated in various works of the literature (Chen et al., 2015, 2018; Liu et al., 2016a, 2019; Zhang et al., 2021), we briefly described them here for the convenience of readers.

Given an RNA sequence R with length l, it can be noted as follows:

$$R = r_1 r_2 ... r_l, \tag{1}$$

where r_i (i = 1, 2, 3, ... l) ∈ {A, C, G, U} is the i-th residue in R.

The k-mer frequencies are the appearance frequency of 4^k type's k consecutive nucleotides. The sequence R is separated into l–k + 1 k-mers, which are $r_1 r_2 \ldots r_k$, $r_2 r_3 \ldots r_{k+1}$, ..., and $r_{l-k+1} r_{l-k+2} r_l$. We noted the k-mer frequency as a vector of 4^k dimensions (Wei et al., 2014), which can be noted as follows:

$$\mathbf{F}_1(k) = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,4^k} \end{bmatrix}^T, \tag{2}$$

where $f_{1,i}$ (i = 1, 2, ... , 4^k) is the frequency of the i-th type of k-mer, and T is the transpose operator.

The mismatch feature is proposed by Leslie et al. as an alternative method of k-mer frequencies (Leslie et al., 2004). The method considers inaccurate matching and calculates the number of occurrences of k consecutive nucleotides that differ by at most m mismatches (m = 0, 1, ... , k-1). We define the mismatch feature vector as follows:

$$\mathbf{F}_2(k,m) = \left(\sum_{j=0}^{m} c_{1,j} \quad \sum_{j=0}^{m} c_{2,j} \quad \cdots \quad \sum_{j=0}^{m} c_{4^k,j} \right)^T, \tag{3}$$

where $c_{i,j}$ (i = 1, 2, ... , 4^k and j = 0, 1, ... , m) is the number of occurrence of the i^{th} type k-mer in sequence R with exactly j mismatches.

The subsequence feature is a method that allows non-continuous matching, which considers more matching situations (Lodhi et al., 2002). The value of the feature vector is determined by the number of occurrences of the subsequence and a decay factor $\delta \in [0, 1]$. The subsequence feature vector of sequence R is defined as follows:

$$\mathbf{F}_3(k,m) = \left(\sum_{a_1} \delta^{l(a_1)} \quad \sum_{a_2} \delta^{l(a_2)} \quad \cdots \quad \sum_{a_{4^k}} \delta^{l(a_{4^k})} \right)^T, \tag{4}$$

where a_i (i = 1, 2, ... , 4^k) is a subsequence in R with possibly non-contiguous matching to the i^{th} type of k-mer, and l (a_i) a length function can be defined as follows:

$$l(a_i) = \begin{cases} 0 & a_i \text{ is a contiguous matching of the } i-\text{th type of } k-\text{mer} \\ |a_i| & \text{otherwise} \end{cases}. \tag{5}$$

|.| is the operator to calculate the length of a string.

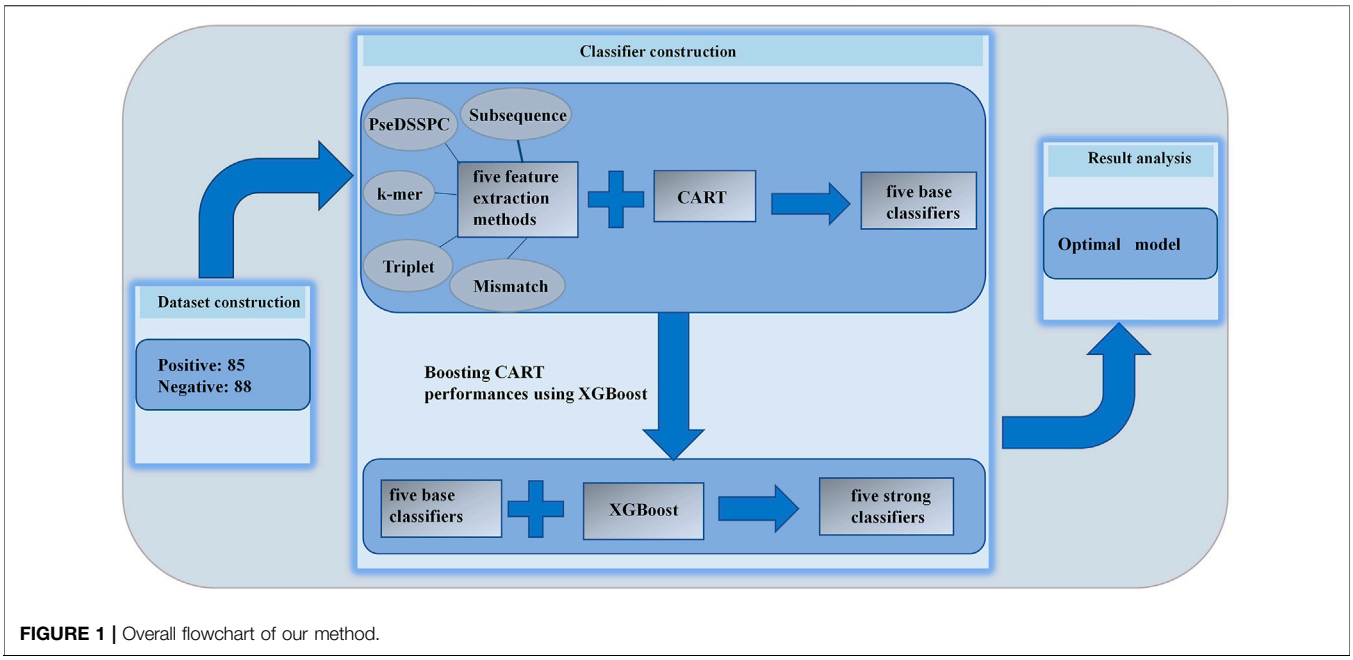


FIGURE 1 | Overall flowchart of our method.

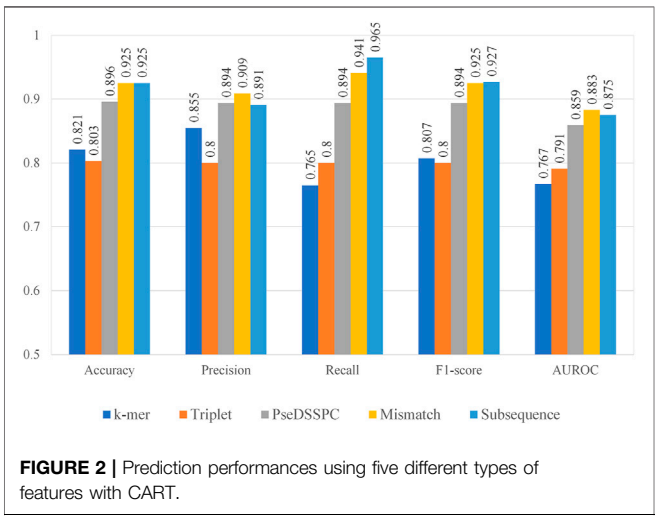


FIGURE 2 | Prediction performances using five different types of features with CART.

Triplet feature is a combination of the primary sequence and secondary structural information of RNA. It was proposed by Xue et al.(2005). By using the ViennaRNA package (Lorenz et al., 2011), we can estimate the secondary structure of R as follows:

$$S = s_1 s_2 s_3 \cdots s_l, \tag{6}$$

where s_i (i = 1, 2, .., l) ∈ { ' (', ')', '.' } denotes the secondary structure status of the i^{th} residue. The " (' and ')" represent the residue in a pairing status, while "." represents the unpairing status. By ignoring the difference between " (' and ')", there are eight possible structural statuses of a triplet. Combining the structural status and the centered nucleotide of a triplet, 32 types of possible structural triplets can be obtained. Therefore, a 32-dimensional vector can be constructed to describe the appearance frequency of all structural triplets, which can be noted as follows:

$$\mathbf{F}_4 = \left[f_{4,1} \quad f_{4,2} \quad \cdots \quad f_{4,32} \right]^T, \tag{7}$$

where $f_{4,i}$ (i = 1, 2, . . . , 32) is the normalized frequency of the i-th structural triplet.

PseDSSPC was proposed by Liu *et al.* (Liu et al., 2016a). It represents the RNA sequence by considering both local and global information of secondary structures. Let t_i (i = 1, 2, . . . , l) ∈ {A, C, G, U, A-U, U-A, G-C, C-G, G-U, and U-G} be the structural status of the i-th residue, where A, C, G, and U represent the four types of unpaired residues, while A-U, U-A, G-C, C-G, G-U, and U-G represent the six paired status. For every t_i, its free energy e (t_i) can be calculated. We first computed the raw appearance frequency of each of the 10 structural status, which can be noted as $g_{5,1}$, $g_{5,2}$, . . . $g_{5,10}$. Given a parameter d, we can calculate the appearance frequency of all structural status pairs with a distance in the range [1, d]. These can be noted as $g_{5,11}$, $g_{5,12}$, . . . , $g_{5,110}$, $g_{5,111}$, $g_{5,112}$, . . . , $g_{5,210}$, . . . , $g_{5,10+(d\text{-}1)100+1}$, $g_{5,10+(d\text{-}1)100+2}$, . . . , $g_{5,10+100d}$. After that, with a lag parameter λ, correlation coefficients can be computed for the serial of free energy values. The k^{th} tier correlation coefficient can be defined as follows:

$$g_{5,10+100d+k} = \frac{1}{l-k} \sum_{i=1}^{l-k} [e(t_i) - e(t_{i+k})]^2, \tag{8}$$

where k = 1, 2, . . . , λ.

With all aforementioned definitions, we can construct PseDSSPC features as follows:

$$\mathbf{F}_5 = \left[f_{5,1} \quad f_{5,2} \quad \cdots \quad f_{5,10+100d+\lambda} \right]^T, \tag{9}$$

where T is the transpose operator,

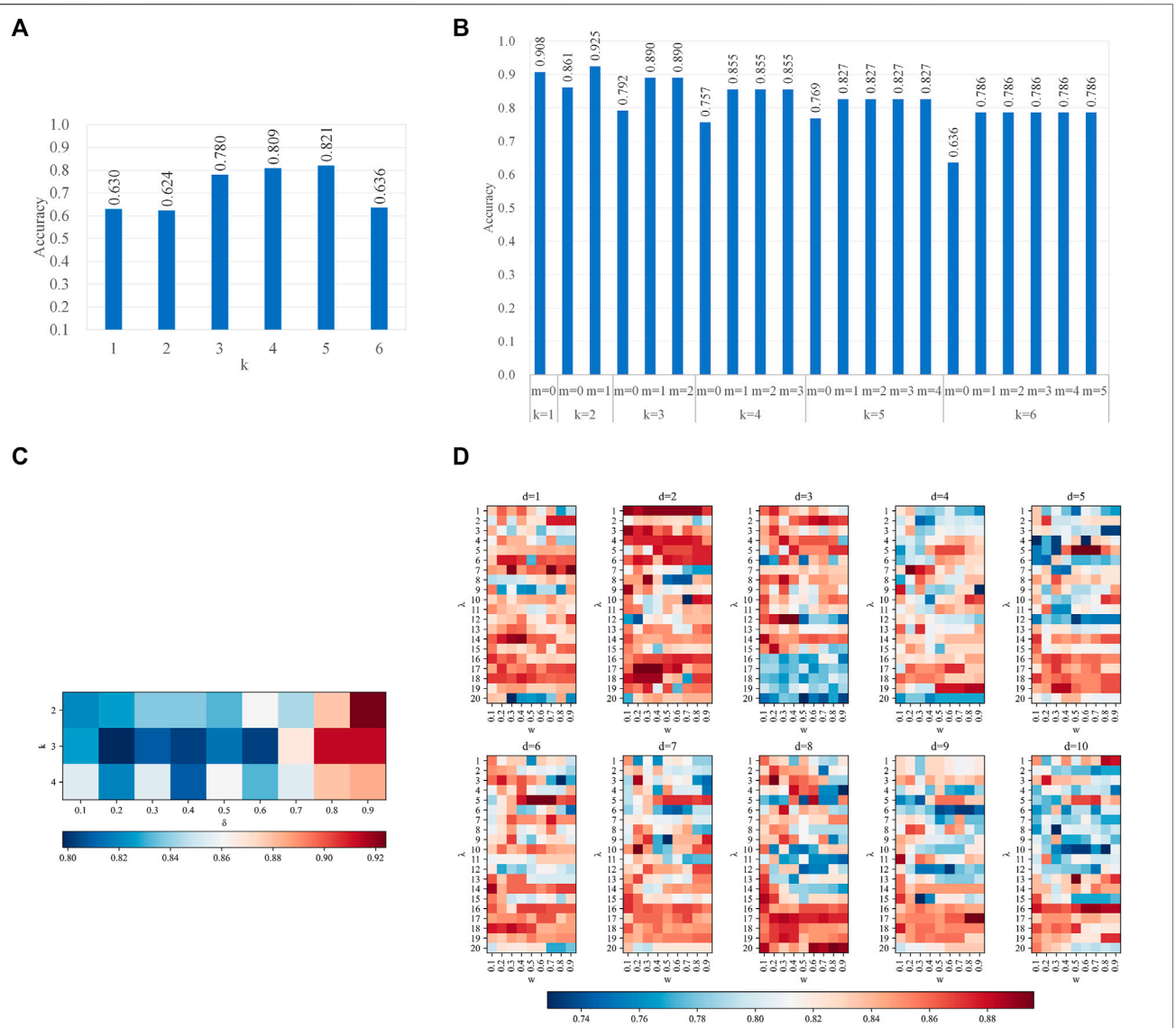


FIGURE 3 | Parameter effects on CART with different types of features. Parameters of features are scanned **(A)**. *k*-mer features **(B)**; mismatch features **(C)**; subsequence features **(D)**; PseDSSPC features. In **(A)** and **(B)**, the vertical axis is the accuracy in leave-one-out cross-validation. In **(C,D)**, the heat color represents the accuracy in leave-one-out cross-validation. The optimized parameter is $k = 5$ for the *k*-mer features, $k = 2$ and $m = 1$ for the mismatch features, $k = 2$ and $\delta = 0.9$ for the subsequence features, and $d = 5$, $\lambda = 5$, and $w = 0.5$ for the PseDSSPC features.

$$f_{5,i} = \begin{cases} \dfrac{g_{5,i}}{1 + d + w \sum\limits_{k=10+100d+1}^{10+100d+\lambda} g_{5,k}} & 1 \le i \le 10 + 100d \\ \dfrac{w g_{5,i}}{1 + d + w \sum\limits_{k=10+100d+1}^{10+100d+\lambda} g_{5,k}} & 10 + 100d + 1 \le i \le 10 + 100d + \lambda \end{cases}, \quad (10)$$

and w is a balancing parameter.

XGBoost With Classification Trees as Base Classifiers

We used CART (Classification and Regression Trees) with the Gini index as the purity function (Grajski et al., 1986) to create base classifiers in this work. Given a sample set D, the Gini function is defined as follows:

$$G(D) = \sum_{i=1}^{k} p_i (1 - p_i) = 1 - \sum_{i=1}^{k} p_i^2, \quad (11)$$

where k is the number of classes in the set, and p_i is the proportion of the i^{th} class.

Considering an attribute α, the set D is divided into several subsets according to different values of α. The purity at this branching node is defined as follows:

$$I(D, \alpha) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} G(D_j), \quad (12)$$

where v is the number of subsets, D_j is the j-th subset, and $|.|$ is the cardinal operator of a set.

FIGURE 4 | Parameter effects on CART with different types of features. Parameters of CART are scanned. The accuracy, F1-score, and AUROC are presented in each panel. **(A)**, **(B)**, and **(C)** are scanning CART parameters *S*, *D*, and *M* on *k*-mer features, respectively, and **(D)**, **(E)**, and **(F)** are scanning CART parameters *S*, *D*, and *M* on mismatch features, respectively; **(G)**, **(H),** and **(I)** are scanning CART parameters *S*, *D*, and *M* on subsequence features, respectively; **(J)**, **(K)**, and **(L)** are scanning CART parameters *S*, *D*, and *M* on PseDSSPC features, respectively; **(M)**, **(N)**, and **(O)** are scanning CART parameters *S*, *D*, and *M* on the triplet features, respectively. The best parameter for *k*-mer features is *S* = 'best', *D* = 8, and *M* = 490. The best parameter for mismatch features is *S* = 'best', *D* = 4, and *M* = 13. The best parameter for subsequence features is *S* = 'best', *D* = 3, and *M* = 14. The best parameter for PseDSSPC features is *S* = 'best', *D* = 3, and *M* = 460. The best parameter for the triplet features is *S* = 'best', *D* = 4, and *M* = 30.

TABLE 1 | Performance of the five strong classifiers.

Models	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUROC[a] (%)
k-mer	82.7	80.9	84.7	82.8	86.4
Mismatch	96.0	94.3	97.6	96.0	96.4
Subsequence	93.1	94.1	94.1	94.1	97.3
PseDSSPC	90.8	91.6	89.4	90.4	94.8
Triplet	80.9	80.9	80.0	80.4	85.3

[a]*AUROC is the area under a receiver operating characteristic curve.*

XGBoost (Chen and Guestrin, 2016) was used to create ensembles for boosting performances of classification trees.

Performance Measures

Four statistics, including accuracy (*Acc*), precision (*Pre*), recall (*Rec*), and F1-score (*F*), are used to quantitively describe the performance of our method. They are defined as follows:

$$Acc = \frac{TN + TP}{FN + FP + TN + TP}, \tag{13}$$

$$Pre = \frac{TP}{TP + FP}, \tag{14}$$

$$Rec = \frac{TP}{TP + FN}, \tag{15}$$

$$F = \frac{2Pre \cdot Rec}{Pre + Rec}, \tag{16}$$

where *TP*, *TN*, *FP*, and *FN* are the number of true positives, true negatives, false positives, and false negatives, respectively. We also used the area under the receiver operating characteristic (AUROC) curve to measure the performance of our model.

Parameter Calibration

We used a grid search strategy with leave-one-out cross-validation to find the optimal parameters. For *k*-mer features, we scanned k = 1, 2, 3, 4, 5, and 6. For mismatch features, we scanned k = 1, 2, 3, 4, 5, and 6 and $m \in [0, k\text{-}1]$ with a step of 1. For subsequence features, we scanned k = 2, 3, and 4, and $\delta \in [0.1, 0.9]$ with a step of 0.1. In PseDSSPC, we scanned $d \in [1,10]$ with a step of 1, $\lambda \in [1, 20]$ with a step of 1 and $w \in [0.1, 0.9]$ with a step of 0.1.

Different combinations of parameter values in CART and XGBoost are explored. We adjusted three parameters in the CART algorithm, including the randomness of branching (*S*), the maximum depth (*D*), and the maximum number of features (*M*). We scanned $S \in$ ["best", "random"], $D \in [3,10]$ with a step of 1 and $M \in [3, n]$ with a step of 1, where n is the number of sample features. We adjusted *S*, *D*, and *M* in order; when the former parameters are being scanned, the latter ones are set as default values. The best value of the former is applied to the latter parameter adjustment. We adjusted four parameters in XGBoost, including the number of trees (*T*), the learning rate (*R*), the maximum depth of trees (*D*), and the regularization parameter (λ). We scanned $T \in [50, 500]$ with a step of 10, $R \in [0.1, 0.5]$ with a step of 0.02, $D \in [3, 10]$ with a step of 1, and $\lambda \in [0, 2]$ with a step of 0.1. Similar strategies to the CART parameter optimization were applied.

System Implementation

The CART and XGBoost algorithms are implemented using Python with the scikit-learn package. The whole flowchart of this work is illustrated in **Figure 1**.

RESULTS AND DISCUSSIONS

Performance Analysis by CART

We combined each of the five feature extraction methods with CART. We optimized the parameters of each kind of features. The best performances of each type of features can be found in **Figure 2**. The evaluation was performed on Ru's dataset. Leave-one-out cross-validation protocol was applied on each type of features. The entire record of the parameter optimization process can be found in **Supplementary Tables S1–S5**.

From **Figure 2**, the subsequence features seem to have the best performances among the five. It has the highest or second to the highest value in terms of all performance measures. On the contrary, the performances of *k*-mer features and triplet features seem not as high as the others. The *k*-mer features have lowest performance values in terms of recall and the AUROC. The triplet features have the lowest performance values in terms of accuracy, precision, and F1-score. However, the precision value of *k*-mer and the recall value of triplet features are still competitive, which make them still worth a further boosting analysis. It should be noted that the PseDSSPC features, which by design would preserve most of the sequence information, did not give outstanding performances. This may be the result of the CART classifier, which cannot sufficiently utilize the information in this form.

With the optimal features, we analyzed the effect of different parameters in two steps. The first step is to analyze the effect of parameters in features, the latter one for the parameters in CART. When we performed the first step analysis, the parameters in the second step were fixed as their optimal values and vice versa. **Figure 3** recorded the effects of parameters on all type of features. On all four types of features, which have at least one parameter each, the prediction accuracy peaks at some combinations of parameters, while it valleys with other combinations. Therefore, the parameters of features affect the performances. **Figure 4** recorded the effects of CART parameters on all types of features. The peaks of the parameter *D* are the most significant. Although the parameter *M* causes the most fluctuation on performances, it is generally a random oscillation without easily observable patterns. Due to limited

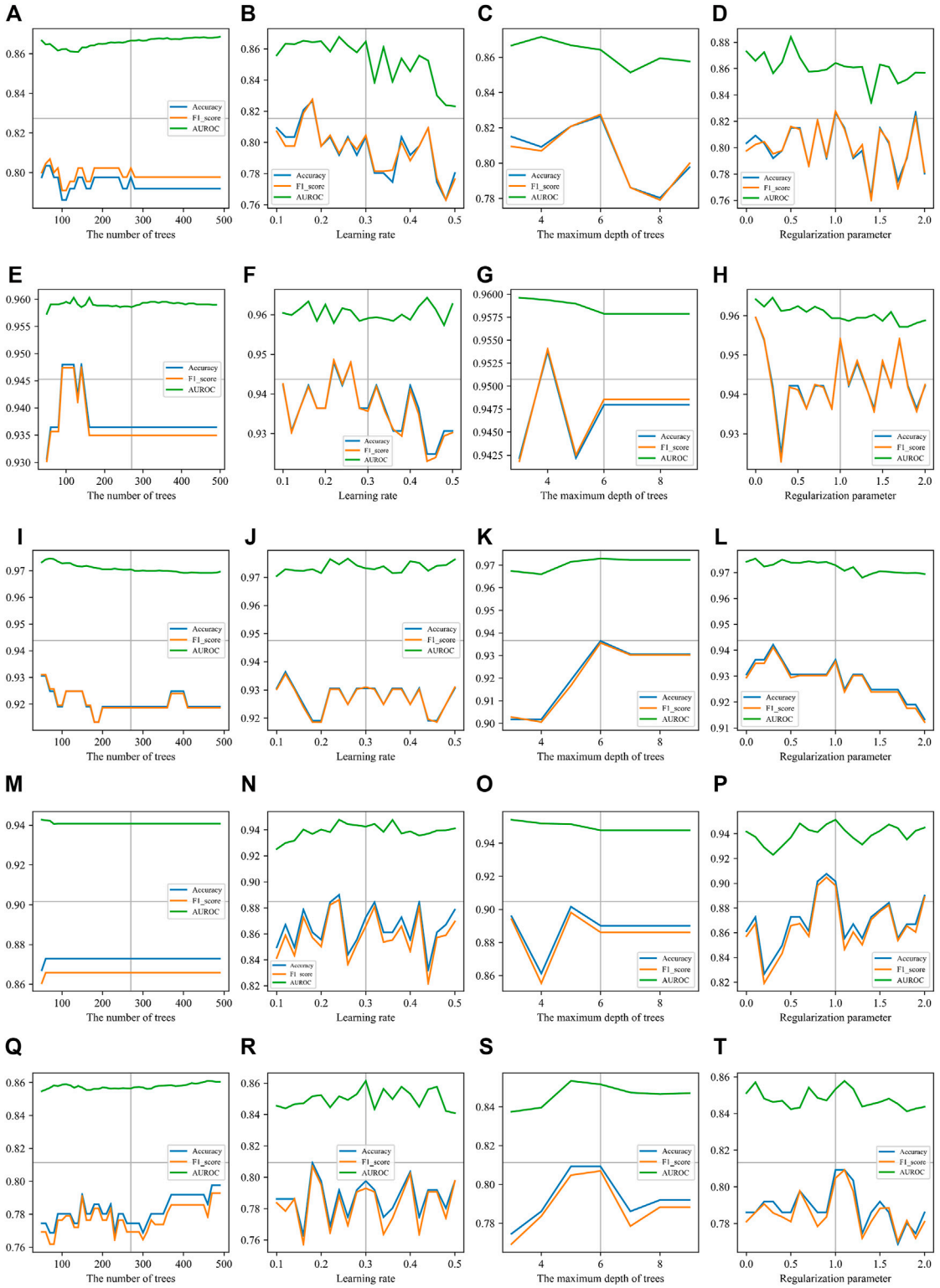


FIGURE 5 | Parameter effects on XGBoost with different types of features. Parameters of XGBoost are scanned. The accuracy, F1-score, and AUROC are presented in each panel. The number of trees **(T)**, the learning rate **(R)**, the maximum depth of trees **(D)**, and the regularization parameter (λ) are scanned on each type of sequence features. **(A), (B)**, **(C)**, and **(D)** are scanning parameters on *k*-mer features. The best parameter values are $T = 60$, $R = 0.18$, $D = 6$, and $\lambda = 1$. **(E)**, **(F)**, **(G)**, and **(H)** are scanning parameters on mismatch features. The best parameter values are T = 80, R = 0.22, D = 4, and $\lambda = 0$. **(I), (J)**, **(K)**, and **(L)** are scanning parameters on subsequence features. The best parameter values are T = 50, R = 0.12, D = 6, and $\lambda = 0.3$. **(M)**, **(N)**, **(O)**, and **(P)** are scanning parameters on Pse-DSSPC features. The best parameter values are T = 60, R = 0.24, D = 5, and $\lambda = 0.9$. **(Q) (R)**, **(S)**, and **(T)** are scanning parameters on triplet features. The best parameter values are T = 500, R = 0.18, D = 5, and $\lambda = 1$.

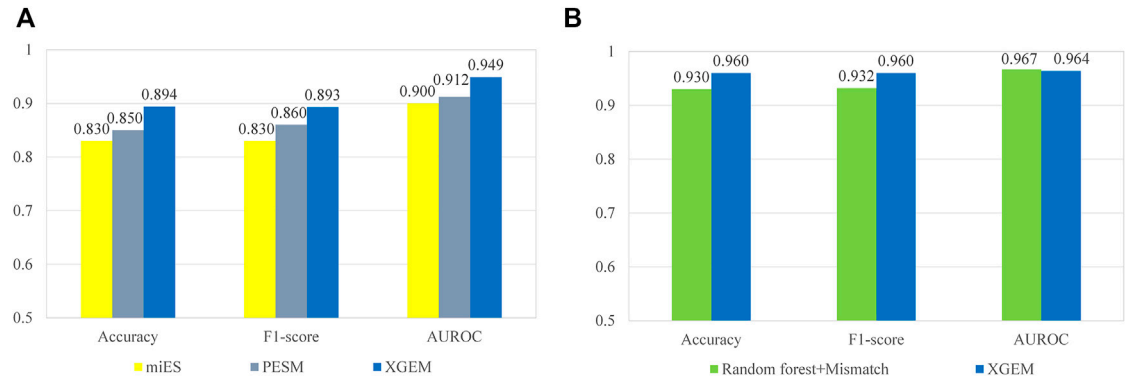


FIGURE 6 | Comparison of different methods on mouse pre-miRNA datasets. The accuracy, F1-score, and AUROC are compared. **(A)** A comparison between the XGEM, miES, and PESM method on the miES-PESM dataset; **(B)** A comparison between XGEM and Ru's work on Ru's dataset.

figure panel spaces, we only present a subset of performance measures in the figures. As we have mentioned, a comprehensive and quantitative record can be found in **Supplementary Tables S1–S5**.

Boosting CART Performances Using XGBoost

We applied XGBoost on the CART classifiers with each of the five types of features. The parameters of XGBoost are optimized to get the best AUROC. Leave-one-out cross-validations were performed on Ru's dataset. The prediction performances of the best boosted classifiers are listed in **Table 1**.

According to **Table 1**, the subsequence features achieved 97.3% AUROC after boosted by XGBoost, which is the highest AUROC among all five models. However, its performances in terms of other measures are not as high as the mismatch features. The mismatch features achieved the best values in accuracy, precision, recall, and F1-score. Therefore, the mismatch features and the subsequence features with XGBoost are better choices than the other three for predicting essential miRNAs.

Similar to the analysis on non-boosted CART classifiers, we performed an analysis to see the results with different XGBoost parameter values. **Figure 5** gives the details of all results when the parameters are adjusted. Due to limited space in the figure panels, we only presented three performance measures. Full records can be found in **Supplementary Table S6** . All curves in **Figure 5** show that the AUROC is just slightly affected by the parameters of XGBoost. The accuracy and F1-score ride the same tides when parameters are turned. Because of the theoretical relationship between F1-score and the accuracy, this observation indicated that the classifier is boosted in a balanced manner by XGBoost. This is an expected behavior of a good boosting framework on an informative and balanced training dataset.

Independent Dataset Test

We selected mismatch features with XGBoost and subsequence features with XGBoost as the optimal models. We tested the feasibility of the two models in predicting potential essential miRNAs. We collected 16 mouse pre-miRNAs from various works of the literature, which had no overlap with our training dataset, as an independent testing dataset (**Supplementary Table S7**). Among them, eight were essential, and the others were non-essential. On this testing dataset, the mismatch features with XGBoost achieved 90.6% AUROC. The subsequence features with XGBoost achieved 81.2% AUROC. Therefore, we believe that the mismatch features with XGBoost is the one best choice for predicting essential miRNAs. We named this method XGEM (XGBoost for essential miRNAs).

Genome-wide Prediction

We downloaded all 1,234 mouse pre-miRNA sequences from the miRbase (Kozomara et al., 2019). The 85 essential miRNAs and 88 non-essential miRNAs in the training dataset were removed. The 16 sequences in the testing dataset were also removed, leaving 1,045 sequences with unknown essentiality. XGEM was applied to create predictions for all of them. The results are recorded in **Supplementary Table S8**. It can provide guidance for the study of miRNA biological function experiments. It should be noticed that XGEM was trained on balanced datasets. However, the real world is highly imbalanced. Therefore, false positives are inevitable in the prediction results. But this does not diminish the value of the results as the prediction shrinks the range of potential essential miRNAs to a much smaller scale, which is exactly the purpose of computational predictions.

Comparison With State-of-the-Art Methods

We compared XGEM to all existing state-of-the-art methods, including Ru's work (Ru et al., 2019), miES (Song et al., 2019), and PESM (Yan et al., 2020).

The comparisons with miES and PESM were performed on the miES-PESM dataset. A 50-time repetition of 5-fold cross-validation was performed by all three methods on the same dataset. The repetition was used to eliminate inevitable randomness in the process of 5-fold cross-validation. The average performance values of the 50-time repetition were compared. The comparison with Ru's work was performed on Ru's dataset. Leave-one-out cross-validation was performed by both methods on the same dataset. The comparison details are depicted in **Figure 6**. XGEM

performed the best in both comparisons. Although the benefits of XGEM is not large enough for us to claim that XGEM is definitely a better choice in predicting essential miRNAs, it is enough to state that XGEM is a better or at least comparable method to all state-of-the-art methods.

CONCLUSION

Determining essentiality of non-coding genes is an important and fruitful research area, particularly for computational biology. In this article, we developed XGEM, which is a computational tool for predicting essential miRNAs. We evaluated the performance of XGEM in the mouse genome, with comparison to other state-of-the-art methods. The results indicated that XGEM has a potential to identify essential miRNAs. This is useful in understanding the biological functions of miRNA genes. We plan to establish a web server for hosting the implementation of XGEM. Due to the availability of limited resources currently, we will do this as a future work. In addition, the technology for developing XGEM can be extended to identify other types of essential non-coding genes, particularly those non-coding small RNA genes.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. These data can be found here: https://github.com/minhui803/XGEM.

AUTHOR CONTRIBUTIONS

HM collected the data, implemented the algorithm, performed the experiments, analyzed the results, and partially wrote the manuscript; X-HX helped in designing the algorithm and analyzed the results; C-QG analyzed the results and partially wrote the manuscript; LW and P-FD directed the whole study, conceptualized the algorithm, supervised the experiments, analyzed the results, and wrote the manuscript.

FUNDING

This work was supported by the National Natural Science Foundation of China (NSFC 61872268) and the National Key R&D Program of China (2018YFC0910405).

ACKNOWLEDGMENTS

We would like to acknowledge Mr. Yunkai Guo for helpful discussions in this work.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2022.877409/full#supplementary-material

REFERENCES

Ahmed, K., LaPierre, M. P., Gasser, E., Denzler, R., Yang, Y., Rülicke, T., et al. (2017). Loss of microRNA-7a2 Induces Hypogonadotropic Hypogonadism and Infertility. *J. Clin. Invest.* 127, 1061–1074. doi:10.1172/JCI90031

Bartel, D. P. (2018). Metazoan MicroRNAs. *Cell* 173, 20–51. doi:10.1016/j.cell.2018.03.006

Bartel, D. P. (2004). MicroRNAs. *Cell* 116, 281–297. doi:10.1016/S0092-8674(04)00045-5

Campos, T. L., Korhonen, P. K., Gasser, R. B., and Young, N. D. (2019). An Evaluation of Machine Learning Approaches for the Prediction of Essential Genes in Eukaryotes Using Protein Sequence-Derived Features. *Comput. Struct. Biotechnol. J.* 17, 785–796. doi:10.1016/j.csbj.2019.05.008

Cao, J., Liu, G.-S., Zou, N.-Z., Zhang, H., He, X.-X., Sun, P.-L., et al. (2022). microRNA-200c-3p Suppresses Proliferation and Invasion of Nephroblastoma Cells by Targeting EP300 and Inactivating the AKT/FOXO1/p27 Pathway. *neoplasma*. doi:10.4149/neo_2022_210922N1340

Chen, T., and Guestrin, C. (2016).XGBoost, in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco California USA (ACM), 785–794. doi:10.1145/2939672.2939785

Chen, W., Zhang, X., Brooker, J., Lin, H., Zhang, L., and Chou, K.-C. (2015). PseKNC-General: a Cross-Platform Package for Generating Various Modes of Pseudo Nucleotide Compositions. *Bioinformatics* 31, 119–120. doi:10.1093/bioinformatics/btu602

Chen, Z., Zhao, P., Li, F., Leier, A., Marquez-Lago, T. T., Wang, Y., et al. (2018). iFeature: a Python Package and Web Server for Features Extraction and Selection from Protein and Peptide Sequences. *Bioinformatics* 34, 2499–2502. doi:10.1093/bioinformatics/bty140

Fridrichova, I., and Zmetakova, I. (2019). MicroRNAs Contribute to Breast Cancer Invasiveness. *Cells* 8, 1361. doi:10.3390/cells8111361

Grajski, K. A., Breiman, L., Di Prisco, G. V., and Freeman, W. J. (1986). Classification of EEG Spatial Patterns with a Tree-Structured Methodology: CART. *IEEE Trans. Biomed. Eng.* BME-33, 1076–1086. doi:10.1109/TBME.1986.325684

He, L., and Hannon, G. J. (2004). MicroRNAs: Small RNAs with a Big Role in Gene Regulation. *Nat. Rev. Genet.* 5, 522–531. doi:10.1038/nrg1379

Kozomara, A., Birgaoanu, M., and Griffiths-Jones, S. (2019). miRBase: from microRNA Sequences to Function. *Nucleic Acids Res.* 47, D155–D162. doi:10.1093/nar/gky1141

Larrimore, K. E., and Rancati, G. (2019). The Conditional Nature of Gene Essentiality. *Curr. Opin. Genet. Develop.* 58-59, 55–61. doi:10.1016/j.gde.2019.07.015

Lee, Y. S., and Dutta, A. (2009). MicroRNAs in Cancer. *Annu. Rev. Pathol. Mech. Dis.* 4, 199–227. doi:10.1146/annurev.pathol.4.110807.092222

Lee, R. C., Feinbaum, R. L., and Ambros, V. (1993). The *C. elegans* Heterochronic Gene Lin-4 Encodes Small RNAs with Antisense Complementarity to Lin-14. *Cell* 75, 843–854. doi:10.1016/0092-8674(93)90529-Y

Leslie, C. S., Eskin, E., Cohen, A., Weston, J., and Noble, W. S. (2004). Mismatch String Kernels for Discriminative Protein Classification. *Bioinformatics* 20, 467–476. doi:10.1093/bioinformatics/btg431

Liu, B., Fang, L., Liu, F., Wang, X., and Chou, K.-C. (2016a). iMiRNA-PseDPC: microRNA Precursor Identification with a Pseudo Distance-Pair Composition Approach. *J. Biomol. Struct. Dyn.* 34, 223–235. doi:10.1080/07391102.2015.1014422

Liu, B., Gao, X., and Zhang, H. (2019). BioSeq-Analysis2.0: an Updated Platform for Analyzing DNA, RNA and Protein Sequences at Sequence Level and Residue Level Based on Machine Learning Approaches. *Nucleic Acids Res.* 47, e127. doi:10.1093/nar/gkz740

Liu, B., Liu, F., Fang, L., Wang, X., and Chou, K.-C. (2016b). repRNA: a Web Server for Generating Various Feature Vectors of RNA Sequences. *Mol. Genet. Genomics* 291, 473–481. doi:10.1007/s00438-015-1078-7

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. (2002). Text Classification Using String Kernels. *J. Machine Learn. Res.* 2, 419–444. doi:10.1162/153244302760200687

Lorenz, R., Bernhart, S. H., Höner zu Siederdissen, C., Tafer, H., Flamm, C., Stadler, P. F., et al. (2011). ViennaRNA Package 2.0. *Algorithms Mol. Biol.* 6, 26. doi:10.1186/1748-7188-6-26

Martin, E. C., Qureshi, A. T., Dasa, V., Freitas, M. A., Gimble, J. M., and Davis, T. A. (2016). MicroRNA Regulation of Stem Cell Differentiation and Diseases of the Bone and Adipose Tissue: Perspectives on miRNA Biogenesis and Cellular Transcriptome. *Biochimie* 124, 98–111. doi:10.1016/j.biochi.2015.02.012

Nematbakhsh, S., Pei Pei, C., Selamat, J., Nordin, N., Idris, L. H., and Abdull Razis, A. F. (2021). Molecular Regulation of Lipogenesis, Adipogenesis and Fat Deposition in Chicken. *Genes* 12, 414. doi:10.3390/genes12030414

Reinhart, B. J., Slack, F. J., Basson, M., Pasquinelli, A. E., Bettinger, J. C., Rougvie, A. E., et al. (2000). The 21-nucleotide Let-7 RNA Regulates Developmental Timing in *Caenorhabditis elegans*. *Nature* 403, 901–906. doi:10.1038/35002607

Ru, X., Cao, P., Li, L., and Zou, Q. (2019). Selecting Essential MicroRNAs Using a Novel Voting Method. *Mol. Ther. - Nucleic Acids* 18, 16–23. doi:10.1016/j.omtn.2019.07.019

Song, F., Cui, C., Gao, L., and Cui, Q. (2019). miES: Predicting the Essentiality of miRNAs with Machine Learning and Sequence Features. *Bioinformatics* 35, 1053–1054. doi:10.1093/bioinformatics/bty738

Wang, Y., Ru, J., Jiang, Y., and Zhang, J. (2019). Adaboost-SVM-based Probability Algorithm for the Prediction of All Mature miRNA Sites Based on Structured-Sequence Features. *Sci. Rep.* 9, 1521. doi:10.1038/s41598-018-38048-7

Wei, L., Liao, M., Gao, Y., Ji, R., He, Z., and Zou, Q. (2014). Improved and Promising Identification of Human MicroRNAs by Incorporating a High-Quality Negative Set. *Ieee/acm Trans. Comput. Biol. Bioinf.* 11, 192–201. doi:10.1109/TCBB.2013.146

Wojciechowska, A., Osiak, A., and Kozar-Kamińska, K. (2017). MicroRNA in Cardiovascular Biology and Disease. *Adv. Clin. Exp. Med.* 26, 868–874. doi:10.17219/acem/62915

Xue, C., Li, F., He, T., Liu, G.-P., Li, Y., and Zhang, X. (2005). Classification of Real and Pseudo microRNA Precursors Using Local Structure-Sequence Features and Support Vector Machine. *BMC Bioinformatics* 6, 310. doi:10.1186/1471-2105-6-310

Yan, C., Wu, F.-X., Wang, J., and Duan, G. (2020). PESM: Predicting the Essentiality of miRNAs Based on Gradient Boosting Machines and Sequences. *BMC Bioinformatics* 21, 111. doi:10.1186/s12859-020-3426-9

Zeng, P., Chen, J., Meng, Y., Zhou, Y., Yang, J., and Cui, Q. (2018). Defining Essentiality Score of Protein-Coding Genes and Long Noncoding RNAs. *Front. Genet.* 9, 380. doi:10.3389/fgene.2018.00380

Zhang, B., Pan, X., Cobb, G. P., and Anderson, T. A. (2007). MicroRNAs as Oncogenes and Tumor Suppressors. *Dev. Biol.* 302, 1–12. doi:10.1016/j.ydbio.2006.08.028

Zhang, J., Xu, Y., Liu, H., and Pan, Z. (2019). MicroRNAs in Ovarian Follicular Atresia and Granulosa Cell Apoptosis. *Reprod. Biol. Endocrinol.* 17, 9. doi:10.1186/s12958-018-0450-y

Zhang, W.-Y., Xu, J., Wang, J., Zhou, Y.-K., Chen, W., and Du, P.-F. (2021). KNIndex: a Comprehensive Database of Physicochemical Properties for K-Tuple Nucleotides. *Brief Bioinform* 22, bbaa284. doi:10.1093/bib/bbaa284

Zhang, Y., Yun, Z., Gong, L., Qu, H., Duan, X., Jiang, Y., et al. (2018). Comparison of miRNA Evolution and Function in Plants and Animals. *MIRNA* 7, 4–10. doi:10.2174/2211536607666180126163031

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers | Frontiers in Genetics

SYSTEMATIC REVIEW
published: 26 April 2022
doi: 10.3389/fgene.2022.845305

A Comprehensive Review of Artificial Intelligence in Prevention and Treatment of COVID-19 Pandemic

Haishuai Wang[1], *Shangru Jia*[2], *Zhao Li*[3], *Yucong Duan*[4], *Guangyu Tao*[5]* *and Ziping Zhao*[2]*

[1]College of Computer Science, Zhejiang University, Hangzhou, China, [2]Department of Computer and Information Engineering, Tianjin Normal University, Tianjin, China, [3]Alibaba-ZJU Joint Research Institute of Frontier Technologies, Zhejiang University, Hangzhou, China, [4]College of Computer Science and Technology, Hainan University, Haikou, China, [5]Department of Radiology, Shanghai Chest Hospital, Shanghai Jiaotong University, Shanghai, China

The unprecedented outbreak of the Corona Virus Disease 2019 (COVID-19) pandemic has seriously affected numerous countries in the world from various aspects such as education, economy, social security, public health, etc. Most governments have made great efforts to control the spread of COVID-19, e.g., locking down hard-hit cities and advocating masks for the population. However, some countries and regions have relatively poor medical conditions in terms of insufficient medical equipment, hospital capacity overload, personnel shortage, and other problems, resulting in the large-scale spread of the epidemic. With the unique advantages of Artificial Intelligence (AI), it plays an extremely important role in medical imaging, clinical data, drug development, epidemic prediction, and telemedicine. Therefore, AI is a powerful tool that can help humans solve complex problems, especially in the fight against COVID-19. This study aims to analyze past research results and interpret the role of Artificial Intelligence in the prevention and treatment of COVID-19 from five aspects. In this paper, we also discuss the future development directions in different fields and prove the validity of the models through experiments, which will help researchers develop more efficient models to control the spread of COVID-19.

Keywords: Artificial Intelligence, clinical diagnosis, COVID-19, medical imaging, Pandemic Prediction, pandemic, COVID-19 review, telemedicine

OPEN ACCESS

Edited by:
Andrey Ivanov,
Emory University, United States

Reviewed by:
Haizhou Liu,
Wuhan Institute of Virology (CAS), China
Dinesh Gupta,
International Centre for Genetic Engineering and Biotechnology, India

***Correspondence:**
Guangyu Tao
121058820@qq.com
Ziping Zhao
zhaoziping@tjnu.edu.cn

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *29 December 2021*
Accepted: *30 March 2022*
Published: *26 April 2022*

Citation:
Wang H, Jia S, Li Z, Duan Y, Tao G and Zhao Z (2022) A Comprehensive Review of Artificial Intelligence in Prevention and Treatment of COVID-19 Pandemic. Front. Genet. 13:845305. doi: 10.3389/fgene.2022.845305

INTRODUCTION

In December 2019, COVID-19 hit Hubei, China, and many pneumonia cases of unknown cause were found in some hospitals in Wuhan. The pandemic has been infecting millions of people afterwards, which was eventually confirmed as an acute respiratory infection caused by Novel Coronavirus 2019 infection. On 11 February 2020, the World Health Organization (WHO) named it "COVID-19" (Wang et al., 2020a; He et al., 2020; Sohrabi et al., 2020), and the fight against COVID-19 began around the world. This disease is a highly contagious and highly pathogenic infectious disease, which may cause various forms of disease from mild to severe (Chen et al., 2020a; Paules et al., 2020). For example, it can transfer the mild self-limiting respiratory illness to severe pneumonia and even cause multiple organ failure, or death. Up till to 23 September 2021, there have been 230,773,965 COVID-19 infections worldwide, as shown in **Figure 1**, the number of confirmed COVID-19 infections is still increasing. **Figure 2** shows the top 15 countries with the highest cumulative number of confirmed cases and the highest

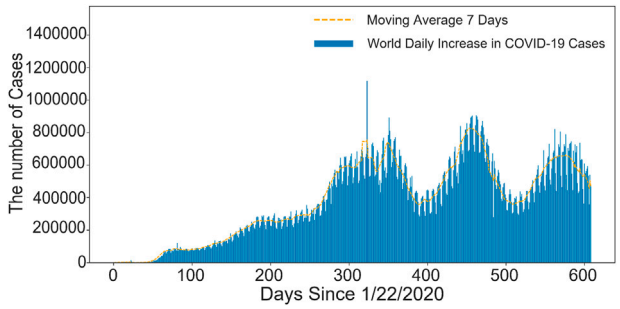


FIGURE 1 | The number of new confirmed cases worldwide every day. The abscissa is the timeline and the ordinate is the number of COVID-19 confirmed cases. The number of COVID-19 cases is increasing in the first 300 days, and there is a wavy line in the second 300 days.

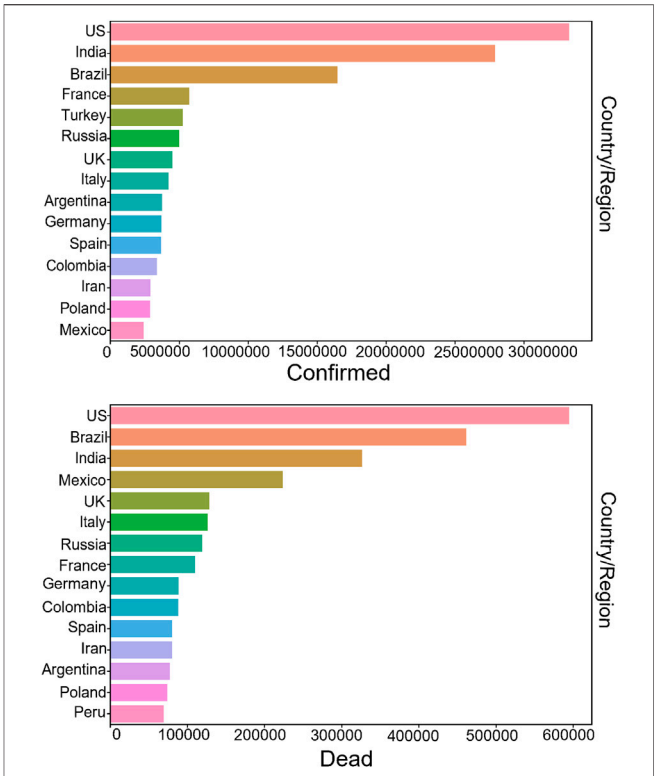


FIGURE 2 | Top 15 countries with cumulative confirmed cases and deaths. Confirmed cases are shown above, and deaths are shown below. The United States, India and Brazil are the top three, with the United States having the most cumulative confirmed cases and deaths.

number of deaths globally, where the top three are United States, Brazil and India. [The data in **Figures 1**, **2** are from the website: https://github.com/CSSEGISandData/COVID-19. This COVID-19 data repository is from the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University. The data was downloaded on 20 September 2021]. Thus, it is worth thinking about what caused the pandemic. The outbreak of the pandemic is due to the lack of relevant information in the early stages and the prediction of its future transmission, resulting in delayed national containment measures and low awareness of self-protection among the population. Moreover, in some areas with poor medical conditions, there is not enough vaccine for the public, and patients cannot afford systematic treatment or expensive hospital expenses, thereby they have to self-isolate which greatly increased the risk of infection.

At present, due to the lack of effective antiviral drugs for COVID-19, patients with mild symptoms can be treated with general treatment, such as bed rest, timely and effective oxygen therapy, appropriate application of antibiotics, antiviral therapy and glucocorticoid therapy, etc. In the treatment of critically ill patients, the treatment principle is based on general treatment, such as actively prevent complications, treat basic diseases, prevent secondary infections, support organ functions, and respiratory support, etc. However, these methods are not able to completely stop the death toll from rising, hence, developing a drug that targets COVID-19 would be an effective way to stop the spread of the pandemic (Bayat et al., 2021).

Recently, more and more AI researchers have devoted to the prevention and treatment of COVID-19 from different fields (Chamola et al., 2020), including clinical medicine, economics, infectious diseases, computer science, psychology, government management, etc. Therefore, Artificial Intelligence is able to help us better understand the protein structure of COVID-19 virus and develop effective drugs to cure patients (Rahman et al., 2020; Soomro et al., 2022), which will greatly save the time of drug design and vaccine development. It can also diagnose whether it is infected by learning clinical data and Computed Tomography (CT) images, which greatly saves the problem of manpower shortage, in order to help to control suspected patients as soon as possible, and implement measures such as isolation and monitoring (Yu et al., 2020). Second, machine learning can also be used to make reasonable predictions about the future development trend of the COVID-19, so as to help decision-makers implement corresponding control measures to prevent the spread of COVID-19. Finally, the construction of telemedicine platform is inseparable from the participation of AI. Therefore, AI plays an extremely important role in combating the COVID-19 pandemic.

Nowadays, researchers have been widely applying AI to against the outbreak of COVID-19. In this paper, we aim to systematically review the active role of AI in prevention the outbreak of COVID-19 pandemic, and the current challenges in the related research. In addition, we also summarized and demonstrated the recently studies in terms of the results and conclusions from different aspects. Chapter 2 discusses the interpretation of medical images by AI. Chapter 3 introduces the use of clinical data modeling to detect the severity of patients. Chapter 4 discusses the application of AI in the

TABLE 1 | Main methods of Medical Imaging for COVID-19.

Classifier	Data set	Accuracy	Data availability	References
CNN	2000 x-rays images (162 COVID-19 positive, 4280 common pneumonia positive, 400 TB positive)	99.92%	https://github.com/ieee8023/covid-chestxray-dataset	Das et al. (2020)
CNN + PCA	500 X-ray images (250 COVID-19 positive cases and 250 normal healthy cases.)	97.6–100%	https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia	Rasheed et al. (2021)
CNN + ACGAN	1124 X-ray images (403 images of COVID-19 and 721 normal images)	95%	https://github.com/agchung/Figure1-COVID-chestxray-dataset	Waheed et al. (2020)
Based on deep convolutional neural network CovXNet	1583 normal X-ray images, (1493 COVID-19 pneumonia X-ray images and 2780 bacterial pneumonia X-ray images)	97.4% (Second category) 90.2% (Multiple categories)	https://github.com/Perceptron21/CovXNet	Mahmud et al. (2020)
Deep CNN transfer learning method	423 COVID-19, 1485 viral pneumonia and 1579 normal chest X-ray images	99.7% (Second category) 97.9% (Three categories)	https://www.kaggle.com/tawsifurrahman/covid19-radiography-database	Chowdhury et al. (2020)
Deep CNN model CoroNet	X-ray images of 1203 normal cases, 1591 viral pneumonia cases	95% (Three categories) 93% (Four categories)	https://github.com/drkhan107/CoroNet	Khan et al. (2020)
COVID-Net	COVID X Open access to the benchmark data set (13,975 CXR images, 358 COVID-19 CXR images.)	98.9%	https://github.com/lindawangg/COVID-Net	Wang et al. (2020b)
nCOVnet	142 COVID-19 X-ray images 5863 non-COVID-19 X-ray images	97%	https://github.com/ieee8023/covid-chestxray-dataset	Panwar et al. (2020)
DenseNet121	2724 C T images (1029 COVID-19 images,)	90.8%	https://wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI	Harmon et al. (2020)
DarkNet model based on Deep Learning	500 normal and 500 COVID-19 images	98.08% (Second category) 87.02% (Multiple categories)	https://github.com/muhammedtalo/COVID-19	Tulin et al. (2020)
Deep transfer learning (DTL) model with DenseNet201	1,262 COVID-19 positive images, 1,230 negative images	99.82%	https://www.kaggle.com/plameneduardo/sarscov2-ctscan-dataset	Vijay et al. (2020)
An automated COVID-19 screening (ACoS)	696 normal, 696 pneumonia and 696 COVID-19 X-ray images	98.062%	https://github.com/ieee8023/covid-chestxray-dataset	Chandra et al. (2021)
Based on deep Bayes-Extrusion Network-COVID Diagnosis-Net	X-ray images (1583 normal persons, 4290 cases of common pneumonia, and 76 cases of COVID-19 infection)	100% (Second category) 98.3% (Three categories)	https://data.mendeley.com/datasets/rscbjbr9sj/2	Ucar and Korkmaz, (2020)
Deep learning model and transfer learning based on VGG-16	250 COVID-19 images, 2753 other lung diseases images, and 3520 health images	98%	https://github.com/muhammedtalo/COVID-19	Brunese et al. (2020)
A weakly supervised deep learning framework	TCIA Open data set (150 3D volumetric chest CT exams of COVID-19, CAP and NP patients)	92.3%	https://www.cancerimagingarchive.net/collections/	Hu et al. (2020)
A technique based on a deep residual network	1345 viral pneumonia cases, 10,200 normal cases and 3616 COVID-19 cases	92.1% (Four categories)	https://github.com/pawelparker/DNN-lung-infection-Pattern	Panahi et al. (2022)
Transfer learning 29 different types of AI-based models	352 chest X-ray images (51 COVID-19, 21 non-COVID-19,160 pneumonia,54 TB, and 66 normal images)	93.8% (Validation accuracy)	https://github.com/arunsharma8osdd/covidpred	Sharma et al. (2020)
A multi-view feature learning method	1092 X-ray images (364 COVID-19, 364 normal, and 364 pneumonia)	99.82% (Three categories)	https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia	Hamidreza, (2022)

treatment of patients with COVID-19. Chapter 5 summarizes the COVID-19 epidemic prediction model represented by mathematical models and machine learning models. Chapter 6 introduces the current development of telemedicine technology. Finally, the challenges and future development directions of AI technology in the prevention and treatment of COVID-19 are discussed in Chapter 7.

ARTIFICIAL INTELLIGENCE INTERPRETATION OF CHEST RADIOLOGY IMAGES

Recently, with the development of computer technology, AI interpretation of medical images can help doctors identify and detect the types of diseases and determine the affected areas. As

COVID-19 is persistently ravaging the world, researchers have been leveraging medical images (e.g., chest X-rays and CT images) as the main tools for COVID-19 diagnosis. This section summarizes the main methods of medical imaging for COVID-19 in **Table 1**. Methods based on deep learning, such as deep feature extraction, pre-trained Convolutional Neural Network (CNN) and end to end training CNN models, have been widely used for image classification tasks. For depth feature extraction, most of the preprocessed depth CNN models are used, such as Residual Neural Network 18 (ResNet18), Residual Neural Network 50 (ResNet50), Residual Neural Network 101 (ResNet101), Visual Geometry Group 16 (VGG16) and Visual Geometry Group 19 (VGG19). For the classification of deep features, a Support Vector Machine (SVM) classifier is used together with various functions, e.g., Linear, Quadratic, Cubic and Gaussian, etc.

Das et al. (Das et al., 2020) proposed a CNN-based model to identify infected cases from viral pneumonia or healthy cases. This work used 6 datasets which contain 7,000 X-ray images, in order to classify the COVID-19 positive, positive ordinary pneumonia, *tuberculosis* positive and healthy patients. The classification accuracy (AUC) of the model for COVID-19 positive and negative cases was 99.96% (AUC was 1.0). Similarly, it has an accuracy of 99.92% (AUC 0.99) in classifying pneumonia, *tuberculosis* and COVID-19 positive cases. Rasheed et al. (Rasheed et al., 2021) added a dimension reduction method based on Principal Component Analysis (PCA) on the basis of the CNN to further accelerate the learning process and improve classification accuracy by selecting features with high discriminability. The results showed that the overall accuracy was 95.2%–97.6% without PCA and 97.6–100% with PCA for positive case identification. The applicability of PCA dimension reduction is illustrated. In addition, Waheed et al. (Waheed et al., 2020) proposed a method for synthesizing chest X-ray (CXR) images by developing models based on auxiliary classifier Generative Adversarial Networks (GAN), which improved accuracy 10% by adding synthetic images generated by Covid-GAN. Recently, transfer learning has been widely used in this field. Chowdhury et al. (Chowdhury et al., 2020) proposed a robust technique for automatic detection of COVID-19 pneumonia from chest X-ray images by leveraging pre-trained deep learning algorithms to maximize detection accuracy, which achieved 99.7% accuracy.

So which model is more effective at detecting COVID-19? Elasnaoui et al. (El Asnaoui and Chawki, 2021) introduced a deep learning model (VGG16, VGG19, Densenet201, Inception_ResNet_V2, Inception_V3, Resnet50, And MobileNet_V2) conducted a comparative study on the detection and classification of COVID-19. Results showed that the use of ResnetV2 and Densnet201 had better results than other models used in this study (accuracy of ResnetV2 and Densnet201 was 92.18 and 88.09%, respectively). Ismael et al. (Ismael and Şengür, 2021) proposed a new end-to-end training CNN model. The Support Vector Machines (SVM) classifier was used to classify the deep features, and different functions were matched. The results show that the deep features extracted from the ResNet50 model and the SVM classifier with linear kernel function produce an accuracy of 94.7%, which is the highest among all the obtained results. In addition, it also shows that deep learning methods are better than local descriptors. Especially the performance of deep features and SVM classifier is better than other methods. In deep feature classification, the cubic function is usually better than all other functions. The ResNet50 model usually produces better results than other preprocessed CNN models. Finally, for end-to-end training, deep CNN models produce better results than shallow networks. Therefore, we also tried to use Resnet to classify chest X-ray images into three categories: normal, viral pneumonia and COVID-19. The accuracy rate in the validation set is 96%. **Figure 3** shows the good performance of the model, which can accurately classify X-ray images. The specific structure of this model is shown in **Figure 4**.

While AI has made some progress in medical imaging, with many models achieving near 100% accuracy on open data sets, there is still a long way to go. We believe that the following points need to be paid attention to in the future: 1. A large open data set is very necessary. So we must continue to increase data sharing and jointly build a complete large-scale database for researchers to use. 2. Hospital imaging data may be incomplete, so we need to improve the accuracy of segmentation and classification to prevent diagnostic errors for COVID-19. 3. As the current epidemic is normalized, we need to develop a system to reduce the pressure on doctors and better apply it to clinical practice. 4. Marking data manually is expensive and time-consuming, so unsupervised deep learning models will be the focus of future research. Finally, we hope that medical image recognition can be deployed to hospitals as soon as possible, so that more patients can receive immediate treatment and save more lives.

ARTIFICIAL INTELLIGENCE ANALYSIS OF PATIENT CLINICAL DATA

Since 2019, the COVID-19 has gripped the world. The COVID-19 is shockingly transmissible and is constantly mutating. even in an era dominated by information technology, clinical information data on COVID-19 patients is still scarce, and clinical predictions of morbidity, mortality, severity and prognosis are lagging behind. This requires the sharing of Electronic Health Records (EHR) clinical data with researchers and public health agencies. Brat et al. (Brat et al., 2020) formed an International Consortium (4CE) consisting of 96 hospitals in five countries. Successfully leveraged the open source Informatics for Integrating Biology & the Bedside (I2B2) tool KIT10-17 to manage, complete, and share data extracted from the EHR. The goal is to integrate, share and interpret data about the clinical trajectory of patients. Of course, we also hope that more websites around the world can share data with hospitals, which will make a great contribution to clinical intelligence in the future.

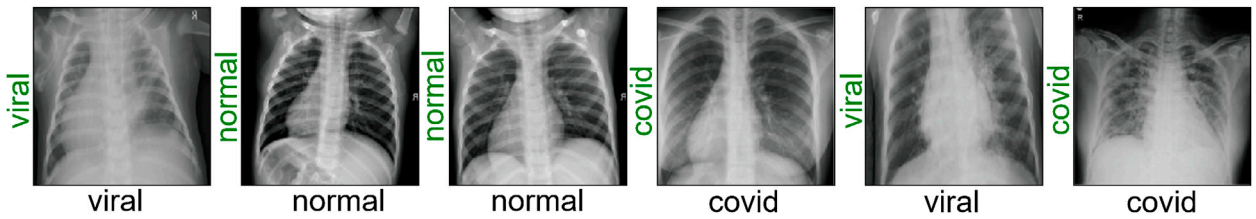


FIGURE 3 | Classification results of Resnet model. If the classification is correct, a green label will appear, otherwise a red label will appear. The Resnet model can correctly classify normal, viral pneumonia, and COVID-19 after training.

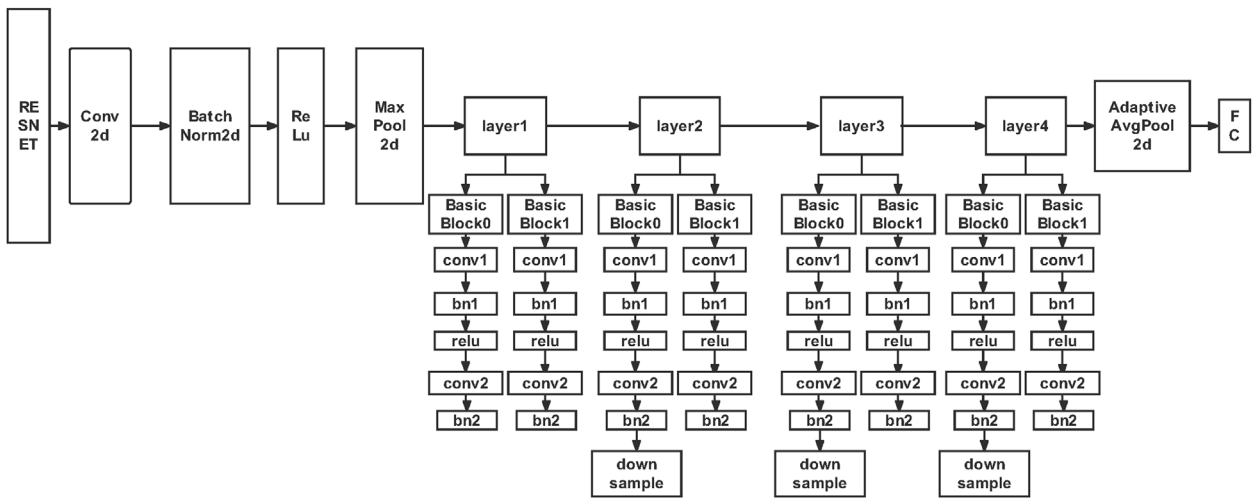


FIGURE 4 | Resnet model framework structure. The characteristic of Resnet is that it is easy to optimize and can improve accuracy by increasing depth. The internal residual block uses jump connection to alleviate the problem of gradient disappearance.

As shown in **Table 2**, currently many researchers have begun to make use of limited clinical data to predict the severity of COVID-19 patients and conduct feature screening for critical indicators in clinical data in combination with Artificial Intelligence methods. Razavian et al. (Razavian et al., 2020) used clinical data and EHR data of 3345 retrospective and 474 prospective hospitalized cases, and based on real-time data values, vital signs and oxygen support variables, established and verified a black box model to identify patients with good prognosis within 96 h. The results showed that the Light Gradient Boosting Machine (Light GBM) model performed well in EHR data, with a positive predictive value of 93.3%. In addition, Arjun et al. (Yadaw et al., 2020) applied machine learning technology to 3841 patients treated by Mount Sinai Health System in New York City, The United States, implemented a systematic machine-learning-based framework by using missing value interpolation, 6 feature selection, 7 classification and 4 statistical techniques. It was found that three highly accessible clinical parameters of patient age, minimum oxygen saturation, and type of patient encounter were fed into an automated Extreme Gradient Boosting (XGBoost) algorithm that accurately classified patients as likely to survive or die. In addition, Liang et al. introduced a machine learning variable selection algorithm called Least Absolute Shrinkage and Selection Operator (LASSO), it was used to identify 10 variables with statistical significance ($p < 0.05$) hazard ratio characteristics (Liang et al., 2020a), (Liang et al., 2020b), a COVID-Gram-based online calculator was developed to allow clinicians to enter the values of the 10 variables required for the risk score and automatically calculate the likelihood of a COVID-19 inpatient developing critical illness (95%CI). Covino et al. used Multivariate proportional hazards (COX) regression to determine the risk factors related to progression (Marcello et al., 2020), (Ji et al., 2020), and a new predictive scoring model was established. Liang also compared the deep learning survival COX model with the classical COX model (Liang

TABLE 2 | Modeling method of EHR data.

Model	Data set	Result	Important features	Availability	References
Three models, clinical feature model (C model), radiological semantic feature model (R model), and clinical and radiological semantic feature combination model (CR model)	CT images and clinical data from 70 COVID-19 and 66 non-COVID-19 pneumonia patients	The CR model has the highest accuracy and specificity with a maximum AUC of 0.98	GGO with consolidation, tree-in-bud, offending vessel augmentation in lesions, temperature, heart ratio, etc.	https://doi.org/10.1007/s00330-020-06829-2	Chen et al. (2020b)
Four models (Logistic Regression, Random Forest, Light-GBM, and a collection of these three models)	Clinical data and EHR data of 3345 retrospective and 474 prospective Inpatients	The Light-GBM model achieved the best performance on the validation set	Age, Sex, Race, Neutrophils Percent, Lymphocytes Percent, Eosinophils Percent, C-Reactive Protein, C-Reactive Protein, etc.	https://doi.org/10.1038/s41746-020-00343-x	Razavian et al. (2020)
Recursive Feature Elimination method, Logistic Regression, Support Vector Machine, Random Forest and Extreme Gradient Enhancement (XGBoost) algorithm for prediction.	In 3841 patients at Mount Sinai Health System, 961 retrospective and 249 prospective patients	XGBoost algorithm can accurately classify patients as likely to live or die.	Age, minimum oxygen saturation, and type of patient encounter, etc.	https://github.com/SBCNY/Clinical-predictors-of-COVID-19-mortality	Yadaw et al. (2020)
$\chi 2$ test or Kruskal-Wallis test, Multivariate Regression analysis	1,951 charts of confirmed cases in 26 hospitals in Italy.	mortality is predicted by age and the presence of comorbidities.	Age, diabetes, chronic obstructive pulmonary disease (COPD) and chronic kidney disease, etc.	https://www.clinicaltrials.gov	Wang et al. (2020c)
Mann-whitney U, $\chi 2$ test, Univariate Cox Analysis	Clinical data from 69 patients	The risk of death in elderly patients may be independent of age, and the presence of severe dementia is a risk factor for this population.	Lactate dehydrogenase and blood oxygen saturation, etc.	https://doi.org/10.1111/ggi.13960	Marcello et al. (2020)
Minimum absolute contraction selection operator (LASSO) and Logistic Regression	COVID-19 patients from 575 hospitals in 31 provincial-level regions in China	The predictive variables were extracted and the severity of the patients was calculated successfully	Age, Dyspnea, Cancer history, COPD, Comorbidity, X-ray abnormality, etc.	https://github.com/cojocchen/covid19_critically_ill	Liang et al. (2020b)
Multivariate COX Regression	Clinical data of 208 patients	The CALL scoring model was established, and the area under ROC curve was 0.91	Age, Comorbidity, Lymphocyte, D-dimer, LDH, Lymphocyte, etc.	https://doi.org/ 10.1093/cid/ciaa414	Ji et al. (2020)
Machine learning variable selection algorithm for Minimum Absolute Contraction and Selection Operator (LASSO), Combined with Cox deep learning model	1590 patients at 575 medical centers	Deep learning survival Cox model is better than traditional Cox model	Age, hemoptysis, dyspnea, unconsciousness, number of comorbidities, cancer history, neutrophil-to-lymphocyte ratio, etc.	http://118.126.104.170/	Liang et al. (2020a)
Models Based on Whole Clinical Parameters	A publicly available dataset consisting of clinical parameters and protein profile data	The best classification model based on clinical parameters achieved a maximum accuracy of 89.47%	Serum creatinine, age, absolute lymphocyte count, and D-dimer and proteins.	http://14.139.62.220/covidprognosis/	Sardar et al. (2021)
Unsupervised hierarchical clustering and principal component analysis.	Patients. Rotterdam cohort samples	An immune-type based scheme to stratify COVID-19 patients at hospital admittance into high and low risk clinical categories	Serum pro-inflammatory, anti-inflammatory and anti-viral cytokine and anti-SARS-CoV-2 antibody measurements	https://bitbucket.org/immunology-emc/covid_severity_publication/src/master/	Mueller et al. (2022)

et al., 2020b), and found that the deep learning survival COX model is better.

Finally, multimodal clinical data information can more accurately diagnose and predict the risk level of patients. Chen et al. (Chen et al., 2020b) combined modeling of medical images and clinical data, and found that the combined model of clinical and radiological semantic features achieved the best effect, with the highest accuracy and specificity, and the maximum AUC was 0.986. Liang (Liang et al., 2020b) added the abnormality of X-ray image into clinical information, and found that the abnormality of X-ray image was the first predictive variable of critical condition. So we hope that future EHR data will be more readily available, a more authoritative and comprehensive database will be established. In this way, our research will be in-depth, and the proposed model is applicable.

At last, we use a variety of EHR data modeling, and it is found that The Neural Network Classifier (NN), Random Forest (RF), K-Nearest Neighbor (KNN), SVM, Naive Bayes (NB), Logistic Regression (LR) and Linear Discriminant Analysis (LDA) have a good effect with an accuracy of 95.4422%. The comparison of

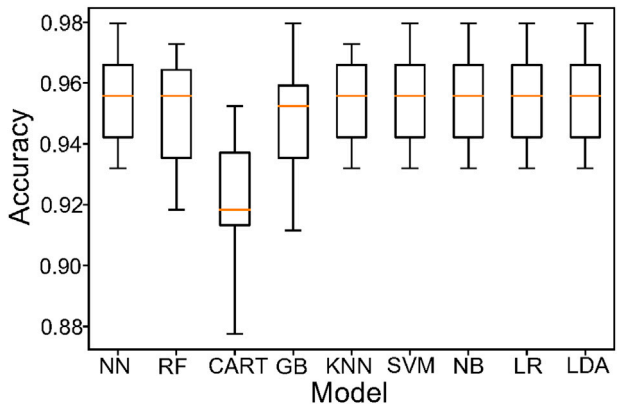


FIGURE 5 | Model comparison of EHR data. Using a box plot to reflect the classification characteristics of different models. The NN,RF,SVM, NB, LR and LDA are better than other models.

different algorithms is shown in **Figure 5**. We compare the advantages and disadvantages of different algorithms by using a box plot, which is composed of five numerical points, namely, minimum observed value, 25% quantile, median, 75% quantile and maximum observed value. We can conclude from the figure that the average value (yellow line) of NN, RF, KNN, SVM, NB, LR, and LDA is 95.4422%, which is better than Gaussian Bayes (GB) Classification and Regression Trees (CART). Although, we can use the above model to analyze the clinical data of patients, so as to obtain the corresponding prediction results (whether they have COVID-19 or not). Since machine learning model and deep learning model are black box models, we need to study their interpretability more, so that we can understand the mechanism of model prediction and its practicability more easily. In addition, we can combine clinical data with medical imaging data to build a comprehensive model. The problems are analyzed from the multi-modal perspective by integrating various disciplines.

ARTIFICIAL INTELLIGENCE DISCOVERY OF DISEASE TREATMENTS

At present, COVID-19 has spread all over the world. The cunning virus is constantly mutating and posing a serious threat to human health in the world. It means the use of Artificial Intelligence to identify the host protein and the possible targeting mechanism of the COVID-19 protein has important implications for prevention and treatment of COVID-19. Das et al. (Das et al., 2021) proposed a computational scheme for reconstructing the host virus protein-protein interaction network, using host proteins from 17 important signaling pathways to investigate possible targeting mechanisms of Severe Acute respiratory Syndrome Coronavirus 2 (SARS-CoV-2) proteins. The results showed that Non-structural Proteins (NSP3) and Structural Protein (Spike) were the most influential proteins in interacting with multiple host proteins. The Mitogen-activated Protein Kinase (MAPK) pathway is the most severely affected pathway in SARS-COV-2 infection. Some proteins involved in multiple pathways are highly concentrated in host Protein-Protein Interactions (PPI) and are mainly targeted by multiple viral proteins. The most prominent drug molecules highlighted in the study are arsenic trioxide, dexamethasone and hydroxychloroquine, which may play an important role in preventing deaths. Yaar et al. (Yaar et al., 2021) used Deep Learning (DL), Random Forest (RF), and Gradient Boosted Trees (GBTs) were used to predict the relationship between disease severity and protein in 93 samples (60 COVID-19 patients, 33 controls) and 370 variables from open websites. The study identified TGB1BP2 in cardiovascular group II and MILR1 in inflammatory group as the two most important proteins associated with disease severity. Compared with other algorithms, the proposed model (GBTs) achieves the best prediction of disease severity based on protein. The results also suggest that changes in blood protein associated with the severity of COVID-19 can be used for disease surveillance, early diagnosis and treatment.

In addition, Artificial Intelligence can also be used to discover effective drugs to treat COVID-19. Kong et al. (Kong et al., 2020) described a Web server that can predict binding patterns between COVID-19 targets and ligands, including small molecules, peptides and antibodies. The server provides a friendly interface and binding pattern visualization for the results, which makes it a useful tool for discovering COVID-19 drugs. Wang et al. (Wang, 2020) effectively provided possible treatment options for the outbreak of COVID-19 infectious diseases through computer-aided drug design. This study found that some drugs can act as inhibitors of the major proteases in novel coronavirus, including Carfilzomib, lopinavir et al. Contribute to rational drug design for COVID-19 major proteases. Beck et al. (Beck et al., 2020) used a pretrained Deep-Learning-based drug targeting interaction model, namely molecular converter-drug targeting interaction (MT-DTI), to identify commercially available drugs that act on SARS-COV-2 virus proteins. An antiretroviral drug used for the treatment and prevention of Human Immunodeficiency Virus (HIV) was found to be the best compound, with an inhibitory effect of 94.94 nm against SARS-CoV-2 3C-like proteases. Ton et al. (Ton et al., 2020) introduced a new Deep Learning platform, Deep Docking (DD). The DD combined with Glide can be used to quickly estimate the docking fraction between 1.3 billion chemical structures and the new SARS-CoV-2MPro active site, so that drugs with higher docking fraction can be found compared with known protease inhibitors. Beata et al. (Beata et al., 2020) used Cryo-electron tomography and molecular dynamics simulation were used to help us understand SARS-CoV-2 infection and develop safe vaccines.

At last. Senior et al. (Senior et al., 2020) trained a Neural Network to accurately predict the distance between residue pairs, which conveyed more information about the structure than contact prediction, and determined the most likely three-dimensional shape of the protein through energy

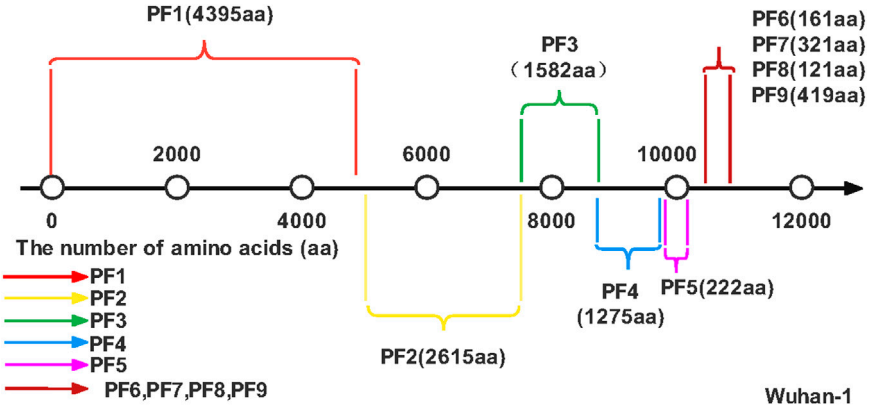


FIGURE 6 | SARS-COV-2 genome. The number line represents the number of amino acids, and different colors represent different Protein Fragments.

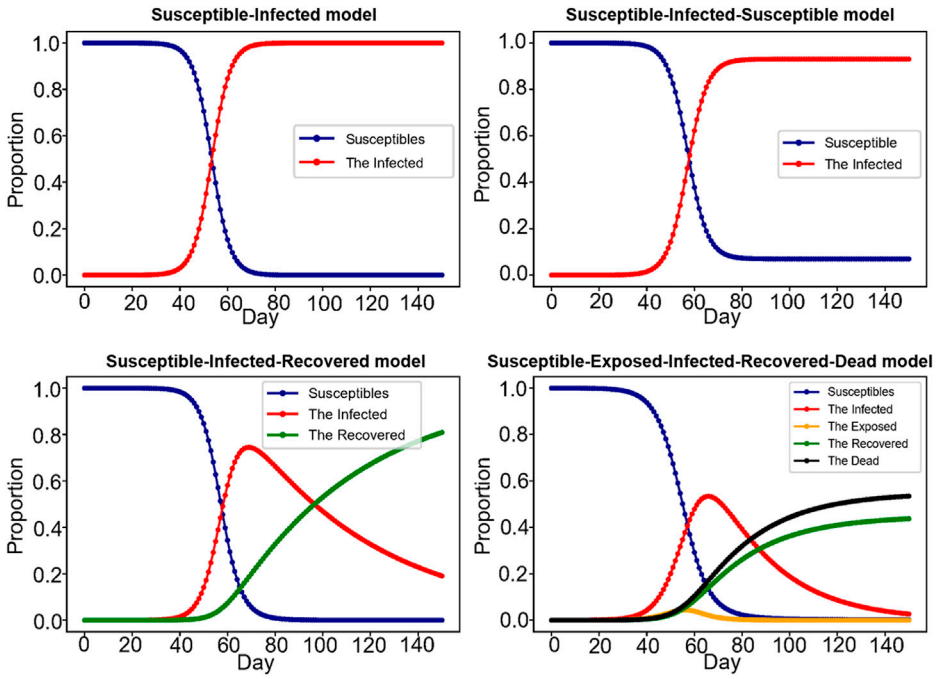


FIGURE 7 | Mathematical model. The picture shows four classic models of infectious diseases, Susceptible-Infected model (SI), Susceptible-Infected- Susceptible model (SIS), Susceptible-Infected-Recovered model (SIR), Susceptible-Exposed-Infected-Recovered-Dead model (SEIRD), with each letter representing a state. For example, SEIRD model represents susceptible, infected, exposed, recovered, and dead.

minimization. This adds to our understanding of the COVID-19 and helps us develop effective treatments for patients with COVID-19. Schaarschmidt et al. (Schaarschmidt et al., 2017) analyzed protein prediction using Coevolution and Machine Learning methods, compared it with previous CASP experiments, and discussed the results of structure prediction and prediction provided on finite target sets. They found that in more than half of the targets, especially those with many homologous sequences, the accuracy was more than 90%, and in some cases the best predictors were 100% accurate. In conclusion, AI can help us get out of the COVID-19 sooner or later!

Figure 6 shows the genome organization of SARS-COV-2. The organization of a genome is the linear sequence of genetic

material (DNA/RNA) and its division into specific functional segments. We can use Artificial Intelligence to extract some Protein Fragment. And compared with SARS-COV genome tissue. In experiments, Although the SARS-COV genome is very similar to that of SARS-COV-2, we know that the DNA/RNA of the two viruses are very different by measuring the editing distance. Finally, The Protein Fragments (PF) such as PF1, PF2, etc. were extracted and the amino acid (aa) numbers of bases on different fragments was obtained. As shown in **Figure 6**, different colors represent different PF, each PF contains a different number of amino acids, for example, the red line is the first Protein Fragment (PF1), which consists of 4,395 amino acids. Using Artificial Intelligence to study the COVID-19 genome will help enhance our understanding of the virus's genes and speed up the development of specific drugs and vaccines against COVID-19. In addition, we can also carry out different experiments to screen different drugs, the effect of clinical treatment, and get the best treatment drugs and methods.

ARTIFICIAL INTELLIGENCE PREDICTIONS OF COVID-19 PANDEMIC

As we all know, the rapid spread of the COVID-19 has brought public health departments in some countries to the brink of collapse, with shortages of basic medical equipment such as Intensive Care Unit (ICU) beds, ventilators, masks and protective suits. Therefore, a reasonable AI prediction model plays an important role in predicting the future development trend of COVID-19, formulating scientific and reasonable prevention and control measures, consolidating the existing epidemic prevention achievements, maintaining the safety of life and property of the public and stabilizing the social development order (Foppa et al., 2017).

At present, the most important COVID-19 prediction models at home and abroad mainly focus on traditional mathematical models, such as Susceptible-Infected model (SI), Susceptible-Infected- Susceptible model (SIS), Susceptible-Infected-Recovered model (SIR), Susceptible-Exposed-Infected-Recovered model (SEIR), Susceptible-Infected-Recovered-Dead model (SIRD), Susceptible-Exposed-Infected-Recovered-Dead model (SEIRD), etc., and popular machine learning models (such as Linear Regression model, Polynomial Regression model, Support Vector Machine model, Artificial Neural Network model, etc.). The traditional mathematical model refers to the mathematical analysis of the transmission mode, transmission speed and transmission range of infectious disease on the basis of population number, and expresses it in the form of differential equations. Treating infectious diseases from a mathematical perspective can reveal the internal model and potential structure of epidemic control, and contribute to an in-depth understanding of the transmission dynamics of infectious diseases and the potential effects of different public health intervention strategies (Rahimi et al., 2021). As the World Health Organization puts it, real-time mathematical models play a key role in responding to outbreaks. **Supplementary Appendix SA** shows some basic mathematical models, their specific differential equations and parameter meanings. **Figure 7** shows some basic mathematical models.

Apparently, these are basic mathematical models, and because COVID-19 is subject to so many uncertainties, cultural, economic, political and sociological are critical to understand the epidemic. Therefore, only by taking into account various factors as much as possible, such as seasonal influence, in-and-out rate of national population, infection rate of latent population, ICU beds in hospital, efficiency of receiving and treating, etc. this model can be closer to the actual situation and simulate the real effect. **Supplementary Appendix SB** shows the current epidemic prediction models and their respective strengths and weaknesses. For example, Wang et al. (Jia et al., 2020) introduced an extended SIR model, which combined real-time isolation measures and expanded the SIR model to adapt to the real-time changing transmission rate in the population, and covered the effects of different epidemic prevention measures. Ivorra et al. (Ivorra et al., 2020) proposed a new θ-SEIHRD model (not SIR, SEIR, or other general models) to simulate the propagation of infectious diseases. The model takes into account known characteristics of COVID-19, such as the presence of undetected infectious cases and the different infectious characteristics of hospitalized patients. The method also takes into account the fraction θ of detected cases relative to the total number of actual infections, the need for hospital beds can be estimated, and so on. The SEIQR model proposed by Mandal et al. (Mandal et al., 2020). They introduced isolation levels and government interventions, such as lockdown, media coverage of social distancing, and improved public health, to reduce disease transmission. Since many people had little information about the COVID-19 virus in the early stages of the epidemic, Zhao et al. (Zhao et al., 2021) considered that information could influence human behavior, thus influencing the dynamic transmission process of the epidemic layer. Therefore, the proposed SEIR/ V-UA model incorporates an information mechanism to better fit the future development trend of COVID-19.

In addition to mathematical models, the Machine Learning model shown in **Supplementary Appendix SC** has also become an important tool for researchers (Chimmula and Zhang, 2020; Rustam et al., 2020). One application of Neural Network is for time series prediction algorithm. Neural Network can learn the behavior of time-related data, and can predict the future value. Oliveira et al. (de Oliveira et al., 2021) proposed an Artificial Neural Network model, in which an ANN model was applied to predict the number of confirmed COVID-19 cases and deaths, as well as the time series for the next 7 days in Brazil, Portugal and the United States. Mohimont et al. (Mohimont and Chemchem, 2020) mainly studied a number of models based on CNN, and also proposed a layered transfer learning scheme. Finally, good national and regional accuracy is obtained, and the performance of ordinary CNN is improved. It is now integrated into a COVID-19 surveillance and prediction instrument. Leslie (Leslie and Yeager, 2020) developed a predictive model for the outbreak of COVID-19 in Canada using deep learning (DL) models. The model uses recursive Long Short-Term Memory (LSTM) networks to adapt to the nonlinearity of a given COVID-19 data set, which can overcome the limitations of traditional time

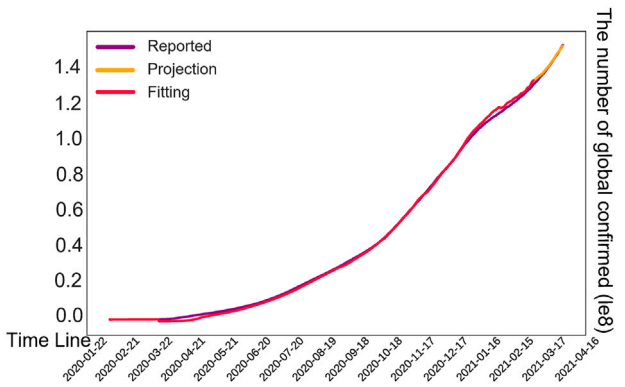


FIGURE 8 | Prediction results of our model. The vertical axis represents the number of global confirmed cases. The red lines are fitted trends, the purple lines are actual cases, and the orange lines are predicted trends over the next month.

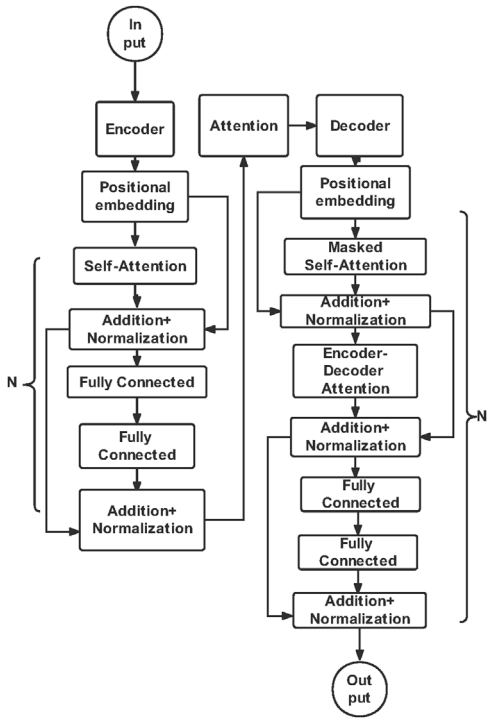


FIGURE 9 | Transformer model structure diagram. Transformer models have Input, Output, Attention mechanisms and Encoder-Decoder architecture. In our T-SIRGAN prediction module, the encoder models the relationships among orders in the sequence, and the decoder learns the variable representation vector.

series prediction techniques and produce the latest results on time data. Bhimala et al. (Bhimala and Patra, 2021) assessed the relationship between weather factors and COVID-19 cases, and established a predictive model using deep learning model LSTM. The results show that the multivariate LSTM model based on temperature time series data performs well in the high humidity regions of Kerala, Tamil Nadu and West Bengal. It shows that certain high humidity areas are more conducive to the outbreak of COVID-19.

The next, some classical time series prediction models are also favored by researchers. Time series models Autoregressive Integrated Moving Average (ARIMA) and Seasonal Autoregressive Integrated Moving Average (SARIMA) were used to predict COVID-19 pandemic trends in the top 16 countries with 70–80% of the global cumulative cases (ArunKumar et al., 2021). The results showed that SARIMA's predictions were more realistic than ARIMA's, confirming the existence of seasonality in COVID-19 data. ARIMA, Brownian exponential smoothing and RNN-LSTM were compared (Guleryuz, 2021). It is found that the ARIMA model can fit the new outbreak situation well. Molin et al. (Molin et al., 2020) compared and analyzed many models, and found that in all scenarios, the models ranked from the best to the worst in accuracy were Support Vector Regression (SVR), Stacking Ensemble Learning, ARIMA, CUBIST, RIDGE, and RF models. Of course, these models have their advantages and disadvantages. Mathematical model with the combination of machine learning model could become a hotspot of research on the future, Zhong team (Yang et al., 2020) considered each province between the flow of population, and use the modified SEIR model and LSTM model, proves the rationality of China's strict control measures, according to the analysis of China's the outbreak at its peak in late February 2020, By the end of April 2020, it showed a gradual decline. A 5-day delay would triple the size of the outbreak in mainland China. Lifting quarantine in Hubei will result in a second epidemic peak in Hubei province in mid-March 2020 and extend the epidemic until late April, as confirmed by Machine Learning predictions. The dynamic SEIR model can effectively predict the peak and magnitude of the COVID-19 epidemic. The control measures implemented on 23 January 2020 are essential to reduce the eventual scale of the COVID-19 epidemic.

Finally, we proposed a T-SIRGAN model to predict the future trend of the epidemic (Wang et al., 2022). Due to the lack of data volume, we used Generative Adversarial Networks (GAN) to amplify the data, and replaced the random noise of GAN with the noise regulated by SIR model. Then, Transformers are used to predict the future trends of COVID-19 based on the generated synthetic data. We found that this model performs well compared to LSTM, ARIMA, Decision Tree Regression, SVM, K Neighbors Regression and other models. In addition, the development trend of COVID-19 next month was successfully fitted with an error of 0.0035 MSE, as shown in **Figure 8** and **Figure 9** shows the model structure of the Transformer model for predictive tasks.

To sum up, the fusion of classical mathematical model of infectious diseases and deep learning model will be a research direction in the future, and the advantages of both can be combined to make a more accurate prediction of the future trend of COVID-19. We also need to study the transmission

mechanism of COVID-19 from all angles, including season, temperature, demographic, social, economic, medical, educational and political. Make timely predictions and implement the best control measures to stop the spread of COVID-19.

ARTIFICIAL INTELLIGENCE CONSTRUCTION OF TELEMEDICINE PLATFORM

There are many ways of detecting COVID-19. In addition to nucleic acid tests, clinical manifestations, CT images, etc. In recent years, with the development of 5G technology, telemedicine has gained great development space. We can check our health status with some smart devices. This allows us to see what's going on in our bodies without leaving the house, and if something goes wrong, we can treat it immediately and prevent it from getting worse. Wosik et al. (Wosik et al., 2020) introduced the role of telemedicine in three stages of American medical service: 1) Home clinics 2) Mitigated the proliferation of pandemic hospitals 3) Post pandemic recovery. The COVID-19 pandemic is forcing all health systems, hospitals and clinics to quickly implement telemedicine services, and telemedicine's time has come. Due to the large gap between urban and rural medical conditions, Hirko et al. (Hirko et al., 2020) pointed out that the rapid implementation of telemedicine plan in rural areas in response to the COVID-19 pandemic would solve the gap in rural medical conditions to a great extent. In order to build a telemedicine platform, it is necessary to obtain user information in the data system of mobile phone suppliers. Leslie et al. (Leslie and Yeager, 2020) proposed to promote the openness of data so as to promote the construction of telemedicine platform. A full spectrum of researchers will need to be mobilized to understand and respond to the challenges posed by the epidemic.

Apparently, not only the germ of theory, but also Rao et al. (Srinivasa Rao and Vazquez, 2020) proposed a Machine Learning algorithm to collect travel history and common symptoms through online surveys based on smartphones. The data collected can be used to assist in the initial screening and early identification of possible COVID-19 infections. Thousands of data points can be collected and processed through an Artificial Intelligence (AI) framework that can ultimately assess individuals at risk of contracting the virus and categorize them into no risk, lowest risk, medium risk, and high risk. Cases identified in high-risk groups can be isolated earlier, reducing the chance of transmission. Turer et al. (Turer et al., 2020) recommended to use Electronic Personal Protective Equipment (EPPE) to protect employees and preserve Personal Protective Equipment (PPE) during the COVID-19 pandemic, as well as to provide rapid emergency care to low risk patients. Tucker et al. (Tucker, 2020) provided a remote patient monitoring solution for COVID-19 patients (Get Well Loop). Minimizing the exposure rate of COVID-19 patients. Remote patient monitoring is an effective way to manage COVID-19 patients at home.

Telemedicine platforms should provide users with the latest epidemic trends, remind them to take appropriate prevention and control measures, and help users check whether they have had close contact with confirmed cases. If the user has physical discomfort, can immediately call the police or emergency call, so as to get the corresponding isolation and treatment. In the future, the popularization of telemedicine can not only alleviate the shortage of hospital resources during the epidemic, but also monitor the activities of the incubation period population in real time, facilitating screening and controlling the spread of the epidemic. Of course, the premise is to get users' permission, and protect the security of users' information, to prevent the use of illegal elements (Islam et al., 2020).

DISCUSSIONS AND FUTURE RESEARCH DIRECTIONS

Above all, Artificial Intelligence technology plays an extremely important role in the prevention and control of COVID-19, especially in the field of clinical medicine, it can quickly identify the patient's CT and X-ray images to diagnose the type of pneumonia patients. To learn the clinical data of patients, find out the clinical features of COVID-19 patients, and predict the current severity level, so as to send a warning message to the medical staff. However, this study argues that there are still some challenges regarding the application of Artificial Intelligence algorithms in the field of medicine (Mohamadou et al., 2020).

First of all, the main challenge of COVID-19 detection is the problem of data imbalance. Due to the scarcity of lung image data of COVID-19 patients, the development model cannot be evaluated and tested on a large number of data sets, and the best Artificial Intelligence algorithm cannot be selected. This requires us to establish an open and shared data set for researchers to train and test models (Islam and Islam, 2020). Secondly, there is still a lack of available label data, and extending existing data sets or using a small number of samples in model training are the current strategies that must be chosen. However, most current models are weakly supervised methods, because manual tagging of imaging data is time-consuming and expensive. In the future, we may need unsupervised deep learning models and transfer learning methods to process imaging data. It can not only ensure the accuracy of the algorithm, but also break the limitation of labeled data. Moreover, the diagnosis of medical imaging using artificial intelligence requires sufficient evidence to prove its correctness, because artificial intelligence is regarded as a black box. Thus, the interpretability of artificial intelligence models is of importance in the field of medicine. Finally, medical images cannot fully reflect whether COVID-19 is really infected or not, and a model needs to be established from a multimodal perspective. Artificial Intelligence can learn from Multi-modal clinical data to introduce more intelligence to the medical systems to capture the characteristics of disease, so as to obtain reliable results for COVID-19 diagnosis. To develop a more efficient and versatile system to achieve

better clinical medical purposes (Alamo and Daniel, 2020), (Rahman and Sarker, 2021).

In addition, Artificial Intelligence also plays an extremely important role in the discovery of drugs, vaccines, choice of treatment and the medical staff of risk assessment. In the future work, we will not only go toward the direction of more intelligent and precise, but also we need to explore other applications of Artificial Intelligence and modeling for COVID-19 in healthcare (Rahman and Sarker, 2021), (Ricci et al., 2021), (Bhargava and Bansal, 2021). Only in this way, COVID-19 pandemic can be conquered as soon as possible. Finally, we can also use Artificial Intelligence technology to make reasonable predictions of the future development trend of COVID-19, so as to formulate the corresponding prevention and control measures. The British statistician George E.P.Box once said, "All models are wrong, but some models are useful." The prediction models of the COVID-19 epidemic are also the same, where simple models of the early stages of the growing epidemic can still serve as reference information and provide the basis for more complex transmission models. However, it is not possible to say which model fully matches the spread of COVID-19, so the prediction model is only a way to provide us with early warning, and we should treat it with caution (Ulhaq et al., 2020).

Regarding the development direction of infectious disease dynamics models, this study believes that the combination of mathematical model and machine model learning is the main trend of future development, which can not only improve the adaptability of the model, but also increase the scientific rationality of simulation prediction. Secondly, it is necessary to fully grasp the main factors affecting the development trend of the epidemic, including population migration, seasonal factors, isolation control measures, etc. (Ahmad et al., 2020), and they should be integrated into the model, so as to better fit the spread of the epidemic in reality. Finally, we will explore Multi-modal integration, and future research should incorporate data from other sources, such as social media, mass media.

CONCLUSION

One and a half years have passed since the COVID-19 outbreak. During this period of time, vaccines and new treatments have been come out one after another. However, the COVID-19 virus is very cunning. It is constantly mutating in different countries and regions based on local natural geographical environment, population immunity and other factors, seriously threatening human life and health. With the development of Artificial Intelligence technology, more and more researchers are committed to fighting COVID-19 virus through Artificial Intelligence. This paper reviews five aspects of Artificial Intelligence's the interpretation of medical images, modeling of patient clinical data, finding effective drugs to cure patients, predicting the future development trend of epidemic, and building a remote intelligent medical platform. It also introduces the Artificial Intelligence algorithm used in five major aspects, the data sets used, and evaluates the limitations and advantages of the model. Although the current model has made some achievements, there are still great challenges for the future, especially the openness of data sets and the generalization ability of models. Multimodal models will be one of the main research models in the future. In the end, models just provide some advice and information. The most important thing is to rely on our concerted efforts to protect ourselves, cooperate with the government's epidemic prevention policies, and actively vaccinate. Only in this way can we defeat COVID-19 at an early date.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding authors.

AUTHOR CONTRIBUTIONS

HW and SJ wrote the manuscript with support from GT and ZZ. ZL and YD provided some figures and tables, and helped supervise the project. GT and ZZ supervised the project, contributed to the final version of the manuscript, and proofread throughout the manuscript.

FUNDING

Hainan Provence Research Funding No. ZDYF2020023.

ACKNOWLEDGMENTS

This work was partly supported by Zhejiang Provincial Key Laboratory of Service Robot, College of Computer Science, Zhejiang University.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2022.845305/full#supplementary-material

REFERENCES

Ahmad, A., Garhwal, S., Ray, S. K., Kumar, G., Malebary, S. J., and Barukab, O. M. (2020). The Number of Confirmed Cases of Covid-19 by Using Machine Learning: Methods and Challenges. *Arch. Computat Methods Eng.* 28, 2645–2653. doi:10.1007/s11831-020-09472-8

Alamo, T., and Daniel, G. (2020). Covid-19: Open-Data Resources for Monitoring, Modeling, and Forecasting the Epidemic. *Electronics* 9, 827. doi:10.3390/electronics9050827

ArunKumar, K. E., Kalaga, D. V., Sai Kumar, C. M., Chilkoor, G., Kawaji, M., and Brenza, T. M. (2021). Forecasting the Dynamics of Cumulative COVID-19 Cases (Confirmed, Recovered and Deaths) for Top-16 Countries Using Statistical Machine Learning Models: Auto-Regressive Integrated Moving

Average (ARIMA) and Seasonal Auto-Regressive Integrated Moving Average (SARIMA). *Appl. Soft Comput.* 103, 107161–107187. doi:10.1016/j.asoc.2021.107161

Bayat, M., Asemani, Y., Mohammadi, M. R., Sanaei, M., Namvarpour, M., and Eftekhari, R. (2021). An Overview of Some Potential Immunotherapeutic Options against COVID-19. *Int. Immunopharmacology* 95, 107516. doi:10.1016/j.intimp.2021.107516

Beata, T., Sikora, M., and Christoph, S. (2020). *In Situ* structural Analysis of SARS-CoV-2 Spike Reveals Flexibility Mediated by Three Hinges. *Science* 270, eabd5223. doi:10.1126/science.abd5223

Beck, B. R., Shin, B., Choi, Y., Park, S., and Kang, K. (2020). Predicting Commercially Available Antiviral Drugs that May Act on the Novel Coronavirus (SARS-CoV-2) through a Drug-Target Interaction Deep Learning Model. *Comput. Struct. Biotechnol. J.* 18, 784–790. doi:10.1016/j.csbj.2020.03.025

Bhargava, A., and Bansal, A. (2021). Novel Coronavirus (COVID-19) Diagnosis Using Computer Vision and Artificial Intelligence Techniques: a Review. *Multimed Tools Appl.* 80, 19931–19946. doi:10.1007/s11042-021-10714-5

Bhimala, K. R., and Patra, G. K. (2021). Prediction of COVID-19 Cases Using the Weather Integrated Deep Learning Approach for India. *Transboundary Emerging Dis.*, 1–15. doi:10.1111/tbed.14102

Brat, G. A., Weber, G. M., Gehlenborg, N., Avillach, P., Palmer, N. P., Chiovato, L., et al. (2020). International Electronic Health Record-Derived COVID-19 Clinical Course Profiles: the 4CE Consortium. *Npj Digit. Med.* 3, 109–118. doi:10.1038/s41746-020-00308-0

Brunese, L., Mercaldo, F., Reginelli, A., and Santone, A. (2020). Explainable Deep Learning for Pulmonary Disease and Coronavirus COVID-19 Detection from X-Rays. *Comput. Methods Programs Biomed.* 196, 105608–105619. doi:10.1016/j.cmpb.2020.105608

Chamola, V., Hassija, V., Gupta, V., and Guizani, M. (2020). A Comprehensive Review of the COVID-19 Pandemic and the Role of IoT, Drones, AI, Blockchain, and 5G in Managing its Impact. *IEEE Access* 8, 90225–90265. doi:10.1109/access.2020.2992341

Chandra, T. B., Verma, K., Singh, B. K., Jain, D., and Netam, S. S. (2021). Coronavirus Disease (COVID-19) Detection in Chest X-Ray Images Using Majority Voting Based Classifier Ensemble. *Expert Syst. Appl.* 165, 113909–113922. doi:10.1016/j.eswa.2020.113909

Chen, X., Tang, Y., Mo, Y., Li, S., Lin, D., Yang, Z., et al. (2020). A Diagnostic Model for Coronavirus Disease 2019 (COVID-19) Based on Radiological Semantic and Clinical Features: a Multi-center Study. *Eur. Radiol.* 30, 4893–4902. doi:10.1007/s00330-020-06829-2

Chen, Y., Liu, Q., and Guo, D. (2020). Emerging Coronaviruses: Genome Structure, Replication, and Pathogenesis. *J. Med. Virol.* 92, 418–423. doi:10.1002/jmv.25681

Chimmula, V. K. R., and Zhang, L. (2020). Time Series Forecasting of COVID-19 Transmission in Canada Using LSTM Networks. *Chaos, Solitons & Fractals* 135, 109864–109870. doi:10.1016/j.chaos.2020.109864

Chowdhury, M. E. H., Rahman, T., Khandakar, A., Mazhar, R., Kadir, M. A., Mahbub, Z. B., et al. (2020). Can AI Help in Screening Viral and COVID-19 Pneumonia? *IEEE Access* 8, 132665–132676. doi:10.1109/access.2020.3010287

Das, D., Santosh, K. C., and Pal, U. (2020). Truncated Inception Net: COVID-19 Outbreak Screening Using Chest X-Rays. *Phys. Eng. Sci. Med.* 43, 915–925. doi:10.1007/s13246-020-00888-x

Das, J. K., Chakraborty, S., and Roy, S. (2021). A Scheme for Inferring Viral-Host Associations Based on Codon Usage Patterns Identifies the Most Affected Signaling Pathways during COVID-19. *J. Biomed. Inform.* 118, 103801–103814. doi:10.1016/j.jbi.2021.103801

de Oliveira, L. S., Gruetzmacher, S. B., and Teixeira, J. P. (2021). COVID-19 Time Series Prediction. *Proced. Comput. Sci.* 181, 973–980. doi:10.1016/j.procs.2021.01.254

El Asnaoui, K., and Chawki, Y. (2021). Using X-ray Images and Deep Learning for Automated Detection of Coronavirus Disease. *J. Biomol. Struct. Dyn.* 39, 3615–3626. doi:10.1080/07391102.2020.1767212

Foppa, I. M., Kermack, W. O., and McKendrick, A. G. (2017). "W.O. Kermack and A.G. McKendrick: A Seminal Contribution to the Mathematical Theory of Epidemics (1927)," in *A Historical Introduction to Mathematical Modeling of Infectious Diseases*. Editor I. M. Foppa (Cambridge, Massachusetts, United States: Academic Press), 59–87. doi:10.1016/b978-0-12-802260-3.00004-3

Guleryuz, D. (2021). Forecasting Outbreak of COVID-19 in Turkey; Comparison of Box-Jenkins, Brown's Exponential Smoothing and Long Short-Term Memory Models. *Process Saf. Environ. Prot.* 149, 927–935. doi:10.1016/j.psep.2021.03.032

Hamidreza, H. (2022). Deep Multi-View Feature Learning for Detecting COVID-19 Based on Chest X-ray Images. *Biomed. Signal Process. Control.* 75, 103595. doi:10.1016/j.bspc.2022.103595

Harmon, S. A., Sanford, T. H., Xu, S., Turkbey, E. B., Roth, H., Xu, Z., et al. (2020). Artificial Intelligence for the Detection of COVID-19 Pneumonia on Chest CT Using Multinational Datasets. *Nat. Commun.* 11, 4080–4088. doi:10.1038/s41467-020-17971-2

He, F., Deng, Y., and Li, W. (2020). Coronavirus Disease 2019: What We Know? *J. Med. Virol.* 92, 719–725. doi:10.1002/jmv.25766

Hirko, K. A., Kerver, J. M., Ford, S., Szafranski, C., Beckett, J., Kitchen, C., et al. (2020). Telehealth in Response to the Covid-19 Pandemic: Implications for Rural Health Disparities. *J. Am. Med. Inform. Assoc.* 27, 1816–1818. doi:10.1093/jamia/ocaa156

Hu, S., Gao, Y., Niu, Z., Jiang, Y., Li, L., Xiao, X., et al. (2020). Weakly Supervised Deep Learning for COVID-19 Infection Detection and Classification from CT Images. *IEEE Access* 8, 118869–118883. doi:10.1109/access.2020.3005510

Islam, M. N., and Islam, A. K. M. N. (2020). A Systematic Review of the Digital Interventions for Fighting COVID-19: The Bangladesh Perspective. *IEEE Access* 8, 114078–114087. doi:10.1109/access.2020.3002445

Islam, M. N., Islam, I., Munim, K. M., and Islam, A. K. M. N. (2020). A Review on the mobile Applications Developed for COVID-19: An Exploratory Analysis. *IEEE Access* 8, 145601–145610. doi:10.1109/access.2020.3015102

Ismael, A. M., and Şengür, A. (2021). Deep Learning Approaches for COVID-19 Detection Based on Chest X-ray Images. *Expert Syst. Appl.* 164, 114054–115065. doi:10.1016/j.eswa.2020.114054

Ivorra, B., Ferrández, M. R., Vela-Pérez, M., and Ramos, A. M. (2020). Mathematical Modeling of the Spread of the Coronavirus Disease 2019 (COVID-19) Taking into Account the Undetected Infections. The Case of China. *Commun. Nonlinear Sci. Numer. Simulation* 88, 105303–105355. doi:10.1016/j.cnsns.2020.105303

Jahanshahi, H., Munoz-Pacheco, J. M., Bekiros, S., and Alotaibi, N. D. (2021). A Fractional-Order SIRD Model with Time-dependent Memory Indexes for Encompassing the Multi-Fractional Characteristics of the COVID-19. *Chaos, Solitons & Fractals* 143, 110632–110643. doi:10.1016/j.chaos.2020.110632

Ji, D., Zhang, D., Xu, J., Chen, Z., Yang, T., Zhao, P., et al. (2020). Prediction for Progression Risk in Patients with COVID-19 Pneumonia: The CALL Score. *Clin. Infect. Dis.* 71, 1393–1399. doi:10.1093/cid/ciaa414

Jia, W., Han, K., Song, Y., Cao, W., and He, Y. (2020). Extended SIR Prediction of the Epidemics Trend of COVID-19 in Italy and Compared with Hunan, China. *Front. Med.* 7, 169. doi:10.3389/fmed.2020.00169

Khan, A. I., Shah, J. L., and Bhat, M. M. (2020). CoroNet: A Deep Neural Network for Detection and Diagnosis of COVID-19 from Chest X-ray Images. *Comput. Methods Programs Biomed.* 196, 105581–105590. doi:10.1016/j.cmpb.2020.105581

Kong, R., Yang, G., Xue, R., Liu, M., Wang, F., Hu, J., et al. (2020). COVID-19 Docking Server: a Meta Server for Docking Small Molecules, Peptides and Antibodies against Potential Targets of COVID-19. *Bioinformatics* 36, 5109–5111. doi:10.1093/bioinformatics/btaa645

Leslie, L., and Yeager, M. B. (2020). Balancing Health Privacy, Health Information Exchange and Re-Search in the Context of the COVID-19 Pandemic. *J. Am. Med. Inform. Assoc.* 27, 963–966. doi:10.1093/jamia/ocaa039

Liang, W., Liang, H., Ou, L., Chen, B., Chen, A., Li, C., et al. (2020). Development and Validation of a Clinical Risk Score to Predict the Occurrence of Critical Illness in Hospitalized Patients with COVID-19. *JAMA Intern. Med.* 180, 1081–1089. doi:10.1001/jamainternmed.2020.2033

Liang, W., Yao, J., Chen, A., Lv, Q., Zanin, M., Liu, J., et al. (2020). Early Triage of Critically Ill COVID-19 Patients Using Deep Learning. *Nat. Commun.* 11, 3543–3550. doi:10.1038/s41467-020-17280-8

Mahmud, T., Rahman, M. A., and Fattah, S. A. (2020). CovXNet: A Multi-Dilation Convolutional Neural Network for Automatic COVID-19 and Other Pneumonia Detection from Chest X-ray Images with Transferable Multi-

Receptive Feature Optimization. *Comput. Biol. Med.* 122, 103869–103879. doi:10.1016/j.compbiomed.2020.103869

Mandal, M., Jana, S., Nandi, S. K., Khatua, A., Adak, S., and Kar, T. K. (2020). A Model Based Study on the Dynamics of COVID-19: Prediction and Control. *Chaos, Solitons & Fractals* 136, 109889–109901. doi:10.1016/j.chaos.2020.109889

Marcello, C., Giuseppe, D. M., Michele, S., Luca, S., Simeoni, B., and Candelli, M. (2020). Clinical Characteristics and Prognostic Factors in COVID-19 Patients Aged ≥80 Years. *Geriatr. Gerontol. Int.* 20, 704–708. doi:10.1111/ggi.13960

Mohamadou, Y., Halidou, A., and Kapen, P. T. (2020). A Review of Mathematical Modeling, Artificial Intelligence and Datasets Used in the Study, Prediction and Management of COVID-19. *Appl. Intell.* 50, 3913–3925. doi:10.1007/s10489-020-01770-9

Mohimont, L., and Chemchem, A. (2020). Convolutional Neural Networks and Temporal CNNs for COVID-19 Forecasting in France. *IEEE Access* 8, 101489–101499. doi:10.1007/s10489-021-02359-6

Molin, M. H. D., Ramon, G. D. S., Marianicd, V. C., and Coelho, L. D. S. (2020). Short-term Forecasting COVID-19 Cumulative Confirmed Cases: Perspectives for Brazil- ScienceDirect. *Chaos, Solitons & Fractals* 135, 109853–109863. doi:10.1016/j.chaos.2020.109853

Mueller, Y. M., Schrama, T. J., Ruijten, R., Schreurs, M. W. J., Grashof, D. G. B., van de Werken, H. J. G., et al. (2022). Stratification of Hospitalized COVID-19 Patients into Clinical Severity Progression Groups by Immuno-Phenotyping and Machine Learning. *Nat. Commun.* 13, 915. doi:10.1038/s41467-022-28621-0

Nisar, K. S., Ahmad, S., Ullah, A., Shah, K., Alrabaiah, H., and Arfan, M. (2021). Mathematical Analysis of SIRD Model of COVID-19 with Caputo Fractional Derivative Based on Real Data. *Results Phys.* 21, 103772–103781. doi:10.1016/j.rinp.2020.103772

Pacheco, C. C., and Lacerda, C. R. D. (2020). Function Estimation and Regularization in the SIRD Model Ap-Plied to the COVID-19 Pandemics. *Inverse Probl. Sci. Eng.* 29, 1613–1628. doi:10.1080/17415977.2021.1872563

Padhi, A., Pradhan, S., Sahoo, P. P., Suresh, K., Behera, B. K., and Panigrahi, P. K. (2020). Studying the Effect of Lockdown Using Epidemiological Modelling of COVID-19 and a Quantum Computational Approach Using the Ising Spin Interaction. *Sci. Rep.* 10, 21741. doi:10.1038/s41598-020-78652-0

Panahi, A., Askari Moghadam, R., Akrami, M., and Madani, K. (2022). Deep Residual Neural Network for COVID-19 Detection from Chest X-ray Images. *SN COMPUT. SCI.* 3, 169. doi:10.1007/s42979-022-01067-3

Panwar, H., Gupta, P. K., Siddiqui, M. K., Morales-Menendez, R., and Singh, V. (2020). Application of Deep Learning for Fast Detection of COVID-19 in X-Rays Using nCOVnet. *Chaos, Solitons & Fractals* 138, 109944–109952. doi:10.1016/j.chaos.2020.109944

Paules, C. I., Marston, H. D., and Fauci, A. S. (2020). Coronavirus Infections-More Than Just the Common Cold. *Jama* 323, 707–708. doi:10.1001/jama.2020.0757

Rahimi, I., Chen, F., and Gandomi, A. H. (2021). A Review on COVID-19 Forecasting Models. Neural Computing and Applications. *Neural Comput. Applic.* doi:10.1007/s00521-020-05626-8

Rahman, M. M., Paul, K. C., and Hossain, M. A. (2020). Machine Learning on the COVID-19 Pandemic, Human Mobility and Air Quality: A Review. *IEEE Access* 9, 72420–72450. doi:10.1109/ACCESS.2021.3079121

Rahman, S., and Sarker, S. (2021). Deep Learning–Driven Automated Detection of COVID-19 from Radiography Images: a Comparative Analysis. *Cogn. Comput.*, 1–30. doi:10.1007/s12559-020-09779-5

Rasheed, J., Hameed, A. A., Djeddi, C., Jamil, A., and Al-Turjman, F. (2021). A Machine Learning-Based Framework for Diagnosis of COVID-19 from Chest X-ray Images. *Interdiscip. Sci. Comput. Life Sci.* 13, 103–117. doi:10.1007/s12539-020-00403-6

Razavian, N., Major, V. J., Sudarshan, M., Burk-Rafel, J., Stella, P., Randhawa, H., et al. (2020). A Validated, Real-Time Prediction Model for Favorable Outcomes in Hospitalized COVID-19 Patients. *Npj Digit. Med.* 3, 130–143. doi:10.1038/s41746-020-00343-x

Ricci, L., Maesa, D. D. F., Favenza, A., and Ferro, E. (2021). Blockchains for COVID-19 Contact Tracing and Vaccine Support: A Systematic Review. *IEEE Access* 9, 37936–37950. doi:10.1109/access.2021.3063152

Rustam, F., Reshi, A. A., Mehmood, A., Ullah, S., On, B.-W., Aslam, W., et al. (2020). COVID-19 Future Forecasting Using Supervised Machine Learning Models. *IEEE Access* 8, 101489–101499. doi:10.1109/access.2020.2997311

Sardar, R., Sharma, A., and Gupta, D. (2021). Machine Learning Assisted Prediction of Prognostic Biomarkers Associated with COVID-19, Using Clinical and Proteomics Data. *Front. Genet.* 12, 636441. doi:10.3389/fgene.2021.636441

Schaarschmidt, J., Monastyrskyy, B., Kryshtafovych, A., and Bonvin, A. M. J. J. (2017). Assessment of Contact Predictions in CASP12: Co-evolution and Deep Learning Coming of Age. *Proteins* 86, 51–66. doi:10.1002/prot.25407

Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., et al. (2020). Improved Protein Structure Prediction Using Potentials from Deep Learning. *Nature* 577, 706–710. doi:10.1038/s41586-019-1923-7

Sharma, A., Rani, S., Gupta, D., and Clough, A. (2020). Artificial Intelligence-Based Classification of Chest X-Ray Images into COVID-19 and Other Infectious Diseases. *Int. J. Biomed. Imaging* 2020, 1–10. doi:10.1155/2020/8889023

Sohrabi, C., Alsafi, Z., O'Neill, N., Khan, M., Kerwan, A., Al-Jabir, A., et al. (2020). World Health Organization Declares Global Emergency: A Review of the 2019 Novel Coronavirus (COVID-19). *Int. J. Surg.* 76, 71–76. doi:10.1016/j.ijsu.2020.02.034

Soomro, T. A., Zheng, L., Afifi, A. J., Ali, A., Yin, M., and Gao, J. (2022). Artificial Intelligence (AI) for Medical Imaging to Combat Coronavirus Disease (COVID-19): a Detailed Review with Direction for Future Research. *Artif. Intell. Rev.* 55, 1409–1439. doi:10.1007/s10462-021-09985-z

Srinivasa Rao, A. S. R., and Vazquez, J. A. (2020). Identification of COVID-19 Can Be Quicker through Artificial Intelligence Framework Using a mobile Phone-Based Survey when Cities and Towns Are under Quarantine. *Infect. Control. Hosp. Epidemiol.* 41, 826–830. doi:10.1017/ice.2020.61

Stern, R. H. (2020). Locally Informed Simulation to Predict Hospital Capacity Needs during the COVID-19 Pandemic. *Ann. Intern. Med.* 173, 679–680. doi:10.7326/l20-1061

Ton, A. T., Gentile, F., Hsing, M., Ban, F., and Cherkasov, A. (2020). Rapid Identification of Potential Inhibitors of SARS-CoV-2 Main Protease by Deep Docking of 1.3 Billion Compounds. *Mol. Inform.* 39, e2000028. doi:10.1002/minf.202000028

Tucker, A. (2020). Rapid Implementation of a COVID-19 Remote Patient Monitoring Program. *J. Am. Med. Inform. Assoc.* 27, 1326–1330. doi:10.1093/jamia/ocaa097

Tulin, O., Muhammed, T., Eylul, A. Y., Ulas, B. B., and Ozal, Y. (2020). Automated Detection of COVID-19 Cases Using Deep Neural Networks with X-ray Images. *Comput. Biol. Med.* 121, 103792–103803.

Turer, R. W., Jones, I., Rosenbloom, S. T., Slovis, C., and Ward, M. J. (2020). Electronic Personal Protective Equipment: A Strategy to Protect Emergency Department Providers in the Age of COVID-19. *J. Am. Med. Inform. Assoc.* 27, 967–971. doi:10.1093/jamia/ocaa048

Ucar, F., and Korkmaz, D. (2020). COVIDiagnosis-Net: Deep Bayes-SqueezeNet Based Diagnosis of the Coronavirus Disease 2019 (COVID-19) from X-ray Images. *Med. Hypotheses* 140, 109761–109773. doi:10.1016/j.mehy.2020.109761

Ulhaq, A., Born, J., Khan, A., Gomes, D. P. S., Chakraborty, S., and Paul, M. (2020). COVID-19 Control by Computer Vision Approaches: A Survey. *IEEE Access* 8, 179437–179456. doi:10.1109/access.2020.3027685

Vijay, K., Aayush, J., Neha, G., Dilbag, S., and Manjit, K. (2020). Classification of the COVID-19 Infected Patients Using Dense-Net201 Based Deep Transfer Learning. *J. Biomol. Struct. Dyn.* 39, 5682–5689. doi:10.1080/07391102.2020.1788642

Waheed, A., Goyal, M., Gupta, D., Khanna, A., Al-Turjman, F., and Pinheiro, P. R. (2020). CovidGAN: Data Augmentation Using Auxiliary Classifier GAN for Improved Covid-19 Detection. *IEEE Access* 8, 91916–91923. doi:10.1109/access.2020.2994762

Wang, H., Tao, G., Ma, J., Jia, S., Chi, L., Yang, H., et al. (2022). Predicting the Epidemics Trend of COVID-19 Using Epidemiological-Based Generative Adversarial Networks. *IEEE J. Sel. Top. Signal. Process.* doi:10.1109/JSTSP.2022.3152375

Wang, J. (2020). Fast Identification of Possible Drug Treatment of Coronavirus Disease-19 (COVID-19) through Computational Drug Repurposing Study. *J. Chem. Inf. Model.* 60 (6), 3277–3286. doi:10.1021/acs.jcim.0c00179

Wang, L., Lin, Z. Q., and Wong, A. (2020). COVID-net: a Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-ray Images. *Sci. Rep.* 10, 19549–19561. doi:10.1038/s41598-020-76550-z

Wang, L. S., Wang, Y. R., Ye, D. W., and Liu, Q. Q. (2020). Age and Multimorbidity Predict Death Among COVID-19 Patients: Results of the SARS-RAS Study of the Italian Society of Hypertension. *Hypertension* 76, 366–372. doi:10.1161/HYPERTENSIONAHA.120.15324

Wang, L., Wang, Y., Ye, D., and Liu, Q. (2020). Review of the 2019 Novel Coronavirus (SARS-CoV-2) Based on Current Evidence. *Int. J. Antimicrob. Agents* 55, 105948–105977. doi:10.1016/j.ijantimicag.2020.105948

Wosik, J., Fudim, M., Cameron, B., Gellad, Z. F., Cho, A., Phinney, D., et al. (2020). Telehealth Transformation: COVID-19 and the Rise of Virtual Care. *J. Am. Med. Inform. Assoc.* 27, 957–962. doi:10.1093/jamia/ocaa067

Yaar, E., Cola, K. C., and Yologlu, S. (2021). Artificial Intelligence-Based Prediction of Covid-19 Severity on the Results of Protein Profiling. *Comput. Methods Programs Biomed.* 202, 105996–106007. doi:10.1016/j.cmpb.2021.105996

Yadaw, A. S., Li, Y. C., Bose, S., Iyengar, R., Bunyavanich, S., and Pandey, G. (2020). Clinical Features of COVID-19 Mortality: Development and Validation of a Clinical Prediction Model. *The Lancet Digital Health* 2, e516–e525. doi:10.1016/S2589-7500(20)30217-X

Yang, Z., Zeng, Z., Wang, K., Wong, S.-S., Liang, W., Zanin, M., et al. (2020). Modified SEIR and AI Prediction of the Epidemics Trend of COVID-19 in China under Public Health Interventions. *J. Thorac. Dis.* 12, 165–174. doi:10.21037/jtd.2020.02.64

Yarsky, P. (2021). Using a Genetic Algorithm to Fit Parameters of a COVID-19 SEIR Model for US States. *Math. Comput. Simul* 185, 687–695. doi:10.1016/j.matcom.2021.01.022

Yu, P., Xia, Z., Fei, J., and Kumar Jha, S. (2020). An Application Review of Artificial Intelligence in Prevention and Cure of COVID-19 Pandemic. *Comput. Mater. Continua* 65, 743–760. doi:10.32604/cmc.2020.011391

Zhao, X., Zhou, Q., Wang, A., Zhu, F., Meng, Z., and Zuo, C. (2021). The Impact of Awareness Diffusion on the Spread of COVID-19 Based on a Two-Layer SEIR/V–UA Epidemic Model. *J. Med. Virol.* 93, 4342–4350. doi:10.1002/jmv.26945

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers | Frontiers in Genetics

ORIGINAL RESEARCH
published: 03 June 2022
doi: 10.3389/fgene.2022.891265

Representation Learning: Recommendation With Knowledge Graph *via* Triple-Autoencoder

Yishuai Geng[1], *Xiao Xiao*[2]*, *Xiaobing Sun*[1]* and *Yi Zhu*[1]

[1]School of Information Engineering, Yangzhou University, Yangzhou, China, [2]Department of Ultrasound, Affiliated Hospital of Yangzhou University, Yangzhou, China

The last decades have witnessed a vast amount of interest and research in feature representation learning from multiple disciplines, such as biology and bioinformatics. Among all the real-world application scenarios, feature extraction from knowledge graph (KG) for personalized recommendation has achieved substantial performance for addressing the problem of information overload. However, the rating matrix of recommendations is usually sparse, which may result in significant performance degradation. The crucial problem is how to extract and extend features from additional side information. To address these issues, we propose a novel feature representation learning method for the recommendation in this paper that extends item features with knowledge graph via triple-autoencoder. More specifically, the comment information between users and items is first encoded as sentiment classification. These features are then applied as the input to the autoencoder for generating the auxiliary information of items. Second, the item-based rating, the side information, and the generated comment representations are incorporated into the semi-autoencoder for reconstructed output. The low-dimensional representations of this extended information are learned with the semi-autoencoder. Finally, the reconstructed output generated by the semi-autoencoder is input into a third autoencoder. A serial connection between the semi-autoencoder and the autoencoder is designed here to learn more abstract and higher-level feature representations for personalized recommendation. Extensive experiments conducted on several real-world datasets validate the effectiveness of the proposed method compared to several state-of-the-art models.

Keywords: personalized recommendation, autoencoder, semi-autoencoder, representation learning, collaborative filtering

OPEN ACCESS

Edited by:
Yucong Duan,
Hainan University, China

Reviewed by:
Peng Zhou,
Anhui University, China
Junwei Lv,
Hefei University of Technology, China
Lei Li,
Hefei University of Technology, China

***Correspondence:**
Xiao Xiao
092298@yzu.edu.cn
Xiaobing Sun
xbsun@yzu.edu.cn

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *07 March 2022*
Accepted: *11 April 2022*
Published: *03 June 2022*

Citation:
Geng Y, Xiao X, Sun X and Zhu Y (2022) Representation Learning: Recommendation With Knowledge Graph via Triple-Autoencoder. Front. Genet. 13:891265. doi: 10.3389/fgene.2022.891265

1 INTRODUCTION

The success of machine learning algorithms and artificial intelligence methods heavily depends on the feature representation learning of original data (Bengio et al., 2013; Zhuang et al., 2017a). In recent decades, feature representation learning has attracted a vast amount of attention and research from multiple disciplines, such as biomedicine and bioinformatics (Wei et al., 2019; Li et al., 2021), computer vision (Kim et al., 2017), knowledge engineering (Liu et al., 2016), and personalized recommendation (Zhuang et al., 2017b; Zhu et al., 2021). In real-world applications, feature representation learning is considered to obtain the different explanatory factors of variation behind the data (Locatello et al., 2019).

For nearly three decades, effective computational methods have accelerated drug discovery and played an important role in biomedicine, such as predicting molecular properties and identifying interactions between drugs/compounds and their target proteins. In early years, quantum mechanics (Hohenberg and Kohn, 1964), such as density functional theory (DFT), was used to determine the molecular structure and calculate properties of interest for a molecule. However, the quantum computational method usually consumes tremendous computational resources and takes hours to days to calculate the molecular properties (Ramakrishnan et al., 2015), which hinders their applications to the fields of high-throughput screening. Nowadays, the powerful ability to learn representation and efficiently recommend algorithms has received significant attention. A key challenge is to learn useful molecular representation information from the huge molecular dataset.

Among all the informatics-related application scenarios, with the rapid development of the Internet, there is an urgent demand for personalized recommendation to tackle the information overload problem (Zhang et al., 2017). Notably, many successful recommendations systems share aspects of feature representation learning and have been widely applied in many online services such as electronic commerce (Ma et al., 2020) and social networks (Botangen et al., 2020). Existing methods for recommendation systems can roughly be categorized into three classes: content-based recommendation, collaborative filtering (CF), and hybrid methods (Batmaz et al., 2019). The content-based recommendation methods learn the descriptive features of items, calculate the similarity between new items and user-liked items based on these features, and generate the final recommendation (Lops et al., 2019). The collaborative filtering methods discover the inclinations of users by considering the user's historical behavior and produce recommendations (Dong et al., 2021). Hybrid recommendation methods leverage multiple approaches together and try to combine the advantages of these approaches.

Recently, collaborative filtering methods have achieved superior performance for the advantages of effectiveness and efficiency, which have far-ranging consequences in practical applications of recommendation systems (Su and Khoshgoftaar, 2009). Most of the traditional collaborative filtering methods are based on matrix factorization (MF), which combines good scalability with predictive accuracy (Luo et al., 2020). The main intuition behind these approaches is to decompose the rating matrix into user and item-based profiles, which allows the recommendation system to treat different temporal aspects separately (Yehuda et al., 2009). However, MF-based methods have inherent limitations in feature representation learning for the recommendation, which prevent further development of these approaches.

On the other hand, deep learning techniques have recently achieved great success in the computer vision and natural language processing fields. Such techniques show great potential in learning feature representations. Therefore, researchers have begun to apply deep learning methods to the field of recommendations (Salakhutdinov et al., 2007). They use a restricted Boltzmann machine instead of the traditional matrix factorization to perform the CF, and Georgiev and Nakov,(2013) expanded the work by incorporating the correlation between users and between items. In addition, Wang et al. (2015), proposed a hierarchical Bayesian model that uses a deep learning model to obtain content features and a traditional CF model to address the rating information. These methods, based on deep learning techniques, more or less make recommendations by learning the content features of items. These methods are not applicable when we are unable to obtain the contents of items. Therefore, enhancing the effectiveness of feature learning is significant. Recent studies have shown that deep neural networks can learn more abstract and higher-level feature representations (Yi et al., 2018), which has made remarkable progress in improving recommendation performance (Chae et al., 2019). For example, He et al. (2017) proposed a general recommendation framework called Neural Network-based Collaborative Filtering, in which a deep neural network is utilized for learning the interaction between user and item features. As we can see, among all the deep neural network-based recommendation methods, many frameworks are realized on top of the autoencoder model, which is one of the most successful deep neural networks and has also been actively adopted as a CF model recently (Shuai et al., 2017; Zhuang et al., 2017c; Chae et al., 2019; Zhong et al., 2020). For example, Zhang et al. proposed a hybrid collaborative filtering framework based on an autoencoder that incorporated auxiliary information for semantic rich representations teaching (Shuai et al., 2017).

Though the autoencoder-based methods have achieved fairly good performance for personalized recommendation, there are two main problems that prevent the further development of these methods. The first is the utilization of auxiliary information from users or items, since the rating matrix in real-world applications is usually very sparse, which inevitably leads to a significant recommendation performance degradation. Most existing methods only introduce some obvious attributes, such as the age, gender, and occupation of users, or the title, release date, and genres of items. The key factors of collaborative filtering, such as the reviews of items by users, have rarely been incorporated into the autoencoder-based networks. The second problem is the optimization of neural networks. When training models to incorporate side information about items and users, the dimensions of the input and output layers are required to be equal in autoencoder-based networks, which greatly limits the scalability and flexibility of networks.

To address these problems, we propose a feature representation learning method for personalized recommendation in this paper which extends items features with knowledge graph via triple-autoencoder (KGTA for short). Specifically, the comment information between users and items is first encoded as sentiment classification. These features are then applied as the input to the autoencoder for generating the auxiliary information of items, which can be used to introduce the comment information from users to items to solve the incorporating problem of auxiliary information. Secondly, the item-based rating, the side information, and the generated comment representations are incorporated into the

semi-autoencoder for reconstructed output. It aims to address the second problem, that the dimensions of the input and output layer are required to be equal. Finally, the reconstructed output generated by the semi-autoencoder is input into a third autoencoder for personalized recommendation. Experimental results on several datasets demonstrate the effectiveness of our proposed method compared to other state-of-the-art matrix factorization methods and deep-based methods.

In summary, the main contributions of our work can be distilled into the following:

- To incorporate the key information between users and items, the comments from each user for item are encoded and reconstructed as the auxiliary information
- To optimize the neural networks, a serial connection of semi-autoencoders and autoencoders are designed to learn more abstract and higher-level feature representations for personalized recommendation
- Extensive experiments on several datasets were conducted to confirm the effectiveness of the proposed method compared to other state-of-the-art matrix factorization methods and deep-based methods

2 RELATED WORK

In this section, we survey the related works of feature representation learning, personalized recommendation methods, and collaborative filtering[1,2].

2.1 Feature Representation Learning

Feature representation learning refers to learning data representations that make it easier to extract useful information in downstream machine learning tasks (Bengio et al., 2013). The last decades have witnessed a vast amount of research and application on feature representation learning in multiple disciplines. For example, in the field of biomedicine and bioinformatics, Wei et al. (2019) developed a bioinformatics tool for the generic prediction of therapeutic peptides. An adaptive feature representation learning method is proposed for different peptide types in the tool. Alshahrani et al. (2017) proposed a knowledge representation learning method with symbolic logic and automated reasoning, which can be applied to biological knowledge graphs for tasks such as finding candidate genes for diseases and protein-protein interactions. Li et al. (2021) proposed a triplet message mechanism to learn molecular representation based on graph neural networks, which can complete molecular property prediction and compound-protein interaction identification with few parameters and high accuracy.

Besides the fields of biomedicine and bioinformatics, feature representation learning has also been widely applied in other fields such as computer vision (Kim et al., 2017), knowledge engineering (Liu et al., 2016) and personalized recommendation (Zhuang et al., 2017b). For example, Wang et al. proposed a high-resolution representation learning network for visual recognition problems (Wang et al., 2020), which can maintain the representation being semantically strong and spatially precise. Xu et al. (2018) proposed an aggregation method for node representation learning that can adapt neighborhood ranges to nodes. It is especially suitable for graphs that have subgraphs with diverse local structures. Niu et al. (2020) proposed a rule and path-based joint embedding method for representation learning on knowledge graphs. The Horn rules and paths are leveraged in this method to enhance the accuracy and explainability of representation learning.

2.2 Personalized Recommendation

In recent decades, with the rapid development of the Internet, personalized recommendations have provoked a vast amount of attention and research (Qian et al., 2013). The advances in personalized recommendation have far-ranging consequences in many online services applications such as electronic commerce (Ma et al., 2020) and social networks (Li et al., 2017). For example, in Facebook, Gupta et al. (2020) conducted a detailed performance analysis of recommendation models on server-scale systems present in the data center. Botangen et al. (2020) proposed a probabilistic matrix factorization-based recommendation method that considers geographic location information for designing an effective and efficient Web service recommendation.

Good feature representations of data do contribute to many machine learning tasks, such as personalized recommendation. For example, Geng et al. (2015) proposed a deep method to learn the unified feature representations for both users and images. This representation from large, sparse, and diverse social networks obviously improves the recommendation performance. Liu et al. (2019) proposed a joint representation learning method for multimodal transportation recommendations, which aims to recommend a travel plan that considers various transportation modes. Ni et al. proposed a recommendation model based on deep representation teaching (Ni et al., 2021). It contained information preprocessing and feature representation modules to generate the primitive feature vectors and the semantic feature vectors of users and items, respectively.

2.3 Collaborative Filtering

In personalized recommendations, the collaborative filtering (CF) methods aim to discover users' preferences through the interactions between users and items. Existing CF methods can be roughly categorized into two classes: matrix factorization methods and deep neural network methods.

In the matrix factorization methods, these methods have difficulty in processing sparse data and have poor generalization ability, but they have low time and space complexity and good scalability. Lee et al. proposed the classical non-negative matrix factorization (NMF) model (Lee and Seung, 2001), which can decompose the rating matrix into user and item profiles. Along this line, Sun et al. proposed a

[1]http://files.grouplens.org/datasets/movielens/ml-100k.zip.
[2]http://files.grouplens.org/datasets/movielens/ml-1m.zip.

Probabilistic Matrix Factorization (PMF) model that scales linearly with the number of observations and performs well on very sparse and imbalanced datasets (Salakhutdinov and Mnih, 2007). In light of PMF, Salakhutdinov et al. also proposed a Bayesian Probabilistic Matrix Factorization (BPMF) model (Salakhutdinov and Mnih, 2008), which controlled model capacity automatically by placing hyper-priors over the hyper-parameters to avoid over-fitting. Koren proposed combining the factor and neighborhood models for a more accurate recommendation performance (Koren, 2008), which further extends the model to exploit both explicit and implicit feedback by the users. In recent years, to address the problem that the attributes of users are often scarce for reasons of privacy, Rashed et al. (2019) proposed a nonlinear co-embedding GraphRec model, which treats the user-item relation as a bipartite graph and constructs generic user and item attributes via the Laplacian of the user-item co-occurrence graph.

Recently, due to the powerful ability of deep learning methods, remarkable progress has been made in learning higher-level and abstract representations for personalized recommendations (Wang et al., 2015; Yu et al., 2019). These methods have nonlinear transformation and powerful representation learning ability, but poor interpretability, large data requirements, and extensive hyper-parameter tuning. For example, He et al. (2017) proposed a general recommendation framework that designs a deep neural network to learn the interaction between a user and item features. Meanwhile, to address the cold start problem and improve performance for personalized recommendations, Ni et al. (2022) proposed a two-stage embedding model to improve recommendation performance with auxiliary information. In this method, two sequential stages, graph convolutional embedding and multimodal joint fuzzy embedding, are designed to fully exploit item multimodal auxiliary information. Among all the deep learning methods for personalized recommendation, we realize many successful frameworks on top of the autoencoder, which is one of the most successful deep neural networks and has also been actively adopted as a CF model recently (Shuai et al., 2017; Zhuang et al., 2017c; Chae et al., 2019; Zhong et al., 2020). For example, Zhuang et al. (2017c) proposed a dual-autoencoder model for recommendation, which simultaneously learns the user-based and item-based features with the autoencoder model. Zhu et al. (2021) proposed a collaborative autoencoder model for personalized recommendation, which learns the hidden features of users and items with two different autoencoders for capturing different characteristics of the data.

3 PRELIMINARIES

3.1 Autoencoder

The autoencoder model aims to minimize the distance between the input and the reconstructed output. The basic autoencoder network (Bengio, 2009) generally consists of an input layer, an output layer, and one or more hidden layers. Given the input as $x \in \mathfrak{R}^{m\times n}$, when there is only one hidden layer, the encoding and decoding layer of autoencoder can be represented as follows:

$$\xi = f(Wx + b), \tag{1}$$

$$x' = g(W'\xi + b'), \tag{2}$$

where $W \in \mathfrak{R}^{k\times m}$, $W' \in \mathfrak{R}^{m\times k}$ and $b \in \mathfrak{R}^{k\times 1}$, $b' \in \mathfrak{R}^{m\times 1}$ are the weighting matrices and bias vectors, respectively. f and g are the nonlinear activation functions of the encode and decode layers, respectively. In our experiments, the sigmoid and identity functions are introduced as f and g. The objective function of the autoencoder can be shown as follows:

$$\min_{W,b,W',b'} J_r = \left\| x' - x \right\|^2. \tag{3}$$

3.2 Semi-Autoencoder

In recent years, many autoencoder-based recommendation methods have achieved fairly good results with the advantages of no labeling requirement and fast convergence speed. However, the classic autoencoder model has the restriction that the dimensions of the input and the output layer must be equal, which has a great impact on introducing auxiliary information for solving the sparse problem of the rating matrix.

To address this problem, a semi-autoencoder model was proposed and generalized into a hybrid CF method for rating prediction (Shuai et al., 2017). Compared with traditional autoencoders, the input layer of semi-autoencoders is longer than the output layer, so semi-autoencoders can be utilized to capture different nonlinear feature representations and reconstructions flexibly by extracting different subsets from the inputs, and it is easy to incorporate side information into the input layer effectively to improve the item feature representation for better recommendation performance. The whole framework of the semi-autoencoder is shown in **Figure 1**, the left and right parts of **Figure 1** show the two cases in which the output layer is longer than the input layer and the output layer is shorter than the input layer, respectively. We observe that the basic framework of a semi-autoencoder is the same as that of a classical autoencoder model, which also includes an input layer, an output layer, and one or more hidden layers. Furthermore, in the right part of **Figure 1**, we can observe that the shorter output layer is the reconstruction of certain parts of the input, and the remaining part in the semi-autoencoder model is auxiliary information to learn better feature representations for addressing the sparse problem of the rating matrix.

4 METHODOLOGY

The whole framework of our proposed recommendation method with knowledge graph via triple-autoencoder (KGTA for short) is illustrated in **Figure 2**, which encompasses three main components. The first one is the representational learning of the comment information between users and items. The comments from users on each item are divided into positive and negative categories. Then the first autoencoder was introduced to reduce the dimensionality of this comment information. The second one is the learning of all the auxiliary information. A semi-autoencoder is utilized to incorporate the side information, the extended features from the knowledge graph, and the generated comment features into the item-based rating. Finally, the low-dimensional output of the semi-autoencoder is input into the

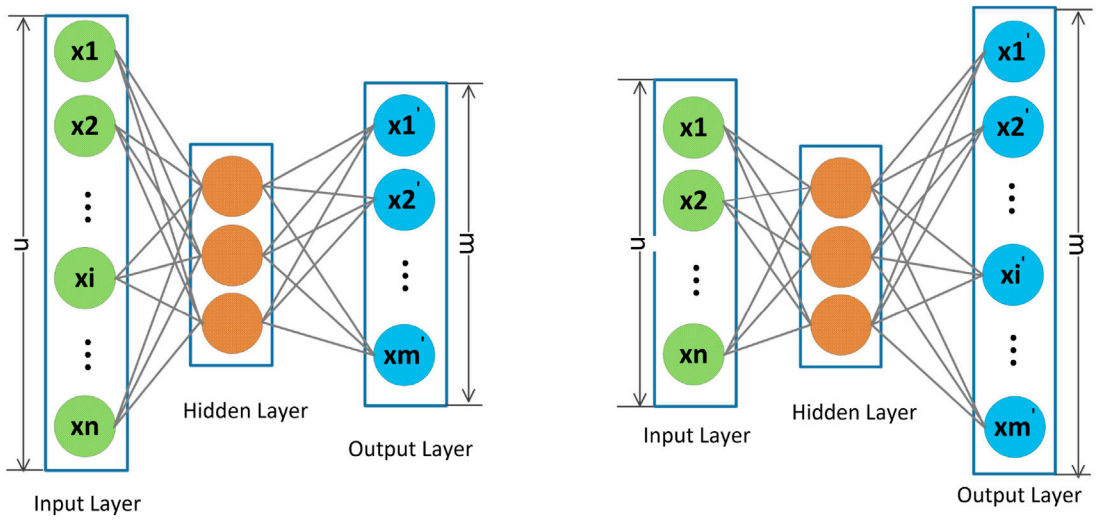


FIGURE 1 | Illustration of a semi-autoencoder where the input and output layers can be inconsistent. The length of the input layer is longer/shorter than the output layer in the left/right part.

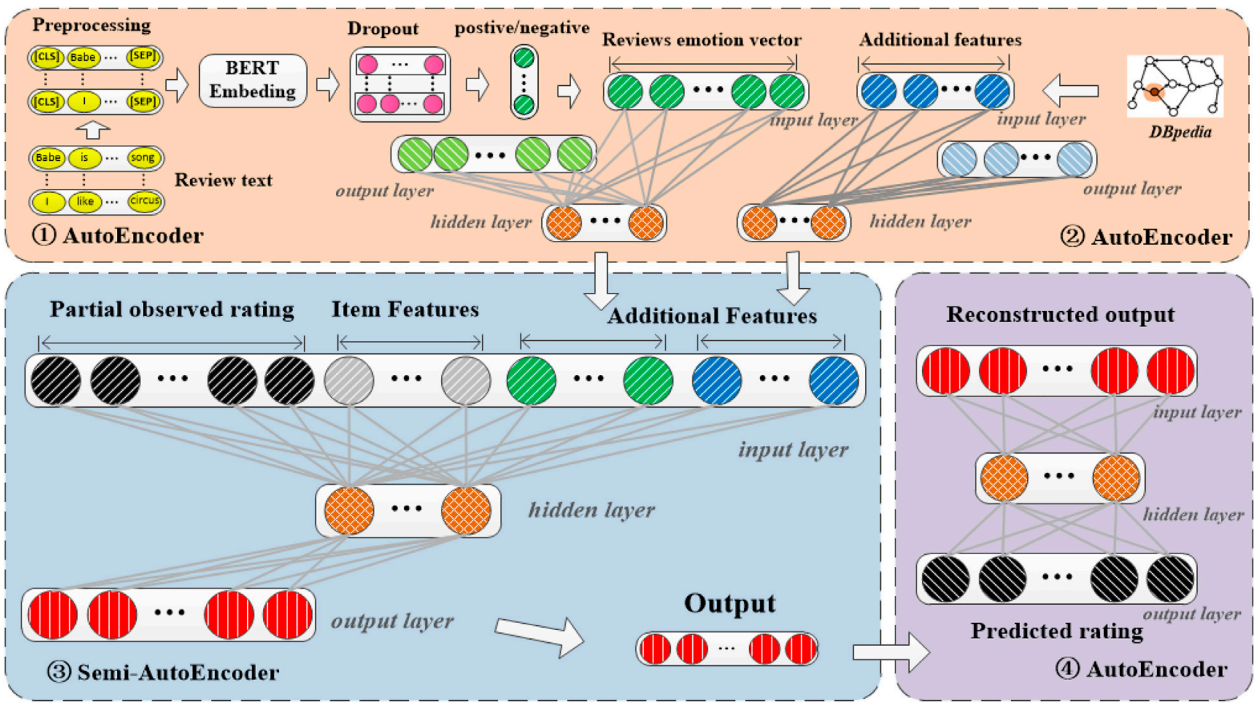


FIGURE 2 | Whole framework of the proposed KGTA

third autoencoder. Different from the semi-autoencoder model that only approximates the item-based rating; the third component tries to reconstruct all the input for the recommendation[3,4].

In the following, first, the commonly used notations in this paper are listed in **Table 1**, and then, the model of KGTA is described in detail.

4.1 Notations

Some important notations used in this paper and their descriptions are listed in **Table 1**.

[3]http://www.librec.net/download.html.
[4]http://github.com/hoyeoplee/MeLU.

TABLE 1 | Important notations used in this article and their descriptions.

Notations	Descriptions
R	The rating matrix
A	The attributes vectors of all items
S	The reconstructed comment vectors of all items
L	The language vectors of all items
R'	The prediction matrix $R' \in R^{n\times m}$
m	The number of users
n	The number of items
r^{μ}	The column of rating matrix
r^{i}	The row of rating matrix
k	The features dimension of hidden units
h	The number of hidden units
x_i	The ith instance of original input
x_i'	The reconstructed output of x_i
ξ	The hidden feature representation matrix
W, W'	The map and remap weight matrix
b, b'	The map and remap bias vectors
$\bullet$	The element-wise product of vectors or matrices

4.2 Comment Information Features

The personalized recommendation is to predict the interest of a user in an item based on the rating matrix information. Since the rating matrix in real-world scenarios is usually very sparse, many methods have introduced auxiliary information to address this problem. However, most existing methods only introduce some obvious attributes and ignore the key factors, such as the comments from users on each item, of collaborative filtering. To address this problem, our method learns the comment information features between users and items with the first autoencoder. The details can be seen in the upper left of **Figure 2**.

In our method, we take natural language text as the input for sentiment classification and output emotion score $\in \{1, -1\}$. -1 represents negative emotion and 1 represents positive emotion. Our method has two stages from input sentence to output score, which are described below.

In the first stage, we perform the following preprocessing steps on the comment text before we feed it into the model. First, we remove all the digits, punctuation symbols, and accent marks, and convert everything to lowercase. Secondly, we then tokenize the text using the WordPiece tokenizer (Schuster and Nakajima, 2012). It breaks the words down into their prefix, root, and suffix to better handle unseen words. Finally, we add the [CLS] and [SEP] tokens at the appropriate positions.

In the second stage, we build a simple architecture with just a dropout regularization (Srivastava et al., 2014) and a softmax classifier layer on top of the pretrained BERT layer. The upper left corner of **Figure 2** shows the overall architecture of our sentiment classification model. There are four main stages. The first is the processing step, as described earlier. Then we compute the sequence embedding from BERT. We then apply a dropout with a probability factor of 0.1 to regularize and prevent over-fitting. Finally, the softmax classification layer will output the probabilities of the input text belonging to each of the class labels such that the sum of the probabilities is 1. The softmax layer is just a fully connected neural network layer with the softmax activation function. The output node with the highest probability is then chosen as the predicted label for the input.

Given the rating matrix $R \in \mathfrak{R}^{m\times n}$, where m and n denote the number of users and items respectively. For each item, the comments from each user are classified by sentiment using BERT (Devlin et al., 2018) first, and then we obtain the comment feature vector c_i for each item. Since the comment information from users to items is usually sparse, just like the rating matrix, the first autoencoder was introduced for feature dimensionality reduction and representation learning. The process of the first autoencoder can be shown as follows:

$$\xi_s = f(W_s C + b_s), \tag{4}$$

$$s = g(W_s'\xi_s + b_s'), \tag{5}$$

where $W_s \in \mathfrak{R}^{k_1\times n}$ and $W_s' \in \mathfrak{R}^{n\times k_1}$ are the weighting matrices, $b_s \in \mathfrak{R}^{k_1\times 1}$ and $b_s' \in \mathfrak{R}^{n\times 1}$ are the bias vectors, f and g are the functions of nonlinear activation, and k_1 is the feature dimension of hidden units. The hidden features of the first autoencoder, i.e., the low-dimensional representations of s, are denoted as S^I, which are incorporated into the second semi-autoencoder for capturing different representations and reconstructions by sampling different subsets from all the inputs.

4.3 Co-Embeddings With the Semi-Autoencoder

After obtaining the reconstructed comment features, a semi-autoencoder is introduced to incorporate the item rating vector r_i and other auxiliary information such as attributes vector a_i, reconstructed comment features s_i, and the KG-extended features l_i. The input of the semi-autoencoder can be defined as $con(r_i, a_i, s_i, l_i)$

$$con(r_i, a_i, s_i, l_i) = connection\ of\ r_i,\ a_i,\ s_i,\ and\ l_i. \tag{6}$$

The $con(R^I, A^I, S^I, L^I) \in \mathfrak{R}^{n\times(m+y+k_1+k_2)}$ refers to the connection of R^I, A^I, S^I and L^I, where $R^I \in \mathfrak{R}^{n\times m}$ represents the item-based rating vectors, $A^I \in \mathfrak{R}^{n\times y}$ represents the attribute vectors of all items, which are the obvious attributes such as the title, release date, and genres in movie recommendation datasets, $S^I \in \mathfrak{R}^{n\times k_1}$ represents the reconstructed comment features for all n items, $L^I \in \mathfrak{R}^{n\times k_2}$ represents the language vectors collected from the knowledge graph and autoencoder. Considering that the experiments are conducted on MovieLens datasets, the languages of the movies are obtained from open KGs such as DBpedia, and the languages are encoded with the multi-hot method and input into the autoencoder model for learning the hidden representations L^I. The process of L^I learning is consistent with that of S^I, the details can be seen in the upper right of **Figure 2**.

Then the $con(R^I, A^I, S^I, L^I)$ is input into the second autoencoder, i.e. a semi-autoencoder, to learn the compressed

reconstructed output, the encode stage of the semi-autoencoder can be defined as (7)

$$\xi = f\left(W con\left(R^I, A^I, S^I, L^I\right) + b\right), \tag{7}$$

where $W \in \mathfrak{R}^{(m+y+k_1+k_2)\times k}$ and $b_1^I \in \mathfrak{R}^k$ are the weight matrix and bias item, respectively, k is the feature dimension of the hidden layer, and f is the sigmoid function for nonlinear activation. Then, the decode stage can be shown as follows:

$$R'_{semi} = g\left(W'\xi + b'\right). \tag{8}$$

Similarly, where $W' \in \mathfrak{R}^{k\times m}$ and $b_2^I \in \mathfrak{R}^m$ are the weight matrix and bias item of decoding layer respectively, g is the identity function for the activation function. Notably, the SGD (stochastic gradient descent) method is utilized in the semi-autoencoder for model optimization. The details can be seen in the bottom left of **Figure 2**.

4.4 Triple-Autoencoder for Recommendation

From **Eqs. 7**, **8**, we can obviously observe that the output of a semi-autoencoder model is the reconstruction of a certain part of the inputs. When computing the loss function, the result of the semi-autoencoder is a reconstruction of the rating matrix R^I instead of the whole input $con(R^I, A^I, S^I, L^I)$, which may result in a performance degradation for recommendation. To this end, we design the third autoencoder model to learn the reconstruction of the whole input, that is triple-autoencoder for the recommendation. The encode and decode stage of the triple-autoencoder can be shown as follows:

$$R' = g\left(W'_t f\left(W_t R'_{semi} + b_t\right) + b'_t\right). \tag{9}$$

To avoid over-fitting, the ℓ_2 norm regularization of the weight matrix W_t and W'_t is added to the objective function, which can be shown as follows:

$$J_r = \left(\|W_t\|_2^2 + \left\|W'_t\right\|_2^2\right). \tag{10}$$

Thus, the objective function of the triple-autoencoder can be shown as follows:

$$J_{item} = \left\|\left(R' - R'_{semi}\right)\right\|^2 + \alpha J_r, \tag{11}$$

where α is the trade-off parameter that controls the balance of regularization terms. To minimize the distance between the input R'_{semi} and the output R', the deviations are minimized to obtain representations for the recommendation. When the model converges, the output layer of the triple-autoencoder is the prediction matrix R' for the recommendation, the details can be shown in the bottom right of **Figure 2**. Details of the proposed KGTA are summarized in Algorithm 1.

Algorithm 1. Recommendation with knowledge graph *via* triple-autoencoder (KGTA)

Input: The rating matrix $R \in \mathfrak{R}^{m\times n}$, trade-off parameter α, the number of latent feature dimensions k, the number of hidden units h, and the dimension of item attribute vector, comment vector and languages auxiliary vector is y, k_1 and k_2.
Output: The predicated rating matrix R'.
Get the additional information vector a_i for each item;
Input the comment between users and each item into the BERT model, and get the raw comment vector c_i from the sentiment classification model;
Get the reconstructed comment features s_i by Eq. (5);
Get the language information for each item from open KG and encode with multi-hot method;
Get the reconstructed language vectors l_i;
Input the concatenation vectors $con\left(R^I, A^I, S^I, L^I\right)$ of the items rating vector R^I, attribute vector A^I, comment features S^I and languages features L^I into semi-autoencoder;
Initialize W, b and W', b' randomly respectively;
Minimize $\left\|R'_{semi} - R^I\right\|_2^2$ with SGD method until convergence;
Input the reconstructed vectors R'_{semi} into autoencoder;
Initialize W_t, b_t and W'_t, b'_t randomly respectively;
Minimize Eq.(11) with SGD method until convergence;
return The predicated rating matrix R';

5 EXPERIMENTS

In this section, experiments are conducted on two datasets, MovieLens 100K and MovieLens 1M, to evaluate the effectiveness of our proposed KGTA. In the following, we first introduce the details of two experimental datasets. Secondly, the compared methods, including the MF-based and deep neural network-based methods, are given. In addition, the evaluation metrics such as MAE and RMSE are also presented. Then, the comparative experimental results and their observations are presented in detail. Finally, the main properties such as parameter sensitivity are analyzed for certain datasets.

5.1 Datasets

The details of two real-world datasets used in the experiments are listed in **Table 2**, including rating density, the number of users, items, and ratings.

MovieLens 100K1: it is a well-known and most widely applied dataset for evaluating recommendation performance. There are 943 users and 1,682 movies with 100,000 ratings on a scale of 1–5, and each user rated at least 20 movies. In MovieLens 100K, item attributes such as the title, release date, and genres of movies are also provided for improving recommendation performance.

MovieLens 1M2: It is an enlarged version of the Movielens 100K dataset, which has also been widely applied in the recommendation. It has 6,040 users and 3,706 movies with 1,000,209 ratings. Similar to Movielens 100K, the ratings are scaled from 1 to 5, and auxiliary information such as movie title, release date, and category are also provided.

5.2 Compared Methods and Evaluation Metrics

5.2.1 Compared Methods

To evaluate the effectiveness of the proposed KGTA, the following matrix factorization methods, meta-learning methods, and deep neural network methods were conducted:

- Non-negative matrix factorization (NMF) (Lee and Seung, 2001). It is the basic matrix factorization method for the

TABLE 2 | Details of the three datasets used in our experiments.

Dataset	Number of users	Number of items	Number of ratings	Rating density (%)
MovieLens 100K	943	1,682	100,000	6.3
MovieLens 1M	6,039	3,883	1,000,209	4.27

TABLE 3 | The performance of RMSE on MovieLens 100K and MovieLens 1M datasets.

Datasets	Methods	Proportion of training data			
MovieLens 100K	-	**50%**	**60%**	**70%**	**80%**
	NMF	0.991	0.976	0.965	0.960
	SVD++	0.943	0.927	0.915	0.909
	MetaHIN	1.062	1.046	1.041	1.032
	MeLU	1.154	1.144	1.132	1.121
	AutoRec	1.023	1.003	0.981	0.964
	HCRSA	0.948	0.937	0.923	0.919
	PRKG	0.928	0.917	0.907	0.899
	KGTA	**0.859**	**0.847**	**0.840**	**0.832**
MovieLens 1M	NMF	0.928	0.925	0.921	0.918
	MetaHIN	1.024	0.993	0.965	0.959
	MeLU	1.082	1.038	1.008	0.973
	NCF	0.914	0.911	0.909	0.907
	AutoRec	0.914	0.905	0.896	0.888
	HCRSA	0.903	0.892	0.884	0.874
	PRKG	0.885	0.875	0.868	0.861
	KGTA	**0.823**	**0.814**	**0.807**	**0.8**

The bold values provided in Table 3 represent the experimental results of our proposed method (KGTA) and are the best results among all the comparison methods.

recommendation. In our experiments, we use the generalized Kullback–Leibler divergence as the update rules in NMF.

- Singular value decomposition plus (SVD++) (Koren, 2008). It exploits explicit and implicit feedback from users to combine the latent factor model and the neighborhood model into a unified model for the recommendation.
- Meta-learned user preference estimator (MeLU) (Lee et al., 2019). It estimates user preferences based on a small number of items to alleviate the cold start problem for the recommendation.
- Meta-learning method for cold start recommendation on Heterogeneous Information Networks (MetaHIN) (Lu et al., 2020). It creates a semantic-enhanced task constructor for exploring rich semantics, and a co-adaptation meta-learner with semantic- and task-wise adaptations within each task.
- Neural collaborative filtering (NCF) (He et al., 2017). It is a general recommendation framework that uses designs a deep neural network to learn the interaction between a user and item features.
- Item-based recommendation via autoencoder (AutoRec) (Sedhain et al., 2015). It is the first autoencoder framework in the recommendation, which learns the effective feature representations of items for collaborative filtering.
- Hybrid Collaborative Recommendation via Semi-Autoencoder (HCRSA) (Shuai et al., 2017). It is a hybrid collaborative filtering framework based on the semi-autoencoder, which incorporates auxiliary information for semantic rich representation learning.
- Personalized recommendation with knowledge graph *via* dual-autoencoder (PRKG) (Yang et al., 2021). The side information of items is extracted from DBpedia and encoded into low-dimensional representations in this method, and a semi-autoencoder is introduced to incorporate this auxiliary information for the recommendation.

5.2.2 Implementation Details and Parameter Settings

The PREA toolkit (Lee et al., 2014) is adopted for the implementation of MF-based methods such as NMF and SVD++. For the methods of MeLU, MetaHIN, and HCRSA, we re-compile the source code as 4, 5, and 6. The default parameters of these three methods remain unchanged as reported in the original paper in the MovieLens dataset. For the method AutoRec, we select an item-based autoencoder that can achieve better performance than the user-based autoencoder. For fairness, the parameters of AutoRec and PRKG are consistent with ours in all two datasets. In our experiments, we set $\alpha = 0.1$ after some preliminary tests for all datasets. The maximum number of iterations in gradient descent is set at 300. The number of hidden units is set at 300 for all datasets[5,6].

5.2.3 Evaluation Metrics

In the experiments, we introduced root mean square error (RMSE) to measure the performance of our proposed KGTA and all compared methods in the recommendation, which can be shown as (12). It is worth mentioning that the smaller value of RMSE indicates better results.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{r_{u,i} \in \mathrm{TestSet}} \left(r_{u,i} - r'_{u,i}\right)^2}{|\mathrm{TestSet}|}}, \tag{12}$$

where $r_{u,i}$ and $r'_{u,i}$ represent the original rating matrix and the predication matrix, respectively.

5.3 Experimental Results

For each data set, the percentages of 50%, 60%, 70%, and 80% are sampled into training data, respectively, and the rest are used for test data. The experimental results of RMSE on the MovieLens 100K and MovieLens 1M datasets are recorded in **Table 3** and **Figures 3**, **4** respectively. Notably, all the results are obtained by

[5]https://github.com/rootlu/MetaHIN.

[6]https://github.com/cheungdaven/semi-ae-recsys.

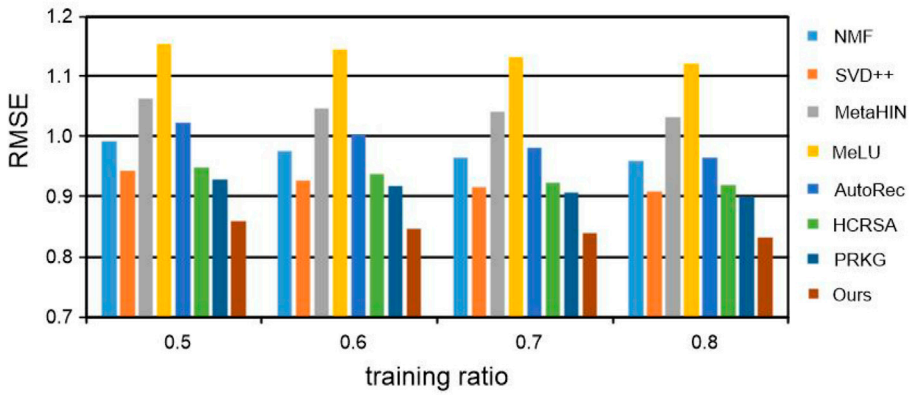


FIGURE 3 | RMSE of our KGTA and compared methods on the MovieLens 100K dataset.

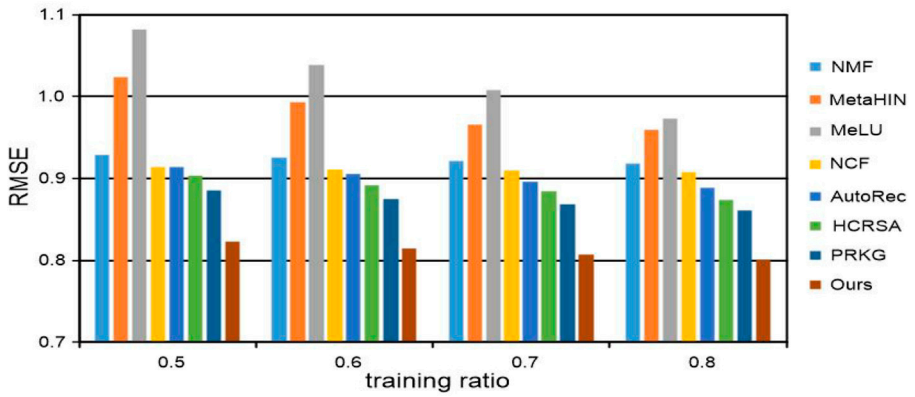


FIGURE 4 | RMSE of our KGTA and compared methods on the MovieLens 1M dataset.

repeating the experiments 5 times and taking the average value. From all the results, we have the following insightful observations:

- The performance of all recommended methods is improved with the increase of training data. It is worth mentioning that meta-learning methods such as MetaHIN and MeLU have not changed much, which may be due to the meta-learning methods being designed to alleviate the cold start problem for the recommendation.
- Generally, among the three types of methods, meta-learning methods perform the worst, probably because they are primarily designed to address the cold start problem. The methods for deep neural networks can achieve more desirable performance in most cases than both meta-learning and matrix factorization methods, which reveals the powerful ability of deep neural networks in learning the feature representations for personalized recommendation.
- Among all the deep neural network methods for recommendation, our KGTA is significantly better than NCF and AutoRec, which shows the superiority of introducing auxiliary information for addressing the problem of data sparsity and improving the performance of personalized recommendations.
- In the method of HCRSA, attributes such as the title, release date, and genre of a movie are introduced to the semi-autoencoder model for prediction. From the results listed in **Table 3** and **Figures 3**, **4**, we can observe that our KGTA consistently outperforms HCRSA, which demonstrates the superiority of incorporating the key factors of collaborative filtering, such as the comments from users to items, to improve the performance of personalized recommendation.
- Although both the methods introduce auxiliary information, our KGTA outperforms PRKG by up to 7 RMSE points on two well-known datasets, which shows the advantage of designing a serial connection of semi-autoencoder and autoencoder for learning more abstract and higher-level feature representations in the recommendation.
- Overall, the proposed KGTA performs best in all groups, which validates the effectiveness of incorporating the key information between users and items and designing a serial connection of semi-autoencoder and autoencoder for the

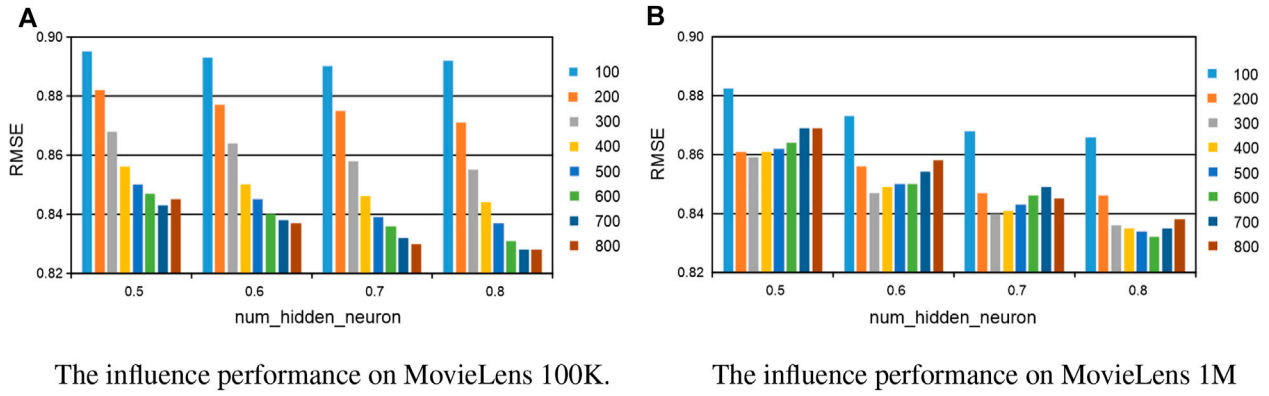


FIGURE 5 | The parameter influence of the number of hidden layer neurons on our KGTA. **(A)** The influence performance on MovieLens 100K. **(B)** The influence performance on MovieLens 1M.

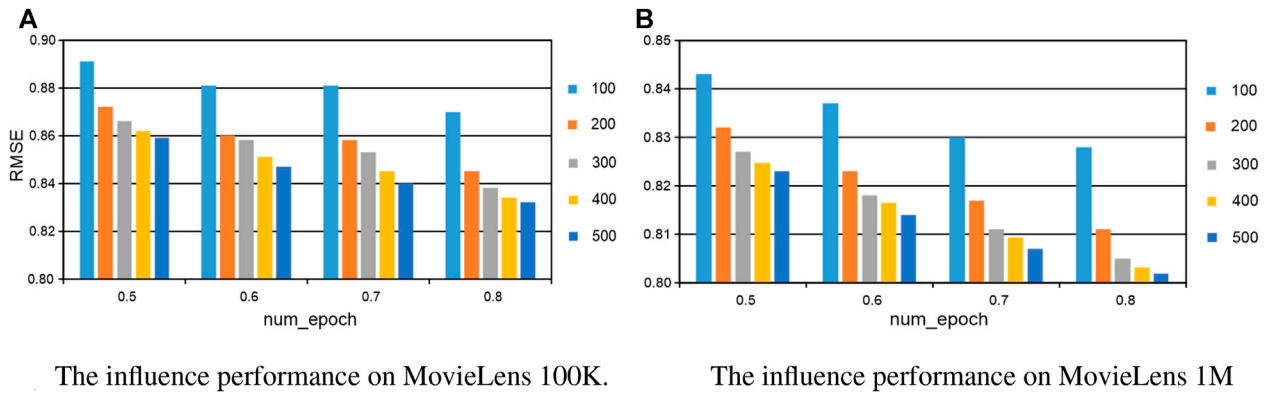


FIGURE 6 | The parameter influence of the number of epochs on our KGTA. **(A)** The influence performance on MovieLens 100K. **(B)** The influence performance on MovieLens 1M.

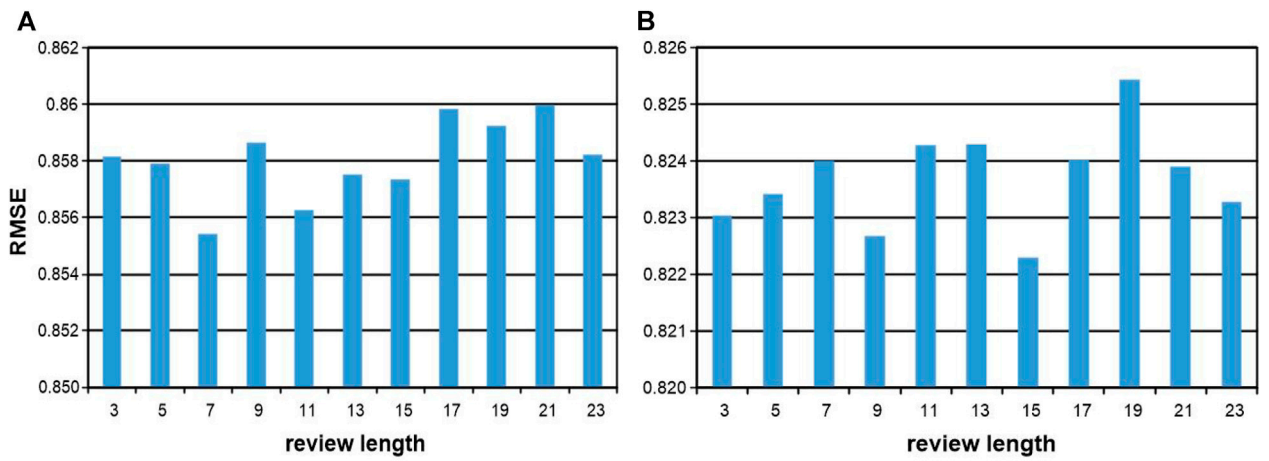


FIGURE 7 | The parameter influence of the length of comments on our KGTA. **(A)** The influence performance on MovieLens 100K. **(B)** The influence performance on MovieLens 1M.

recommendation. It should be noted that KGTA can achieve stable performance in both MovieLens 100K and MovieLens 1M. These results demonstrate that our KGTA can perform well even if the dataset is sparse.

5.4 Parameter Sensitivity

In this section, we investigate the influence of parameters in our proposed method, including the number of hidden layer neurons, the number of epochs, and the length of comments in the training. When one parameter is changed, the others are fixed in the experiments. The number of hidden layer neurons is varied from 100 to 800, the number of epochs is altered from 100 to 500, and the length of comments is sampled from the set {3, 5, 7, 9, 11, 13, 15, 17, 19, 21, and 23}. In the experiments, the validation was conducted on MovieLens 100K and MovieLens 1M, respectively. For the number of hidden layer neurons and the number of epochs, the experiments are conducted with 50%–80% of the training data. All the results are reported in **Figures 5**, **6**, and we set *the number of epoch* = 500 for both datasets, *the number of hidden layer neurons* = 300 and *thenumberofhiddenlayerneurons* = 400 for MovieLens 100K and MovieLens 1M, respectively. For the length of comments, experiments are conducted on 50% of the training data with the best and most stable parameters configuration of the number of hidden layer neurons and epoch, all the results are reported in **Figure 7**, and we set *the length of comments* = 5 for both the datasets.

6 CONCLUSION

In this paper, we propose a feature representation learning method with a knowledge graph via triple-autoencoder for personalized recommendation called KGTA. We propose a serial connection between the semi-autoencoder and autoencoder methods. In our method, we were able to incorporate side information distilled from DBpedia for more useful item feature representations, and the key factors of collaborative filtering, such as comment information between users and items, are incorporated into the autoencoder as auxiliary information. Moreover, the item-based rating and all the external information are incorporated into the semi-autoencoder to obtain low-dimensional information representation. Finally, the reconstructed output generated by the semi-autoencoder is input into a third autoencoder to learn better feature representations for personalized recommendation. Extensive experiments demonstrate the proposed method outperforms other state-of-the-art methods in effectiveness. In future work, we will try to achieve superior performance by incorporating less information and utilizing an attention network to strengthen the feature integration or without auxiliary information from the open knowledge base.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding authors.

AUTHOR CONTRIBUTIONS

YG: methodology, software, formal analysis, and writing. XX: conceptualization, supervision, and project administration. YZ: data curation, visualization, and writing. XS: visualization and validation.

FUNDING

This research was partially supported by the National Natural Science Foundation of China (61906060 and 62076217), Yangzhou University Interdisciplinary Research Foundation for Animal Husbandry Discipline of Targeted Support (yzuxk202015), the Opening Foundation of the Key Laboratory of Huizhou Architecture in Anhui Province under grant HPJZ-2020-02, and the Open Project Program of the Joint International Research Laboratory of Agriculture and Agri-Product Safety (JILAR-KF202104).

REFERENCES

Alshahrani, M., Khan, M. A., Maddouri, O., Kinjo, A. R., Queralt-Rosinach, N., and Hoehndorf, R. (2017). Neuro-Symbolic Representation Learning on Biological Knowledge Graphs. *Bioinformatics* 33 (17), 2723–2730. doi:10.1093/bioinformatics/btx275

Batmaz, Z., Yurekli, A., Bilge, A., and Kaleli, C. (2019). A Review on Deep Learning for Recommender Systems: Challenges and Remedies. *Artif. Intell. Rev.* 52 (1), 1–37. doi:10.1007/s10462-018-9654-y

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (8), 1798–1828. doi:10.1109/tpami.2013.50

Bengio, Y. (2009). Learning Deep Architectures for AI. *Found. Trends® Mach. Learn.* 2 (1), 1–127. doi:10.1561/2200000006

Botangen, K. A., Yu, J., Sheng, Q. Z., Han, Y., and Yongchareon, S. (2020). Geographic-Aware Collaborative Filtering for Web Service Recommendation. *Expert Syst. Appl.* 151, 113347. doi:10.1016/j.eswa.2020.113347

Chae, D.-K., Kim, S.-W., and Lee, J.-T. (2019). Autoencoder-Based Personalized Ranking Framework Unifying Explicit and Implicit Feedback for Accurate Top-N Recommendation. *Knowl. Based Syst.* 176, 110–121. doi:10.1016/j.knosys.2019.03.026

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

Dong, B., Zhu, Y., Li, L., and Wu, X. (2021). Hybrid Collaborative Recommendation of Co-Embedded Item Attributes and Graph Features. *Neurocomputing* 442, 307–316. doi:10.1016/j.neucom.2021.01.129

Geng, X., Zhang, H., Bian, J., and Chua, T.-S. (2015). "Learning Image and User Features for Recommendation in Social Networks," in Proceedings of the IEEE International Conference on Computer Vision, ICCV 2015 took place at the CentroParque Convention Center in Santiago, Chile, December 7–13, 2015. 4274–4282. doi:10.1109/iccv.2015.486

Georgiev, K., and Nakov, P. (2013). "A Non-iid Framework for Collaborative Filtering with Restricted Boltzmann Machines," in International Conference on Machine Learning, Atlanta, GA, USA, June 16–21, 2013. (PMLR), 1148–1156.

Gupta, U., Wu, C.-J., Wang, X., Naumov, M., Reagen, B., Brooks, D., et al. (2020). "The Architectural Implications of Facebook's Dnn-Based Personalized Recommendation," in 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA), San Diego, CA, USA, February 22–26, 2020. 488–501. doi:10.1109/hpca47549.2020.00047

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. (2017). "Neural Collaborative Filtering," in Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, April 3–7, 2017. 173–182. doi:10.1145/3038912.3052569

Hohenberg, P., and Kohn, W. (1964). Inhomogeneous Electron Gas. *Phys. Rev.* 136 (3B), B864–B871. doi:10.1103/physrev.136.b864

Kim, D. H., Baddar, W. J., Jang, J., and Ro, Y. M. (2017). Multi-Objective Based Spatio-Temporal Feature Representation Learning Robust to Expression Intensity Variations for Facial Expression Recognition. *IEEE Trans. Affective Comput.* 10 (2), 223–236. doi:10.1109/TAFFC.2017.2695999

Koren, Y. (2008). "Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model," in Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24–27, 2008. 426–434.

Lee, D. D., and Seung, H. S. (2001). "Algorithms for Non-Negative Matrix Factorization," in International Conference on Neural Information Processing Systems, Shanghai, China, November 11–15, 2001. 556–562.

Lee, H., Im, J., Jang, S., Cho, H., and Chung, S. (2019). "Melu: Meta-Learned User Preference Estimator for Cold-Start Recommendation," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, August 4–8, 2019. 1073–1082.

Lee, J., Sun, M., Lebanon, G., and Sonnenburg, S. (2014). PREA: Personalized Recommendation Algorithms Toolkit. *J. Machine Learn. Res.* 13 (3), 2699–2703. doi:10.3166/EJC.18.485-495

Li, P., Li, Y., Hsieh, C. Y., Zhang, S., Liu, X., Liu, H., et al. (2021). Trimnet: Learning Molecular Representation from Triplet Messages for Biomedicine. *Brief Bioinform* 22 (4), bbaa266. doi:10.1093/bib/bbaa266

Li, Z., Fang, X., and Sheng, O. R. L. (2017). A Survey of Link Recommendation for Social Networks: Methods, Theoretical Foundations, and Future Research Directions. *ACM Trans. Manage. Inf. Syst. (TMIS)* 9 (1), 1–26. doi:10.1145/3131782

Liu, H., Li, T., Hu, R., Fu, Y., Gu, J., and Xiong, H. (2019). Joint Representation Learning for Multi-Modal Transportation Recommendation. *Proc. AAAI Conf. Artif. Intelligence* 33, 1036–1043. doi:10.1609/aaai.v33i01.33011036

Liu, Z., Sun, M., Lin, Y., and Xie, R. (2016). Knowledge Representation Learning: A Review. *J. Comput. Res. Develop.* 53 (2), 247. doi:10.7544/issn1000-1239.2016.20160020

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., et al. (2019). "Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations," in The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA 2019), Cairo, Egypt, March 28–30, 2019. 4114–4124.

Lops, P., Jannach, D., Musto, C., Bogers, T., and Koolen, M. (2019). Trends in Content-Based Recommendation. *User Model. User-Adap Inter.* 29 (2), 239–249. doi:10.1007/s11257-019-09231-w

Lu, Y., Fang, Y., and Shi, C. (2020). "Meta-Learning on Heterogeneous Information Networks for Cold-Start Recommendation," in Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 1563–1573. doi:10.1145/3394486.3403207

Luo, X., Yuan, Y., Chen, S., Zeng, N., and Wang, Z. (2020). Position-Transitional Particle Swarm Optimization-Incorporated Latent Factor Analysis. *IEEE Trans. Knowl. Data Eng.*, 1. doi:10.1109/tkde.2020.3033324

Ma, Y., Narayanaswamy, B., Lin, H., and Ding, H. (2020). "Temporal-Contextual Recommendation in Real-Time," in Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23–27, 2020. 2291–2299. doi:10.1145/3394486.3403278

Ni, J., Huang, Z., Cheng, J., and Gao, S. (2021). An Effective Recommendation Model Based on Deep Representation Learning. *Inf. Sci.* 542, 324–342. doi:10.1016/j.ins.2020.07.038

Ni, J., Huang, Z., Hu, Y., and Lin, C. (2022). A Two-Stage Embedding Model for Recommendation with Multimodal Auxiliary Information. *Inf. Sci.* 582, 22–37. doi:10.1016/j.ins.2021.09.006

Niu, G., Zhang, Y., Li, B., Cui, P., Liu, S., Li, J., et al. (2020). Rule-Guided Compositional Representation Learning on Knowledge Graphs. *Proc. AAAI Conf. Artif. Intell.* 34, 2950–2958. doi:10.1609/aaai.v34i03.5687

Qian, X., Feng, H., Zhao, G., and Mei, T. (2013). Personalized Recommendation Combining User Interest and Social Circle. *IEEE Trans. Knowl. Data Eng.* 26 (7), 1763–1777. doi:10.1109/TKDE.2013.168

Ramakrishnan, R., Dral, P. O., Rupp, M., and von Lilienfeld, O. A. (2015). Big Data Meets Quantum Chemistry Approximations: The δ-Machine Learning Approach. *J. Chem. Theor. Comput.* 11 (5), 2087–2096. doi:10.1021/acs.jctc.5b00099

Rashed, A., Grabocka, J., and Schmidt-Thieme, L. (2019). "Attribute-Aware Non-Linear Co-Embeddings of Graph Features," in Proceedings of the 13th ACM Conference on Recommender Systems (RecSys 2019), Copenhagen, Denmark, September 16–20, 2019. 314–321. doi:10.1145/3298689.3346999

Salakhutdinov, R., and Mnih, A. (2008). "Bayesian Probabilistic Matrix Factorization Using Markov Chain Monte Carlo," in Proceedings of the 25th international conference on Machine learning (ICML 2008), Helsinki, Finland, July 5–9, 2008. 880–887. doi:10.1145/1390156.1390267

Salakhutdinov, R., Mnih, A., and Hinton, G. (2007). "Restricted Boltzmann Machines for Collaborative Filtering," in Neural Information Processing Systems (NIPS 2007), Vancouver, British Columbia, Canada, December 4–7, 2006. 791–798. doi:10.1145/1273496.1273596

Salakhutdinov, R., and Mnih, A. (2007). "Probabilistic Matrix Factorization," in Proceedings of the 24th international conference on Machine learning (ICML 2007), ICML 2007 was held in conjunction with the 2007 International Conference on Inductive Logic Programming at Oregon State University in Corvallis, Oregon, June 20–24, 2007. 1257–1264.

Schuster, M., and Nakajima, K. (2012). "Japanese and Korean Voice Search," in 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012), Kyoto, Japan, March 25–30, 2012. (IEEE), 5149–5152. doi:10.1109/icassp.2012.6289079

Sedhain, S., Menon, A. K., Sanner, S., and Xie, L. (2015). "Autorec: Autoencoders Meet Collaborative Filtering," in Proceedings of the 24th International Conference on World Wide Web (WWW 2015), Florence, Italy, May 18–22, 2015. 111–112.

Shuai, Z., Yao, L., Xu, X., Wang, S., and Zhu, L. (2017). "Hybrid Collaborative Recommendation via Semi-Autoencoder," in International Conference on Neural Information Processing, 185–193.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Machine Learn. Res.* 15 (1), 1929–1958. doi:10.5555/2627435.2670313

Su, X., and Khoshgoftaar, T. M. (2009). A Survey of Collaborative Filtering Techniques. *Adv. Artif. Intell.* 2009, 1–19. doi:10.1155/2009/421425

Wang, H., Wang, N., and Yeung, D.-Y. (2015). "Collaborative Deep Learning for Recommender Systems," in Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, Sydney, NSW, Australia, August 10–15, 2015. 1235–1244. doi:10.1145/2783258.2783273

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., et al. (2020). Deep High-Resolution Representation Learning for Visual Recognition. *IEEE Trans. Pattern Anal. Mach Intell.* 43 (10), 3349–3364. doi:10.1109/TPAMI.2020.2983686

Wei, L., Zhou, C., Su, R., and Zou, Q. (2019). Pepred-Suite: Improved and Robust Prediction of Therapeutic Peptides Using Adaptive Feature Representation Learning. *Bioinformatics* 35 (21), 4272–4280. doi:10.1093/bioinformatics/btz246

Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.-I., and Jegelka, S. (2018). "Representation Learning on Graphs with Jumping Knowledge Networks," in the 35th International Conference on Machine Learning (ICML 2018), Stockholmsmässan, Stockholm, Sweden, July 10–15, 2018. 5453–5462.

Yang, Y., Zhu, Y., and Li, Y. (2021). Personalized Recommendation with Knowledge Graph via Dual-Autoencoder. *Appl. Intell.* 52, 1–12. doi:10.1007/s10489-021-02647-1

Yehuda, K., R, B., and C, V. (2009). Matrix Factorization Techniques for Recommender Systems. *Computer* 42 (8), 30–37. doi:10.1109/MC.2009.263

Yi, Z., Hu, X., Zhang, Y., and Li, P. (2018). Transfer Learning with Stacked Reconstruction Independent Component Analysis. *Knowl. Based Syst.* 152, 100–106. doi:10.1016/j.knosys.2018.04.010

Yu, Z., Lian, J., Mahmoody, A., Liu, G., and Xie, X. (2019). "Adaptive User Modeling with Long and Short-Term Preferences for Personalized Recommendation," in the 28th International Joint Conference on Artificial Intelligence (IJCAI 2019), Macao, China, August 10–16, 2019. 4213–4219. doi:10.24963/ijcai.2019/585

Zhang, Y., Ai, Q., Chen, X., and Croft, W. B. (2017). "Joint Representation Learning for Top-N Recommendation with Heterogeneous Information Sources," in the 2017 ACM on Conference on Information and Knowledge Management (CIKM 2017), Singapore, November 6–10, 2017. 1449–1458. doi:10.1145/3132847.3132892

Zhong, S.-T., Huang, L., Wang, C.-D., Lai, J.-H., and Yu, P. S. (2020). An Autoencoder Framework with Attention Mechanism for Cross-Domain Recommendation. *IEEE Trans. Cybern.* (11), 1–13. doi:10.1109/tcyb.2020.3029002

Zhu, Y., Wu, X., Qiang, J., Yuan, Y., and Li, Y. (2021). Representation Learning with Collaborative Autoencoder for Personalized Recommendation. *Expert Syst. Appl.* 186, 115825. doi:10.1016/j.eswa.2021.115825

Zhuang, F., Cheng, X., Luo, P., Pan, S. J., and He, Q. (2017a). Supervised Representation Learning with Double Encoding-Layer Autoencoder for Transfer Learning. *ACM Trans. Intell. Syst. Technol. (TIST)* 9 (2), 1–17. doi:10.1145/3108257

Zhuang, F., Luo, D., Yuan, N. J., Xie, X., and He, Q. (2017b). "Representation Learning with Pair-Wise Constraints for Collaborative Ranking," in the 10th ACM International Conference on Web Search and Data Mining (WSDM 2017), Cambridge, UK, February 6–10, 2017. 567–575. doi:10.1145/3018661.3018720

Zhuang, F., Zhang, Z., Qian, M., Shi, C., Xie, X., and He, Q. (2017c). Representation Learning via Dual-Autoencoder for Recommendation. *Neural Networks* 90, 83–89. doi:10.1016/j.neunet.2017.03.009

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The handling editor YD declared a past co-authorship with the author XS.

frontiers | Frontiers in Genetics

ORIGINAL RESEARCH
published: 22 July 2022
doi: 10.3389/fgene.2022.893409

Matching Biomedical Ontologies *via* a Hybrid Graph Attention Network

Peng Wang[1,2]* *and Yunyan Hu*[1]

[1]School of Computer Science and Engineering, Southeast University, Nanjing, China, [2]School of Cyber Science and Engineering, Southeast University, Nanjing, China

Biomedical ontologies have been used extensively to formally define and organize biomedical terminologies, and these ontologies are typically manually created by biomedical experts. With more biomedical ontologies being built independently, matching them to address the problem of heterogeneity and interoperability has become a critical challenge in many biomedical applications. Existing matching methods have mostly focused on capturing features of terminological, structural, and contextual semantics in ontologies. However, these feature engineering-based techniques are not only labor-intensive but also ignore the hidden semantic relations in ontologies. In this study, we propose an alternative biomedical ontology-matching framework BioHAN via a hybrid graph attention network, and that consists of three techniques. First, we propose an effective ontology-enriching method that refines and enriches the ontologies through axioms and external resources. Subsequently, we use hyperbolic graph attention layers to encode hierarchical concepts in a unified hyperbolic space. Finally, we aggregate the features of both the direct and distant neighbors with a graph attention network. Experimental results on real-world biomedical ontologies demonstrate that BioHAN is competitive with the state-of-the-art ontology matching methods.

Keywords: biomedical ontology, ontology matching, graph attention network, embedding, hyperbolic attention

OPEN ACCESS

Edited by:
Yucong Duan,
Hainan University, China

Reviewed by:
Jiang Bian,
University of Florida, United States
Xingsi Xue,
Fujian University of Technology, China

***Correspondence:**
Peng Wang
pwang@seu.edu.cn

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *10 March 2022*
Accepted: *20 June 2022*
Published: *22 July 2022*

Citation:
Wang P and Hu Y (2022) Matching Biomedical Ontologies via a Hybrid Graph Attention Network. Front. Genet. 13:893409. doi: 10.3389/fgene.2022.893409

1 INTRODUCTION

Ontology is an explicit, interoperable, extensible, scalable, and formal definition to describe knowledge as a set of domain vocabularies that contain concepts, relations between concepts, and individuals of concepts (Ramis et al., 2014). In past decades, various biomedical ontologies, such as the National Cancer Institute Thesaurus (NCI) (Golbeck et al., 2003), Foundation Model of Anatomy (FMA) (Rosse and Mejino, 2003), Systemized Nomenclature of Medicine (SNOMED-Clinical Terms [SNOMED-CT]) (Donnelly et al., 2006), and Uberon (Mungall et al., 2012) have been widely used for medical data format standardization (Cimino and Zhu, 2006), medical or clinical knowledge representation and integration (Isern et al., 2012), and medical decision making (De Potter et al., 2012) to provide standard semantics. With the continuous evolution of biomedical data, biomedical vocabularies have become complicated and ambiguous, which leads to challenges in developing biomedical applications. Moreover, new biomedical ontologies are constructed independently with diverse ways of defining overlapping biomedical terminologies or components, which also leads to more heterogeneity (Xie et al., 2016). As shown in **Figure 1**, the entities are connected *via* the *subClassOf* relation, and the equivalent concepts are linked via dotted lines. It can be found that for the same concept, "blood vessel" in the source and target ontologies, they are organized and interpreted at different levels of granularity, named conceptual heterogeneity. In addition, the concepts that share the same morphology "capillary" indicate

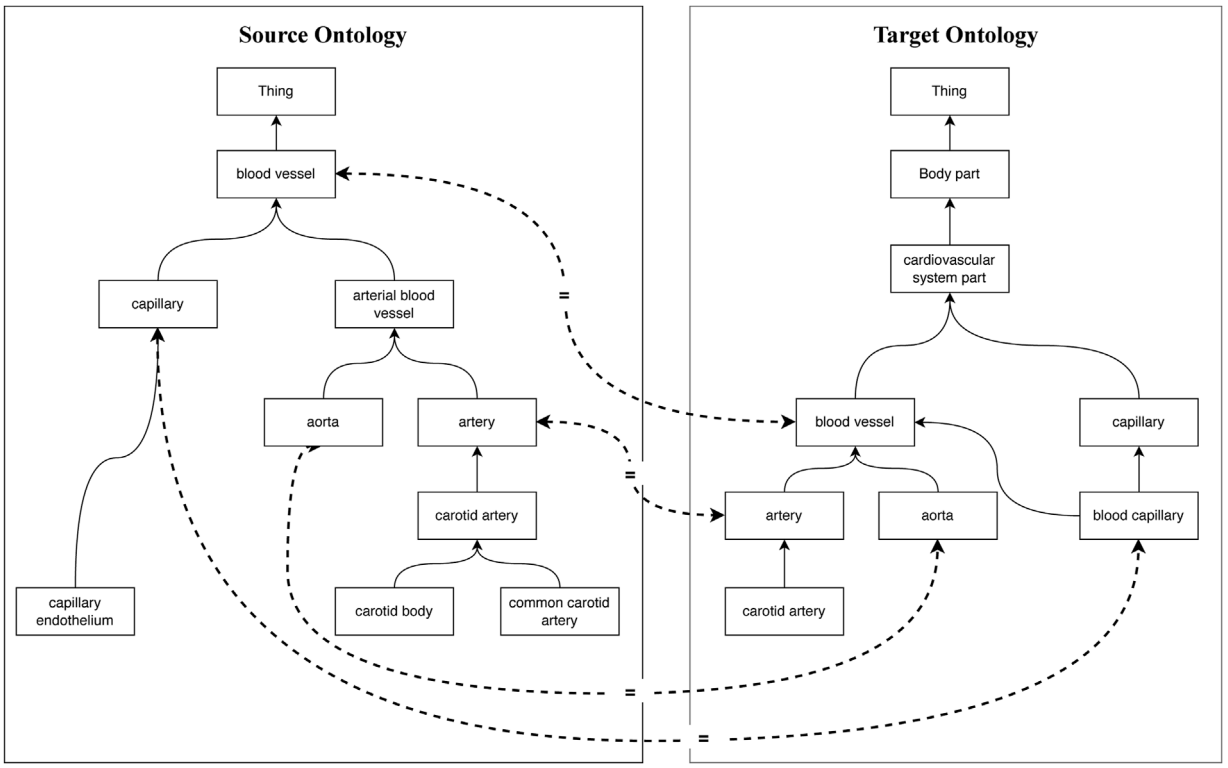


FIGURE 1 | Heterogeneity of biomedical ontologies.

different semantics in different ontologies, which is called semiotic heterogeneity. To implement interoperability across biomedical ontologies, discovering semantic relations between them is critically important (Xue, 2020). Ontology matching is a key technique to find semantic correspondences between the elements of different ontologies to achieve interoperability (Shvaiko and Euzenat, 2011).

Most existing ontology matching methods have focused on extracting features from terminological, structural, extensional (individuals of concepts) information, and external resources (Nezhadi et al., 2011; Otero-Cerdeira et al., 2015; Babalou et al., 2016; Chauhan et al., 2018). They use logical reasoning and rule-based techniques to extract sophisticated features, which are then used to compute the similarities of ontological elements (i.e., concepts, properties, and individuals) that promote ontology matching.

These feature-based methods (e.g., AML (Faria et al., 2013), FCA_Map (Zhao et al., 2018), LogMap (Jiménez-Ruiz and Cuenca Grau, 2011), and XMap (Djeddi and Khadir, 2014)) elaborate features of data to evaluate element similarity and derive semantic correspondences. However, the features in one ontology usually cannot be transferred to others. Consequently, the effectiveness and generality of those ontology matching methods vary significantly (Kolyvakis et al., 2018).

Recently, graph-based representation learning (Kipf and Welling, 2016; Hamilton et al., 2017) has become a powerful model for learning vector representations of graph-structured data. In graph neural networks (GNNs), the representation of a node is learned through recursively aggregating the representations of its local neighboring structure and propagation of features from neighboring nodes. Several studies (Chen et al., 2017; Wang Z et al., 2018; Wu et al., 2019; Sun et al., 2020) exploit GNNs for embedding-based matching in knowledge graphs (KGs), and have achieved promising results. However, existing GNN-based matching models still face some problems in ontology matching. First, ontology matching may face semantic imbalance because the distributions and amounts of semantic descriptions in different ontologies are generally different. We argue that if we can enrich the ontologies by using the metadata, given axioms, and auxiliary descriptions from external domain resources, and incorporate a rich set of semantic relationships, the derived ontologies can be matched with higher precision and recall. To overcome this problem, we consider designing an ontology-enriching method. Second, a distinguishable characteristic of biomedical ontologies, compared to open-domain knowledge bases such as YAGO (Suchanek et al., 2007), Wikidata (Vrandečić and Krötzsch, 2014), and DBpedia (Lehmann et al., 2015), is their domain specificity. These biomedical ontologies often have rich hierarchical structures that systematically organize biomedical concepts into categories and subcategories from general to specific. **Figure 2** shows an example of a hierarchical structure in different biomedical ontologies. The hierarchical structures of the corresponding

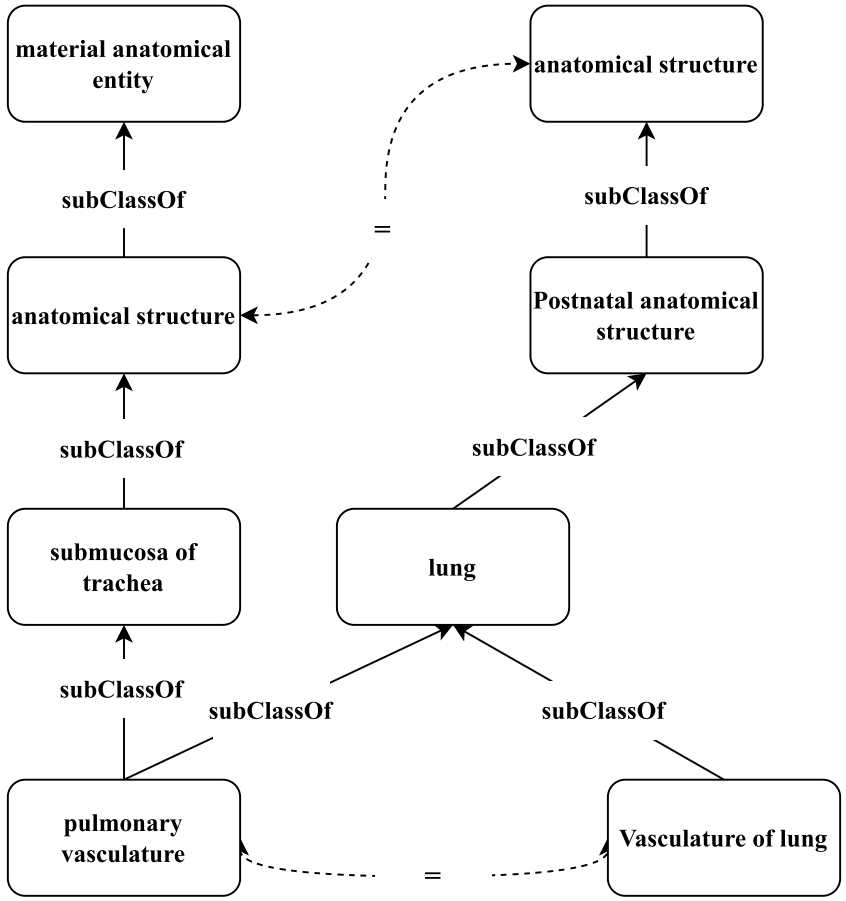


FIGURE 2 | Hierarchical structure in biomedical ontologies UBERON (left) and FMA (right).

pairs in different ontologies are similar to some extent. For example, the hierarchy (through *subClassOf* relation) of "pulmonary vasculature" in UBERON and "Vasculature of lung" in FMA is similar, whereas the terminologies are morphologically different. Therefore, capturing such hierarchical structures would be useful for identifying aligned concepts and improving the matching performance. Finally, since different ontologies usually have heterogeneous schemas and incompleteness (Schneider and Šimkus, 2020), the matching pairs usually have some dissimilar neighboring structures. Even though we assume that the ontologies to be matched are complete, because of the schema heterogeneity, the non-isomorphism in the neighboring structures from different ontologies is still inevitable. As shown in **Figure 2**, the one-hop neighbors of the matching pair ("pulmonary vasculature" and "Vasculature of lung") are different, while they share the same distant neighbor "anatomical structure." Motivated by the phenomenon that the relevant information could appear in both direct and distant neighbors of matching concepts, the aggregated structural semantics of a concept should include not only its local neighbors, but also the related distant neighbors. In addition, to keep the matching performance, we use an attention mechanism to realize the semantic relatedness of different neighbors, which could further discover and aggregate important neighbors.

To address these issues, we propose a biomedical ontology matching framework, BioHAN, with a hybrid graph attention network. The underlying idea is to first enrich and refine the ontologies to be matched with the given axioms and auxiliary semantic descriptions from external resources, such as UMLS (Bodenreider, 2004). Then, the neighborhood information is aggregated within multiple hops in the enriched ontologies, capturing both local and global features, into hyperbolic representations that are complementary to each other. Both representations are jointly optimized to improve ontology matching performance. The main contributions of this study are listed as follows:

- We propose a matching method BioHAN for biomedical ontologies. BioHAN first enriches the ontologies for matching via the axioms and logical rules. Then it further learns the representations with the hierarchical structure to realize ontology matching.
- We propose a lightweight and effective way to enrich and refine ontology with the metadata, axioms, and auxiliary semantic information from external resources, which is

helpful to discover and simplify the hidden and implicit semantics in ontologies.
- To capture the hierarchical features in an ontology, we leverage hyperbolic graph convolution layers to encode the parent and child concepts in the hyperbolic space.
- To further address the heterogeneity and better capture the semantics of concepts, we introduce an attention mechanism to weigh different neighbors and incorporate multi-hop neighbors to learn both the local and global hierarchical structures.
- We implement our proposed matching method and conduct systematic experiments on biomedical ontologies datasets. The evaluation of the Ontology Alignment Evaluation Initiative 2021 (OAEI 2021) shows that our method achieves significantly promising results.

The study is structured as follows. In **Section 2**, we describe relevant preliminaries of ontology matching and the overview of our proposed method. In **Section 3**, we illustrate the ontology-enriching operation, including ontology preprocessing and augmenting. In **Section 4**, the implementation details of our proposed matching method BioHAN are presented. **Section 5** describes our experiments, the results, and the experimental analysis and discussion. In **Section 6**, related work about ontology matching is systematically reviewed and introduced. **Section 7** summarizes our main findings, and presents perspectives on future work.

2 PRELIMINARIES AND METHOD OVERVIEW

2.1 Ontology Matching

Let $\mathcal{C}$ be the set of concepts, $\mathcal{R}$ be the set of relations, and $\mathcal{T} = \mathcal{C} \times \mathcal{R} \times \mathcal{C}$ be the set of triples or statements, then a biomedical ontology can be represented as $O = (\mathcal{C}, \mathcal{R}, \mathcal{T})$. The matching between two ontologies O_s and O_t is $\mathcal{M} = \{m_k | m_k = <e_i, e_j, r, s>\}$ (Euzenat and Shvaiko, 2007), where $\mathcal{M}$ is an alignment; m_k is a correspondence $<e_i, e_j, r, s>$; e_i and e_j are elements from O_s and O_t, respectively; r is the semantic relation between e_i and e_j; and $s \in [0, 1]$ is the confidence about a correspondence. Therefore, an alignment $\mathcal{M}$ is a set of correspondences m_k.

2.2 Graph Neural Networks

Graph neural networks (GNNs) are effective for various applications with graph-structured data (Zhou et al., 2020). A GNN framework usually has a graph encoder and a graph decoder, and its input is an adjacency matrix and features nodes and edges. The encoder uses the graph structure to propagate and aggregate information across nodes, and learns embeddings for local structure. A graph decoder is often used to compute similarity scores for all node pairs. Depending on the graph properties and aggregation strategies, some GNN frameworks have been proposed.

The vanilla GCN is a popular variant of the GNN (Kipf and Welling, 2016), in which the hidden representation of node i at the l-th ($l > 0$) layer $h_i^{(l)}$ is computed as

$$\boldsymbol{h}_i^{(l)} = \sigma\left(\sum_{j \in \mathcal{N}_i \cup \{i\}} \frac{1}{c_{ij}} \boldsymbol{W}_i^{(l)} \boldsymbol{h}_i^{(l-1)} \right) \tag{1}$$

where $\sigma(\cdot)$ is an activation function; $\boldsymbol{W}^{(l)}$ is the weight matrix of the l-th layer and c_{ij} is for normalization; and $\mathcal{N}_i$ denotes the neighbor set of node i. The vanilla GCN encodes node i as the mean pooling of the representations of its neighbors and node i itself from the last layer. The input vector fed to the first layer is denoted as $h_i^{(0)}$.

A graph attention network (GAT) (Veličković et al., 2018) is a novel convolution-style neural network with masked self-attention layers. In contrast to the GCN, it allows for implicitly setting different weights to nodes of the same neighboring node. Moreover, analyzing the learned attention weights could improve interpretability. Formally, the attention weight $\alpha_{ij}^{(l)} \in \mathcal{R}$ between i and j at the l-th layer is computed as follows:

$$\alpha_{ij}^{(l)} = \frac{exp\left(LeaklyReLU\left(\boldsymbol{a}^T \left[\boldsymbol{W}\boldsymbol{h}_i \| \boldsymbol{W}\boldsymbol{h}_j\right]\right)\right)}{\sum_{j \in \mathcal{N}_i} exp\left(LeaklyReLU\left(\boldsymbol{a}^T \left[\boldsymbol{W}\boldsymbol{h}_i \| \boldsymbol{W}\boldsymbol{h}_j\right]\right)\right)} \tag{2}$$

Here, $\cdot^T$ denotes transposition; $\boldsymbol{a}$ is an attention weight matrix; $\|$ is the concatenation operation; and *LeaklyReLU* is used to achieve nonlinear transformation.

2.3 Method Overview

As shown in **Figure 3**, our proposed BioHAN comprises two phases: ontology enriching and ontology matching. Given a biomedical ontology, the ontology enriching phase first preprocesses the ontology with the metadata and axioms, which complements the informative representations hidden in the ontology. It also explores matching seeds between the processed ontologies by supplementing some missing semantics through external resources. The ontology matching phase takes as input the derived ontology. Structures of ontologies are captured via graph attention networks for structural representation learning. Moreover, the lexical semantics of the concepts in ontologies are used, providing complementary clues for ontology matching.

3 ONTOLOGY ENRICHING

In this section, we will discuss ontology preprocessing and augmenting operation to enrich the initial ontology. Specifically, we first preprocess the ontology to discover the hidden semantics and represent them clearly. Then, we use ontology augmentation strategies to enrich the ontologies.

3.1 Ontology Preprocessing

We notice that there are two common facts in biomedical ontologies. On the one hand, some semantic information is hidden or unclear, which is expressed by complex axioms or ontology semantics. However, to further understand an ontology, such information is useful. On the other hand, some triples are used to describe the building and version

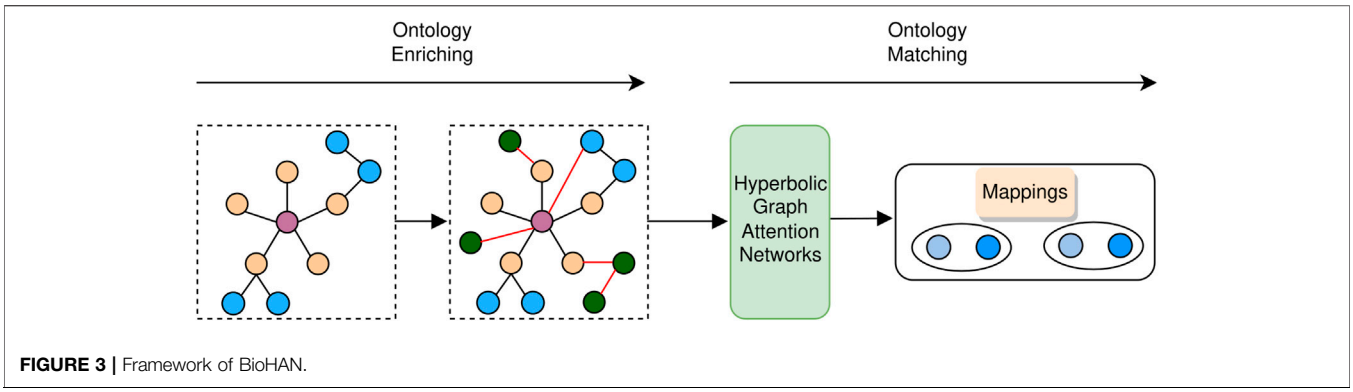


FIGURE 3 | Framework of BioHAN.

TABLE 1 | Example of ontology collection.

< rdf:Description rdf:about = "http://bioontology.org/ontologies/fma/physical state"
< rdfs:range >
< rdf:List >
< rdf:li > Gas < /rdf:li >
< rdf:li > Liquid < /rdf:li >
< rdf:li > Semi-solid < /rdf:li >
< rdf:li > Solid < /rdf:li >
< rdf:List >
< /rdfs:range >
< /rdf:Description >

information for an ontology. These statements simply increase the size of the ontology and are useless for the definitions of concepts and properties. Therefore, we conduct a preprocessing operation to refine ontologies. Specifically, we make the complex expressions of ontologies much simpler and clearer.

For the ontology language RDFS and OWL, they provide mechanisms for describing groups of related resources and the relationships between these resources, where OWL is an extension of RDFS, providing description logic-based primitives with richer expressive ability and stronger reasoning ability. In an ontology, containers (e.g., rdf:Bag, rdf:Seq, and rdf:Alt) and collections (e.g., rdf:List) are used to describe a set of resources in RDFS and OWL. They simplify the ontology expressions but hide some indirect semantics. We clearly define the semantics of the members in containers and collections, and then delete those redundant and complex statements. **Table 1** shows the range of property "physical state" through a collection *rdf:List* in RDFS format. Through the RDFS description, we can know that for the property "physical state" in the ontology "http://bioontology.org/ontologies/fma," its values could be one of "Gas," "Liquid," "Semi-solid," and "Solid." Each value is represented via *rdf:li*. However, the members would be represented as anonymous nodes while parsing the ontology, such as $< physicalstate, range, BN >$, $< BN, range, Liquid >$, where *BN* denotes an anonymous node with no specific meaning. These statements are difficult to understand directly. Therefore, it is necessary to formulate this implicit knowledge, such as $< physicalstate, range, Liquid >$.

In addition, to further mine the semantic descriptions in the biomedical ontologies, a rule-based reasoning method is proposed to discover the hidden information.

1) Enriching domain and range: given a property p_a, let p_b be the sub-property of p_a. Then we can infer that all semantics of the domain and range of p_a could be inherited by p_b. According to this rule, the semantics of sub-properties will be defined more comprehensively.
2) Enriching the concept axioms: given a concept axiom (e.g. owl: oneOf, owl:intersectionOf, owl: unionOf, owl:equivalentClass, etc.), its equivalent semantics could be rewritten by following rules. If a complex concept $A \sqcap B$ is defined by the axiom owl: intersectionOf, where the complex concept has a sub concept *C*, $A \sqsupset C$ and $B \sqsupset C$ could be added to the ontology. If one complex concept $A \sqcup B$ is defined by the axiom owl:unionOf, where the complex concept has a super concept *C*, so $C \sqsupset A$ and $C \sqsupset B$ could be added to the ontology. Similarly, we can also rewrite semantics of owl:oneOf and owl:equivalentClass. Therefore, complex semantics of concept axioms could be clearly defined.
3) Enriching the property axioms: given a property axiom (e.g. owl:SymmetricProperty, owl: TransitiveProperty, owl: equivalentProperty, etc.), relevant semantic extension could be realized by following rules. If a property *p* is declared by axiom owl:SymmetricProperty and there is a statement $< A, p, B >$, a new statement $< B, p, A >$ could be added to the ontology. If a property *p* is declared by axiom owl: TransitiveProperty and there are statements $< A, p, B >$ and $< B, p, C >$, then a new statement $< A, p, C >$ could be added to the ontology.
4) Enriching owl:sameAs axiom: given a statement $< A, owl: sameAs, B >$, then the equivalent individuals *A* and *B* could share their semantic information.
5) Enriching properties in the concept hierarchy: given $< p, rdfs: domain, A >$ and $< B, rdfs: subClassOf, A >$, we can infer an implicit statement $< p, rdfs: domain, B >$. According to this rule, the property's constraints about one concept could be extended to its sub-concepts.

3.2 Ontology Augmentation

Even though the derived ontologies have clearly specified the hidden semantics, they are still insufficient to some extent. Some semantic relationships are still missing, which may lead to the sparse problem of ontology structure. To alleviate this problem, we introduce several augmentation heuristics to enrich biomedical ontologies through the external domain resources, that is, UMLS.

3.2.1 Concept Augmentation

We first explore the anchors between the ontologies to be matched and the external resources, which is performed by using a simple string-based technique. Then, for one concept in ontologies, the relative semantics (e.g. rdfs:label, owl: annotation, owl:equivalentClass, etc.) of its anchored concept in external resources could be transferred and added to the ontology. Concept augmentation can significantly enrich ontologies with available information from external resources.

3.2.2 Neighborhood Augmentation

Relations between source and target concepts could also be derived from the anchored concepts in external resources. Specifically, if there is a relation between concepts i and j of the external resource, their anchors i' and j' are also linked by this relation. The goal is to reduce the semantic gap between ontologies by adding the missing structural information and solving the problem of sparse ontology graphs.

With the augmented ontologies, our matching framework enables sufficient learning of ontology representations. To match the concepts in ontology O_s and ontology O_t, we use graph pooling to obtain the embeddings of concepts. After investigating different graph pooling methods (Hamilton et al., 2017; Ying et al., 2018), we choose mean-pooling to capture information across concept neighbors. Finally, the graph neural networks take the enriched ontologies O_s and O_t as input to find the alignments.

4 MATCHING METHOD

In this section, we first embed the elements in ontologies to low dimension vectors, and then discuss the hyperbolic graph attention mechanism. Subsequently, we elaborate on the matching computation and the model training in detail.

4.1 Embedding

The terminological descriptions of concepts within a biomedical ontology are generally represented by a sequence of words. We leverage deep learning-based embedding methods (Peters et al., 2018; Devlin et al., 2019) to derive a fixed-size terminological description embedding for each concept. In this study, we choose BioBERT, a high-quality medical language model pre-trained on PubMed abstracts and clinical notes (Lee et al., 2020), to encode concepts. Considering the domain specificity of biomedical ontology, the embedding models toward a specific task can provide significant benefits (Alsentzer et al., 2019; Peng et al., 2019), and are much more appropriate than the general pre-training language model. The embeddings are used as the initial states $h^{0,E}$ of concepts, where E indicates the low-dimensional vectors in the Euclidean space.

4.2 Hyperbolic Graph Attention

Conventional GNNs typically capture the graph via message propagation to embed nodes into the Euclidean space. However, it could lead to the distortion of hierarchical structures (Nickel and Kiela, 2017). Hence, we transfer the node representations to a hyperbolic embedding space that can better capture the hierarchical characteristics of tree-like ontologies. In this study, we use a specific model, hyperbolic graph attention network (HGAT) (Zhang et al., 2021), which jointly implements both the expressiveness of a GAT and the superiority of hyperbolic geometry in capturing the hierarchical features. Moreover, multi-hop neighbors are also encoded into concepts, to comprehensively consider a broader context of concepts and alleviate the heterogeneity problem. The network architecture is shown in **Figure 4**.

4.2.1 Hyperbolic Feature Projection

The hyperbolic graph attention layer first establishes transformation between the tangent (Euclidean) and Poincaré ball, which is carried out by exponential and logarithmic maps. Specifically, we project the vector in a tangent space to a hyperbolic manifold through the exponential map, whereas the logarithmic map reverses the hyperbolic representation back to the Euclidean space. The initial hyperbolic embedding $\boldsymbol{h}_i^{0,H}$ of node i is

$$\boldsymbol{h}_i^{0,H} = exp_o^K\left(0, \boldsymbol{h}_i^{0,E}\right) \tag{3}$$

where K determines the constant negative curvature $-1/K(K>0)$ and the tangent space is centered at point o. To transform the hyperbolic features from one layer to the next layer, we follow the following computation:

$$\boldsymbol{h}_i^{l,H} = \left(\boldsymbol{W}^l \otimes^{K_{l-1}} h_i^{i-1,H}\right) \oplus^{K_{l-1}} \boldsymbol{b}^l \tag{4}$$

where $\otimes$ and $\oplus$ are hyperboloid matrix multiplication and addition, respectively.

4.2.2 Hyperbolic Attention Mechanism

To measure the importance of various neighbors and aggregate the neighbors' features to the center node according to their semantic weights, a self-attention mechanism is performed on the nodes. To that end, one parameterized weight matrix $\boldsymbol{W}$ is applied to all nodes to conduct the shared linear transformation. Then, the attention coefficient can be represented with a self-attention weight $\boldsymbol{a}$ on the nodes as follows:

$$e_{ij} = \boldsymbol{a}^T\left(\boldsymbol{W}\left(\boldsymbol{h}_i^h, \boldsymbol{h}_i^h\right)\right) \tag{5}$$

e_{ij} indicates the importance of node j to node i.

In addition, GAT considers only the local neighbors (i.e., one-hop neighbor nodes) for graph attention, while distant neighboring nodes can also contribute semantics to the central node. To reduce the effects of non-isomorphism in neighboring structures, we introduce distant neighboring information. Without loss of generality, we aggregate both the one-hop and two-hop neighboring information in ontologies, obtaining a proximity matrix.

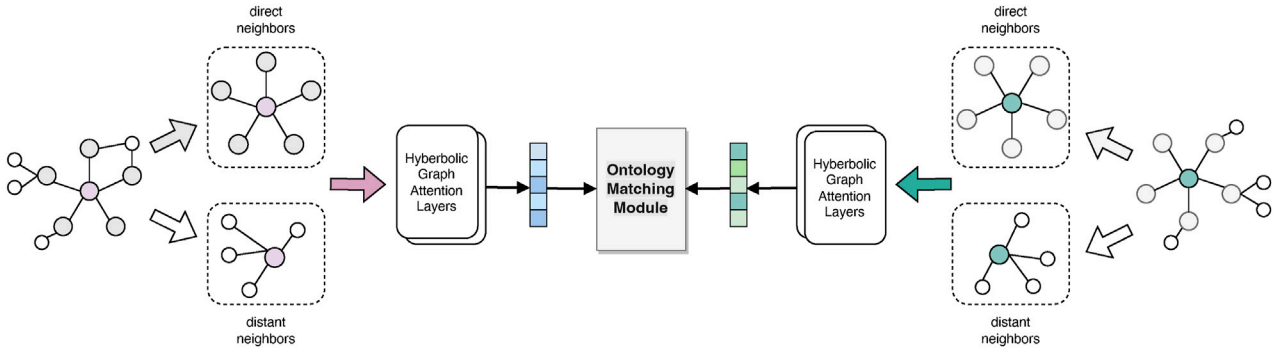


FIGURE 4 | Hyperbolic graph attention layers in BioHAN.

$$P = (B_1 + B_2)/2 \tag{6}$$

where B is the transition matrix and B_k denotes the adjacency matrix of k-th hop. $B_{ij} = 1/d_i$ if there exists an edge between i and j in the k-th hop, otherwise $B_{ij} = 0$. Then, P_{ij} denotes the topological weight that node j exerts on i.

To make coefficients comparable across different concepts, the attention weights are normalized via the softmax function.

$$\alpha_{ij} = softmax\left(e_{ij}\right) = \frac{exp\left(e_{ij}\right)}{\sum_{k\in\mathcal{N}_i} exp\left(e_{ij}\right)} \tag{7}$$

Finally, using the topological weights P and applying the LeakyReLU nonlinearity, the coefficients can be expressed as

$$\alpha_{ij}^{(l)} = \frac{exp\left(LeaklyReLU\left(P_{ij}\cdot \boldsymbol{a}^T\left[\overrightarrow{\boldsymbol{Wh}_i}\|\overrightarrow{\boldsymbol{Wh}_j}\right]\right)\right)}{\sum_{k\in\mathcal{N}_i} exp\left(LeaklyReLU\left(P_{ij}\cdot \boldsymbol{a}^T\left[\boldsymbol{Wh}_i\|\boldsymbol{Wh}_j\right]\right)\right)} \tag{8}$$

4.2.3 Hyperbolic Attention-Based Aggregation

Similar to GAT, the hyperbolic graph convolution layer aggregates features from a node's local neighbors. There is no notion of a vector space structure in a hyperbolic space, while the hyperboloidal aggregation requires multiplication by a weight matrix along with a bias operation. The main idea is to leverage the logarithmic projection to perform the Euclidean transformation and aggregation in the tangent space, and then transfer the obtained vectors back to the hyperbolic space. In addition, an attention mechanism is applied to learn the semantic relatedness between the neighboring nodes and the central node. Then, the neighbors' features are assembled in accordance with the learned attention coefficients. The hyperbolic attention-based aggregation is defined as follows:

$$AGG^K\left(\boldsymbol{h}^H\right)_i = exp\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij} log_{\boldsymbol{h}_i^H}^K\left(\boldsymbol{h}_j^{l-1,H}\right)\right) \tag{9}$$

To avoid semantic loss during the information propagation and maintain its transitivity between different convolutional layers, it is also necessary to incorporate the semantics of the central node itself.

$$AGG^K\left(\boldsymbol{h}^H\right)_i = exp\left(W_{AGG}\left(\boldsymbol{h}_i^H + \sum_{j\in\mathcal{N}_i} \alpha_{ij} log_{\boldsymbol{h}_i^H}^K\left(\boldsymbol{h}_j^{l-1,H}\right)\right)\right) \tag{10}$$

where W_{AGG} is the aggregated weight matrix, and $\boldsymbol{h}_i^H$ is the representation of the central node.

Finally, a non-linear activation function is used to increase the nonlinear expression ability and further improve the performance of the model. Specifically, BioHAN first applies Euclidean non-linear activation in the tangent space and then projects back to the hyperbolic space.

$$\sigma^{\otimes^{K_{l-1},K_l}}\left(\boldsymbol{h}^H\right) = exp_o^{K_l}\left(\sigma\left(log_{o^{K_{l-1}}}\left(\boldsymbol{h}^H\right)\right)\right) \tag{11}$$

The l-th layers of a hyperbolic graph attention layer are

$$\boldsymbol{h}_i^{l,H} = \sigma^{\otimes^{K_{l-1},K_l}}\left(AGG^{K_{l-1}}\left(\boldsymbol{h}^{l-1,H}\right)_i\right) \tag{12}$$

where $-1/K_{l-1}$ and $-1/K_l$ are the hyperbolic curvatures at the (l-1)-th and l-th layer, respectively. After iterative propagation and update of representations between layers, the final hyperbolic vector representations $\boldsymbol{h}^H$ can be obtained to represent the concepts.

4.3 Matching

Based on the learned concept representations $\boldsymbol{h}^H$ from the hyperbolic graph attention layers, our matching module takes as input pairs of concept embeddings from O_s and O_t, and then measures the semantic relatedness with a similarity metric function, defined as follows:

$$sim\left(c_i, c_j\right) = \left\{exp\left[\frac{1}{t}\left(d^K\left(\boldsymbol{h}_i^H, \boldsymbol{h}_j^H\right)^2 - r\right)\right] + 1\right\}^{-1} \tag{13}$$

where $d^K(\cdot,\cdot)$ is the hyperbolic distance, and r and t are hyper-parameters. Then we iteratively match the concepts of two different ontologies using the Stable Marriage algorithm (SM) (Gale and Shapley, 1962) over the concepts' pairwise similarities.

4.4 Training

To improve the matching performance of the proposed method, we jointly consider the reconstruction performance of the

hyperbolic graph attention network and the matching performance of the matching module.

For the hyperbolic graph attention network module, the graph transition matrix of the final output should be as close as possible to the original graph structure. Therefore, the graph reconstruction loss should be minimized.

$$\mathcal{L}^{\mathcal{H}} = \sum_{(i,j)\in E^{+}} p\left(c_i, c_j\right) + \sum_{(i,j)\in E^{-}} \omega\left[\mu - p\left(c_i, c_j\right)\right] \tag{14}$$

where E^{+} is the set of adjacency concept pairs; E^{-} represents the corresponding negative samples; μ is the margin value; ω is a trade-off factor; and $[\cdot]_{+} = \max(0, \cdot)$.

Then, for the matching module, we minimize the contrastive matching loss to actualize that the distances between pre-aligned concepts (positive) are as small as possible whereas the unmatched (negative) pairs have a relatively larger distance.

$$\mathcal{L}^{M} = -\sum_{(i,j)\in\mathcal{M}^{+}} logsim\left(c_i, c_j\right) - \sum_{(i,j)\in\mathcal{M}^{-}} log\left(1 - sim\left(c_i, c_j\right)\right) \tag{15}$$

where $\mathcal{M}^{+}$ is the set of seed correspondences between O_s and O_t, and $\mathcal{M}^{-}$ denotes the corresponding opposite ones.

The final joint loss function is defined as follows:

$$\mathcal{L} = \mathcal{L}^{\mathcal{H}} + \alpha \cdot \mathcal{L}^{M} \tag{16}$$

where α is positive hyper-parameters to control the trade-off among these loss components. The model is trained by minimizing the overall loss and optimized with an Adam (Kingma and Ba, 2014) optimizer.

5 EXPERIMENTS

This section reports the experimental results. To verify the effectiveness of BioHAN, we used Python to implement our approaches in Pytorch and conduct the experiments on a computer with an Intel Xeon 4110 CPU, Nvidia 2080Ti GPU, and 64-GB memory.

5.1 Datasets

The experiments are performed on the biomedical evaluation benchmark from the Ontology Alignment Evaluation Initiative 2021 (OAEI 2021), which organizes annual evaluation campaigns aiming at evaluating ontology matching technologies. Biomedical ontologies are collected from the Large Biomedical track in OAEI 2021, including the Foundational Model of Anatomy Ontology (FMA), SNOMED CT, and the National Cancer Institute Thesaurus (NCI).

The FMA is an ontology for biomedical informatics that symbolically represents the phenotypic structure of the human body (Rosse and Mejino, 2003). FMA has 78,988 concepts together with 78,985 *isA* triples.

The NCI provides reference terminologies for clinical care, translational and basis research, public information, and administrative activities (Golbeck et al., 2003). It comprises 66,724 concepts and 59,794 *isA* triples.

SNOMED CT is a systematically organized collection of medical terms and provides comprehensive, multilingual clinical healthcare terminology for clinical documentation and reporting (Donnelly et al., 2006). It contains 1,22,222 concepts and 1,05,624 *isA* triples.

The matching tasks are FMA-NCI, FMA-SNOMED, and NCI-SNOMED. On account of the primary hierarchical architecture of ontologies and the deficiency of some other relations, except the hierarchical structure, we mainly consider the *SubClassOf* relationship of these datasets. In this study, we only focus on identifying one-to-one equivalence correspondences between concepts. Seed alignments are extracted from the UMLS (Bodenreider, 2004) and trained as positive samples. The negative alignments are sampled by randomly modifying one of the concepts in the positive sample pairs.

5.2 Evaluation Measures

We follow the standard evaluation criteria in OAEI 2021, calculating the precision (P), recall (R), and F1-measure (F1) for each matching task. Given a reference alignment set *Ref* and mapping correspondences *Map*, the precision and recall are calculated as follows:

$$P = \frac{|Map \cap Ref|}{|Map|} \tag{17}$$

$$R = \frac{|Map \cap Ref|}{|Ref|} \tag{18}$$

The F1-measure is the weighted harmonic average of precision and recall, defined as

$$F1 = \frac{2 \times P \times R}{P + R} \tag{19}$$

5.3 Experimental Settings

For our proposed BioHAN, each training takes 1,000 epochs with the learning rates among {0.01, 0.001, 0.0001}. The embedding dimension d is set to 128, and the initial input embedding has the size (d) 512. By default, we stack two hyperbolic graph attention layers in our model. For the hyperbolic graph attention decoder, we set $r = 2.0$, $t = 1.0$, and apply trainable curvature, which refer to the parameter configuration in MEDTO (Hao et al., 2021). We set the trade-off hyper-parameters α_1 to 1.0. In addition, for each seed correspondence, we corrupt it and randomly replace it with five additional concepts to generate negative mapping pairs.

5.4 Experimental Results

5.4.1 Ontology Matching Results

Table 2 shows the matching results of our proposed model compared with several matching methods or systems based on feature engineering and representation learning. The feature engineering-based top-performing matching systems are

TABLE 2 | Results of ontology matching.

Method	FMA-NCI			FMA-SNOMED			SNOMED-NCI		
	P	R	F1	P	R	F1	P	R	F1
AML	**0.958**	0.910	**0.933**	0.923	0.762	**0.835**	0.906	0.746	0.818
LogMap	0.940	0.898	0.919	**0.941**	0.689	0.796	**0.954**	0.667	0.785
LogMapBio	0.904	0.920	0.912	0.911	0.711	0.799	0.909	0.696	0.88
MTransE	0.627	0.640	0.633	0.505	0.475	0.490	0.254	0.378	0.304
GCN-align	0.813	0.783	0.798	0.763	0.729	0.746	0.745	0.775	0.760
DAEOM	0.882	0.689	0.774	0.719	0.693	0.706	0.891	0.682	0.773
MEDTO	0.944	0.874	0.908	0.871	0.762	0.813	0.901	**0.802**	0.849
BioHAN	0.930	**0.922**	0.926	0.898	**0.775**	0.832	0.911	0.797	**0.850**
BioHAN (w/o OB)	0.930	0.922	0.926	0.782	0.731	0.756	0.788	0.709	0.746
BioHAN (w/o HB)	0.831	0.822	0.826	0.771	0.729	0.749	0.850	0.711	0.774
BioHAN (w/o AM)	0.860	0.842	0.851	0.819	0.726	0.770	0.864	0.719	0.785
BioHAN (w/o MN)	0.893	0.849	0.870	0.822	0.745	0.782	0.877	0.701	0.779

Bold values represents the best results for the column in which they are located.

selected according to the results published in the Large Biomedical track by OAEI 2021. The comparative representation learning models are several recent typical embedding-based entity alignment models (MTransE, GCN-Align) and ontology matching models (DAEOM, MEDTO).

Compared with the extensively developed feature-based approaches such as AML, LogMap, and LogMapBio, our method achieves competitive results across all three tasks. The proposed BioHAN outperforms these rule-based approaches in measure R in FMA-NCI and FMA-SNOMED. AML, LogMap, and LogMapBio heavily rely on lexical features extracted from ontologies, while using representation learning could better capture some hidden semantics to discover more complex matching pairs. We can also observe that entity alignment models (MTransE, GCN-Align) designed for general knowledge bases are insufficient for domain-specific ontology matching. Compared to the representative matching methods (DAEOM, MEDTO), BioHAN also achieves competitive performance. The performance difference between MEDTO and BioHAN validates the importance of hierarchical features. BioHAN explicitly distinguishes and models the hierarchical structure, taking into account both the local and global hierarchical features, and obviously leads to promising results in biomedical ontology matching.

5.4.2 Effectiveness of Ontology Enriching

To evaluate the effectiveness of the enriching phase, we further compare the *isA* triple size during ontology enriching. The detailed statistics concerning the size of each ontology matching task are shown in **Table 3**. Here, *Nodes* means the number of ontology entities, and *isA* is the edges between nodes with the relation *owl*: *subClassOf* in the ontology graph, while the *origin* and *enriching* represents the change in *isA* triple size before and after the enriching operation.

We can observe that the change in the triple size of both the ontology NCI and SNOMED is explicit, while the FMA remains. The structure of NCI and SNOMED is sophisticated, and contains substantive *owl:intersectionOf* and *owl:unionOf* property links, especially SNOMED. Specifically, the *owl:intersectionOf* statement describes classes which contain precisely those individuals that are members of the class extension of all class descriptions in the list, while the *owl:unionOf* statement describes an anonymous class containing those individuals occurring in at least one of the class extensions in the list.

TABLE 3 | Summary statistics of ontology enriching.

Ontology	Nodes	isA (origin)	isA (enriching)
FMA	78,988	78,985	78,985
NCI	66,724	59,794	75,454
SNOMED	1,22,222	1,05,624	2,03,942

Moreover, we compare the matching performance between the proposed BioHAN and its variation BioHAN (w/o OB), which does not pay attention to ontology preprocessing and enriching. Results are also shown in **Table 2**. It is obvious that our model BioHAN consistently outperforms across these tasks, with an average increase of 6.0% in the F1 measure. This is attributed to the critically abundant structural features and implicit semantics added to ontology, which indicates that hierarchical information and implicit semantic descriptions contain considerably representative and critical features for ontology matching.

5.5 Discussion

5.5.1 Impact of Ontology Enriching

According to the intuition that there are some hidden informative semantics in ontologies, especially for the complex one, we propose to enrich the ontology through ontology preprocessing and complementing. Through the statistics described in **Table 3**, numerous relationship descriptions are implicit but express a well-established role in ontology matching. Particularly in SNOMED, there are nearly more than twice the hierarchical relationships after enriching. Through the comparison of matching performance between BioHAN and

BioHAN (w/o OB) shown in **Table 2**, we can draw the conclusion that the enriching phase indeed contributes to ontology matching with the sufficient complements of semantic and structural information.

5.5.2 Performance Analysis of BioHAN

BioHAN uses the hyperbolic space projection to solve the intrinsical limitation in encoding complex patterns by its polynomial expanding capacity. In addition, it captures the structure of the concept by iteratively aggregating multi-hop neighborhoods with an attention mechanism. To gain an in-depth analysis of these components, we further design three variants of BioHAN: BioHAN (w/o HB), BioHAN (w/o AM), and BioHAN (w/o MN). BioHAN (w/o HB) replaces the hyperbolic projection with Euclidean space projection. BioHAN (w/o AM) removes the attention mechanism and regards all the neighboring nodes sharing the same weight. BioHAN (w/o MN) only considers the direct local neighbors and removes the multi-hop aggregation module in BioHAN. From the matching results reported in **Table 2**, we observe that the full model BioHAN achieves the best performance across all three matching tasks. It is also worth noting that both BioHAN (w/o AM) and BioHAN (w/o MN) perform better than BioHAN (w/o HB), which indicates that the hierarchical structure of the ontology captures much more essential and representative semantics. The hyperbolic graph convolutional layers can effectively encode such semantic information. By comparing the results of BioHAN (w/o AM) and BioHAN, it is obvious that the attention mechanism plays a significant role in solving the hierarchical heterogeneity of ontologies, which has improved the matching performance of 6.7% in F1 on average. For the multi-hop aggregation, by contrasting the performances of BioHAN (w/o MN) and BioHAN, it also exerts an important influence on capturing the semantics much more precisely than the complex hierarchical structures of biomedical ontologies. Multi-hop neighboring aggregation can discover much more complex matching pairs and has further improved the matching performance, especially in the measure R with an increase of 5.8% on average.

6 RELATED WORK

6.1 Biomedical Ontology Matching

Traditional feature-based approaches have been investigated for ontology matching, using terminological, structural, and semantic features for the discovery of semantically similar elements. LogMap (Jiménez-Ruiz and Cuenca Grau, 2011) uses lexical and structural indexes to enhance its scalability. AML (Faria et al., 2013) also uses various informative features and domain-specific thesauri to complete ontology matching. Feature-based matching systems mainly rely on hand-crafted features to achieve specific tasks. Unfortunately, these methods will be limited for a given scenario with weak informativeness. Representation learning has an important impact on ontology matching. OntoEmma (Wang L et al., 2018) proposes a novel neural architecture for biomedical ontology matching, feeding into amounts of definitions and contexts to encode the concepts. It derives a variety of labeled data for supervised training and augments entities with complementary descriptions from external biomedical thesauri to improve the quality of alignments. MultiOM (Li et al., 2019) models features in ontologies from multiple views: lexical, structural, and resource. Then, it optimizes the vectors by limiting the sampling scope via structural relations in ontologies. Wang et al. (2021) systematically analyze and verify the effectiveness of multi-dimensions matching clues, subsequently aggregating the representation learning clues to boost biomedical ontology matching.

6.2 Graph Representation Learning

Recently, graph representation learning has gained great attention as graph neural networks (GNNs) have achieved state-of-the-art performance in various fields, such as community detection (Gargi et al., 2011), link prediction (Liben-Nowell and Kleinberg, 2007), graph alignment (Sun et al., 2018), and node classification (Bhagat et al., 2011). Some studies (Chen et al., 2017; Wang L et al., 2018) have used GNNs to achieve graph-embedded entity alignment, as similar entities often have similar neighborhoods in knowledge graphs (KG). Considering the attention mechanism, a graph attention network (Veličković et al., 2018) is proposed to learn the relatedness and importance propagated from the neighbors to the centered node. Then the neighboring message is incorporated with the measured weights. DAEOM (Wu et al., 2020) develops Siamese graph attention mechanism-based autoencoders to effectively integrate both the network structure and terminological description for deep latent representation learning in ontology matching. Recently, some researchers have substantiated that data in the form of graphs exhibit a non-Euclidean latent anatomy (Wilson et al., 2014; Bronstein et al., 2017). In addition, some recent works (Bronstein et al., 2017; Nickel and Kiela, 2017) have demonstrated the distinguished representation ability of hyperbolic manifold to model datasets with hierarchical layouts, as the hyperbolic geometry performs well in reflecting hierarchical representations naturally. Inspired by this insight, numerous research studies focus on investigating the hyperbolic geometric graph models, such as those by Nickel and Kiela (2017); Nickel and Kiela (2018); Ganea et al. (2018); and Hao et al. (2021). MEDTO (Hao et al., 2021) encodes the hierarchical features of concepts through hyperbolic graph convolution layers and further captures both local and global structural information of concepts via heterogeneous graph layers to learn better concept representations for ontology matching, and has achieved remarkably competitive performance.

7 CONCLUSION

In this study, we propose BioHAN for biomedical ontology matching, a hybrid graph neural network-based auto encoder to effectively integrate hierarchical structures for latent representation learning in biomedical ontology matching.

The proposed framework BioHAN executes ontology enriching to refine and complement the semantic information and hierarchical structures. Then it encodes the geometrical properties of concepts into a hyperbolic space to capture the hierarchical information through hyperbolic graph attention layers. We further implement multi-hop neighboring aggregation to incorporate both the local and global hierarchical structures with an attention mechanism to learn better concept representations for ontology matching. Our experiments conducted on a variety of biomedical ontologies demonstrate the effectiveness of BioHAN. Nonetheless, our approach only considers the *subClassOf* relationship in the ontology, which would restrict the capability of graph representation learning. In the future, it is promising to investigate some other types of non-isomorphism relations and incorporate the heterogeneous features into biomedical ontology matching. In addition, as for the large-scale biomedical ontology, the matching efficiency would also be taken into account in future research.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/Supplementary Material; further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

PW and YH outlined and wrote the overall manuscript. YH conducted the experiments. All authors actively participated in editing of the manuscript.

FUNDING

The work is supported by the National Key R&D Program of China (2018YFD1100302) and the All-Army Common Information System Equipment Pre-Research Project (Nos. 31514020501 and 31514020503).

REFERENCES

Alsentzer, E., Murphy, J. R., Boag, W., Weng, W.-H., Jin, D., Naumann, T., et al. (2019). "Publicly Available Clinical BERT Embeddings," in Proceedings of the 2nd Clinical Natural Language Processing Workshop, June 2019 (Minneapolis, MN: Association for Computational Linguistics), 72–78.

Babalou, S., Kargar, M. J., and Davarpanah, S. H. (2016). "Large-scale Ontology Matching: A Review of the Literature," in 2016 Second International Conference on Web Research (ICWR), Tehran, Iran, April 27–28, 2016, 158–165. doi:10.1109/ICWR.2016.7498461

Bhagat, S., Cormode, G., and Muthukrishnan, S. (2011). Node Classification in Social Networks. *Soc. Netw. Data Anal.* 5, 115–148. doi:10.1007/978-1-4419-8462-3_5

Bodenreider, O. (2004). The Unified Medical Language System (Umls): Integrating Biomedical Terminology. *Nucleic acids Res.* 32, 267D–270D. doi:10.1093/nar/gkh061

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. (2017). Geometric Deep Learning: Going beyond Euclidean Data. *IEEE Signal Process. Mag.* 34, 18–42. doi:10.1109/msp.2017.2693418

Chauhan, A., Vijayakumar, V., and Sliman, L. (2018). Ontology Matching Techniques: A Gold Standard Model. *arXiv*. [Preprint]. Available at: https://doi.org/10.48550/arXiv.1811.10191.

Chen, M., Tian, Y., Yang, M., and Zaniolo, C. (2017). "Multilingual Knowledge Graph Embeddings for Cross-Lingual Knowledge Alignment," in IJCAI'17: Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, Australia, August 19–25, 2017 (AAAI Press), 1511–1517. doi:10.24963/ijcai.2017/209

Cimino, J. J., and Zhu, X. (2006). The Practical Impact of Ontologies on Biomedical Informatics. *Yearb. Med. Inf.* 15, 124–135. doi:10.1055/s-0038-1638470

De Potter, P., Cools, H., Depraetere, K., Mels, G., Debevere, P., De Roo, J., et al. (2012). Semantic Patient Information Aggregation and Medicinal Decision Support. *Comput. methods programs Biomed.* 108, 724–735. doi:10.1016/j.cmpb.2012.04.002

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding," in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (Minneapolis, MN: Association for Computational Linguistics), 4171–4186.

Djeddi, W. E., and Khadir, M. T. (2014). "A Novel Approach Using Context-Based Measure for Matching Large Scale Ontologies," in International Conference on Data Warehousing and Knowledge Discovery, Munich, Germany, September 1–5, 2014, 320–331. doi:10.1007/978-3-319-10160-6_29

Donnelly, K. (2006). Snomed-ct: The Advanced Terminology and Coding System for Ehealth. *Stud. Health Technol. Inf.* 121, 279–290.

Euzenat, J., and Shvaiko, P. (2007). *Ontology Matching*. Heidelberg, Germany: Springer. doi:10.1007/978-3-540-49612-0

Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I. F., and Couto, F. M. (2013). "The Agreementmakerlight Ontology Matching System," in OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", Graz, Austria, September 13–9, 2013, 527–541. doi:10.1007/978-3-642-41030-7_38

Gale, D., and Shapley, L. S. (1962). College Admissions and the Stability of Marriage. *Am. Math. Mon.* 69, 9–15. doi:10.1080/00029890.1962.11989827

Ganea, O., Bécigneul, G., and Hofmann, T. (2018). "Hyperbolic Entailment Cones for Learning Hierarchical Embeddings," in International Conference on Machine Learning, Stockholm, Sweden, July 10–15, 2018 (PMLR), 1646–1655.

Gargi, U., Lu, W., Mirrokni, V., and Yoon, S. (2011). "Large-scale Community Detection on Youtube for Topic Discovery and Exploration," in Fifth International AAAI Conference on Weblogs and Social Media, Barcelona, Spain, July 17–21, 2011, 486–489.

Golbeck, J., Fragoso, G., Hartel, F., Hendler, J., Oberthaler, J., and Parsia, B. (2003). The National Cancer Institute's Thesaurus and Ontology. *SSRN J.* 1 (1), 75–80. doi:10.2139/ssrn.3199007

Hamilton, W. L., Ying, R., and Leskovec, J. (2017). "Inductive Representation Learning on Large Graphs," in NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems (Red Hook, NY: Curran Associates Inc), 1025–1035.

Hao, J., Lei, C., Efthymiou, V., Quamar, A., Özcan, F., Sun, Y., et al. (2021). "Medto: Medical Data to Ontology Matching Using Hybrid Graph Neural Networks," in Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Washington, DC, August 14–18, 2021 (Association for Computing Machinery), 2946–2954. doi:10.1145/3447548.3467138

Isern, D., Sánchez, D., and Moreno, A. (2012). Ontology-driven Execution of Clinical Guidelines. *Comput. methods programs Biomed.* 107, 122–139. doi:10.1016/j.cmpb.2011.06.006

Jiménez-Ruiz, E., and Cuenca Grau, B. (2011). "Logmap: Logic-Based and Scalable Ontology Matching," in International Semantic Web Conference, Bonn, Germany, October 23–27, 2011, 273–288. doi:10.1007/978-3-642-25073-6_18

Kingma, D. P., and Ba, J. (2014). Adam: A Method for Stochastic Optimization. *arXiv*. [Preprint]. Available at: https://arxiv.org/abs/1412.6980

Kipf, T. N., and Welling, M. (2016). "Semi-supervised Classification with Graph Convolutional Networks," in International Conference on Learning Representations (ICLR 2017) (Toulon, France: OpenReview.net).

Kolyvakis, P., Kalousis, A., and Kiritsis, D. (2018). "Deepalignment: Unsupervised Ontology Matching with Refined Word Vectors," in Proceedings of NAACL, New Orleans, LA, June 1–6, 2018, 787–798. doi:10.18653/v1/N18-1072

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., et al. (2020). Biobert: a Pre-trained Biomedical Language Representation Model for Biomedical Text Mining. *Bioinformatics* 36, 1234–1240. doi:10.1093/bioinformatics/btz682

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., et al. (2015). DBpedia - A Large-Scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic web* 6, 167–195. doi:10.3233/sw-140134

Li, W., Duan, X., Wang, M., Zhang, X., and Qi, G. (2019). Multi-view Embedding for Biomedical Ontology Matching. *OM@ISWC* 2536, 13–24.

Liben-Nowell, D., and Kleinberg, J. (2007). The Link-Prediction Problem for Social Networks. *J. Am. Soc. Inf. Sci.* 58, 1019–1031. doi:10.1002/asi.20591

Mungall, C. J., Torniai, C., Gkoutos, G. V., Lewis, S. E., and Haendel, M. A. (2012). Uberon, an Integrative Multi-Species Anatomy Ontology. *Genome Biol.* 13, R5–R20. doi:10.1186/gb-2012-13-1-r5

Nejhadi, A. H., Shadgar, B., and Osareh, A. (2011). Ontology Alignment Using Machine Learning Techniques. *AIRCC's Int. J. Comput. Sci. Inf. Technol.* 3, 139–150. doi:10.5121/ijcsit.2011.3210

Nickel, M., and Kiela, D. (2017). "Poincaré Embeddings for Learning Hierarchical Representations," in *Advances in Neural Information Processing Systems* (Curran Associates, Inc.) Vol. 30.

Nickel, M., and Kiela, D. (2018). "Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic Geometry," in International Conference on Machine Learning, Stockholm, Sweden, June 10–15, 2018 (PMLR), 3779–3788.

Otero-Cerdeira, L., Rodríguez-Martínez, F. J., and Gómez-Rodríguez, A. (2015). Ontology Matching: A Literature Review. *Expert Syst. Appl.* 42, 949–971. doi:10.1016/j.eswa.2014.08.032

Peng, Y., Yan, S., and Lu, Z. (2019). "Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets," in Proceedings of the 18th BioNLP Workshop and Shared Task (Florence, Italy: Association for Computational Linguistics), 58–65.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., et al. (2018). *Deep Contextualized Word Representations*. New Orleans, LA: NAACL, 2227–2237. doi:10.18653/v1/N18-1202

Ramis, B., Gonzalez, L., Iarovyi, S., Lobov, A., Martinez Lastra, J. L., Vyatkin, V., et al. (2014). "Knowledge-based Web Service Integration for Industrial Automation," in 2014 12th IEEE International Conference on Industrial Informatics, Porto Alegre RS, Brazil, July 27–30, 2014, 733–739. doi:10.1109/INDIN.2014.6945604

Rosse, C., and Mejino, J. L. V., Jr (2003). A Reference Ontology for Biomedical Informatics: the Foundational Model of Anatomy. *J. Biomed. Inf.* 36, 478–500. doi:10.1016/j.jbi.2003.11.007

Schneider, T., and Šimkus, M. (2020). Ontologies and Data Management: a Brief Survey. *Künstl Intell.* 34, 329–353. doi:10.1007/s13218-020-00686-3

Shvaiko, P., and Euzenat, J. (2013). Ontology Matching: State of the Art and Future Challenges. *IEEE Trans. Knowl. Data Eng.* 25, 158–176. doi:10.1109/tkde.2011.253

Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). "Yago: a Core of Semantic Knowledge," in Proceedings of the 16th international conference on World Wide Web, Banff, AB, Canada, May 8–12, 2007, 697–706. doi:10.1145/1242572.1242667

Sun, Z., Hu, W., Zhang, Q., and Qu, Y. (2018). Bootstrapping Entity Alignment with Knowledge Graph Embedding. *Proc. Twenty-Seventh Int. Jt. Conf. Artif. Intell.* 18, 4396–4402. doi:10.24963/ijcai.2018/611

Sun, Z., Wang, C., Hu, W., Chen, M., Dai, J., Zhang, W., et al. (2020). Knowledge Graph Alignment Network with Gated Multi-Hop Neighborhood Aggregation. *Proc. AAAI Conf. Artif. Intell.* 34, 222–229. doi:10.1609/aaai.v34i01.5354

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2018). "Graph Attention Networks,' inInternational Conference on Learning Representations (Vancouver, BC, Canada: OpenReview.net).

Vrandečić, D., and Krötzsch, M. (2014). Wikidata: a Free Collaborative Knowledgebase. *Commun. ACM* 57, 78–85. doi:10.1145/2629489

Wang, P., Hu, Y., Bai, S., and Zou, S. (2021). Matching Biomedical Ontologies: Construction of Matching Clues and Systematic Evaluation of Different Combinations of Matchers. *JMIR Med. Inf.* 9, e28212. doi:10.2196/28212

Wang, L., Bhagavatula, C., Neumann, M., Lo, K., Wilhelm, C., and Ammar, W. (2018). "Ontology Alignment in the Biomedical Domain Using Entity Definitions and Context," in Proceedings of the BioNLP 2018 Workshop (Mwlbourne, Australia: Association for Computational Linguistics), 47–55.

Wang, Z., Lv, Q., Lan, X., and Zhang, Y. (2018). "Cross-lingual Knowledge Graph Alignment via Graph Convolutional Networks," in Proceedings of the 2018 conference on empirical methods in natural language processing, Brussels, Belgium, October 31–November 4, 2018, 349–357. doi:10.18653/v1/d18-1032

Wilson, R. C., Hancock, E. R., Pekalska, E., and Duin, R. P. W. (2014). Spherical and Hyperbolic Embeddings of Data. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 2255–2269. doi:10.1109/tpami.2014.2316836

Wu, Y., Liu, X., Feng, Y., Wang, Z., Yan, R., and Zhao, D. (2019). "Relation-aware Entity Alignment for Heterogeneous Knowledge Graphs," in Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, August 10–16, 2019, 5278–5284. doi:10.24963/ijcai.2019/733

Wu, J., Lv, J., Guo, H., and Ma, S. (2020). Daeom: A Deep Attentional Embedding Approach for Biomedical Ontology Matching. *Appl. Sci.* 10, 7909. doi:10.3390/app10217909

Xie, C., Chekol, M. W., Spahiu, B., and Cai, H. (2016). "Leveraging Structural Information in Ontology Matching," in 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), Crans-Montana, Switzerland, March 23–25, 2016, 1108–1115. doi:10.1109/aina.2016.64

Xue, X. (2020). A Compact Firefly Algorithm for Matching Biomedical Ontologies. *Knowl. Inf. Syst.* 62, 2855–2871. doi:10.1007/s10115-020-01443-6

Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., and Leskovec, J. (2018). Hierarchical Graph Representation Learning with Differentiable Pooling. *Adv. neural Inf. Process. Syst.* 31, 4805–4815. doi:10.1145/3469877.3495645

Zhang, Y., Wang, X., Shi, C., Jiang, X., and Ye, Y. F. (2021). Hyperbolic Graph Attention Network. *IEEE Trans. Big Data* 8, 1. doi:10.1109/tbdata.2021.3081431

Zhao, M., Zhang, S., Li, W., and Chen, G. (2018). Matching Biomedical Ontologies Based on Formal Concept Analysis. *J. Biomed. Semant.* 9, 1–27. doi:10.1186/s13326-018-0178-9

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., et al. (2020). Graph Neural Networks: A Review of Methods and Applications. *AI Open* 1, 57–81. doi:10.1016/j.aiopen.2021.01.001

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers | Frontiers in Genetics

ORIGINAL RESEARCH
published: 22 July 2022
doi: 10.3389/fgene.2022.900242

MLEE: A method for extracting object-level medical knowledge graph entities from Chinese clinical records

Genghong Zhao[1,2], Wenjian Gu[3], Wei Cai[2], Zhiying Zhao[4], Xia Zhang[1,2]* and Jiren Liu[1,5]**

[1]School of Computer Science and Engineering Northeastern University, Shenyang, China, [2]Neusoft Research of Intelligent Healthcare Technology, Shenyang, China, [3]School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China, [4]Department of Clinical Epidemiology, Shengjing Hospital of China Medical University, Shenyang, China, [5]Neusoft Corporation, Shenyang, China

As a typical knowledge-intensive industry, the medical field uses knowledge graph technology to construct causal inference calculations, such as "symptom-disease", "laboratory examination/imaging examination-disease", and "disease-treatment method". The continuous expansion of large electronic clinical records provides an opportunity to learn medical knowledge by machine learning. In this process, how to extract entities with a medical logic structure and how to make entity extraction more consistent with the logic of the text content in electronic clinical records are two issues that have become key in building a high-quality, medical knowledge graph. In this work, we describe a method for extracting medical entities using real Chinese clinical electronic clinical records. We define a computational architecture named MLEE to extract object-level entities with "object-attribute" dependencies. We conducted experiments based on randomly selected electronic clinical records of 1,000 patients from Shengjing Hospital of China Medical University to verify the effectiveness of the method.

Keywords: knowledge graph (KG), medical entity extraction, natural language processing (computer science), EMR data mining, Chinese clinical records

OPEN ACCESS

Edited by:
Yucong Duan,
Hainan University, China

Reviewed by:
Pu-Feng Du,
Tianjin University, China
Mona Alshahrani,
King Abdullah University of Science and Technology, Saudi Arabia

****Correspondence:***
Genghong Zhao
1810626@stu.neu.edu.cn
Xia Zhang
zhangx@neusoft.com
Jiren Liu
liujr@neusoft.com

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *20 March 2022*
Accepted: *16 June 2022*
Published: *22 July 2022*

Citation:
Zhao G, Gu W, Cai W, Zhao Z, Zhang X and Liu J (2022) MLEE: A method for extracting object-level medical knowledge graph entities from Chinese clinical records. Front. Genet. 13:900242. doi: 10.3389/fgene.2022.900242

1 INTRODUCTION

Since Google proposed the concept of a knowledge graph in 2012, it has become one of the hottest technologies in knowledge reasoning. An increasing number of researchers use the "entity-relationship" approach to express the real world (Zheng et al., 2021). This kind of knowledge representation has achieved perfect results in a search engine, question and answer (Q&A) format, etc. Various vertical fields are building more innovative application scenarios based on knowledge graphs. As a typical knowledge-intensive industry, healthcare is a popular vertical field that utilizes knowledge graph technology (Shi et al., 2017).

The shortage of global medical resources caused by Coronavirus Disease 2019 (COVID-19) has become a global disaster. Improving the medical efficiency of healthcare has become an urgent problem that needs to be solved by researchers worldwide (Zhu et al., 2017). Historically, many researchers have attempted to help doctors build a medical base and improve clinical efficiency (Jonnagaddala et al., 2015; Li et al., 2020b). Knowledge graph technology is currently a popular research direction in this field.

In medical knowledge graph technology, the first and most crucial step is to build a high-quality medical knowledge graph. In this step, researchers need to discuss the main issues from two perspectives: the data source for constructing the medical knowledge graph and the algorithm for extracting entities and relationships.

Data sources are divided into two types: data sources that use authoritative knowledge bases and data sources that use clinical record data. Building a knowledge graph based on a traditional knowledge base can usually ensure the accuracy of the data source because its knowledge is neatly organized. Although building a knowledge graph using such data is easy, due to the large individual differences among patients in the real world, the basis for judgment in clinical diagnosis is relatively complex. Enumeration in authoritative knowledge bases is challenging (Abhyuday et al., 2020). In addition, the lag in the update of such knowledge bases is problematic for inference calculations such as clinical decision support systems (CDSS). With the development of medical informatization in recent years, an increasing number of electronic medical records (EMRs), laboratory information systems (LISs), and PACKS have been established, providing a massive data foundation for the use of clinical data analysis, modeling, and information extraction. When using clinical records to build a knowledge graph, all patient data are entered and updated in real time, ensuring the validity and diversity of real-world data (Mykowiecka et al., 2009). However, the use of clinical records to build a knowledge graph has difficulties. When doctors write clinical records in natural language, the complexity of the patient's condition is difficult for machines to understand (Louise et al., 2010).

In the process of using algorithms to construct a medical knowledge graph, in addition to using crawler technology to obtain data from a medical knowledge base with a relatively regular presentation structure (Li et al., 2020a), another technical route mainly uses deep learning to achieve both entity extraction and entity-relationship extraction. Relation extraction is a classification calculation in most research processes, and deep learning can usually achieve very high accuracy. However, challenges still exist when extracting and calculating medical entity recognition. First, when doctors write clinical records, they are not recorded for analysis by algorithms. The content of the records is usually complicated by the complexity of the patient's condition, which is a challenge for both feature conversion and information extraction (Kang et al., 2017). Second, the medical information cannot precisely express medical entities through simple strings due to its particularity. For example, for the "fever" entity, multiple factors, such as the cause, occurrence time, duration, body temperature, and peak heat of the patient's fever, need to be shown. When describing a patient's fever, clinicians may even use only a description of the above information without using the word "fever".

The main contributions of this study are presented as follows:

By analyzing the relationship between clinical records and medical knowledge graphs, a set of methods for extracting medical entities from clinical data and constructing knowledge graphs is explored.

Through "punctuation correction", the problem of entity recognition boundary errors caused by irregular medical records written by doctors is perfectly solved so that medical entities appearing in medical records can be stored in a complete semantic expression, avoiding information loss caused by the source.

Through clinical practice and data experiments, the hidden category attributes of sentences in medical records are verified, minimizing the semantic space of each category of medical entities during extraction, thereby improving the accuracy of entity recognition.

Last, two layers of basic sequence annotation calculation are used to extract medical entity fragments and entity attributes from the text to complete the extraction of medical entities from clinical medical records.

The clinical records are parsed by simulating how clinicians write records, and then medical entities are extracted.

The medical entity extracted by this method is a solid entity with "object attributes". Such entities can be directly utilized to construct medical knowledge graphs and can serve as input data for knowledge graph reasoning calculations. By increasing the diversity of information within entities, reasoning accuracy using knowledge graphs is improved.

The remainder of this paper is organized as follows: The second chapter introduces the current methods from researchers to construct medical knowledge graphs and to extract medical entities from various types of data. The third chapter introduces the detailed process of extracting medical entities from clinical record data in this study. The fourth chapter introduces the experimental results of this method using actual clinical data to extract medical entities. The fifth chapter introduces the conclusions of this research and prospects for future work. The source code is available at https://github.com/cocojoe0220/MLEE.

2 RELATED WORK

Research on building knowledge graphs has become very popular in recent years—researchers complete entity recognition and entity-relationship recognition by constructing novel computational architectures (Uzuner et al., 2010; Weng et al., 2017; Zhao et al., 2017; Cheng et al., 2019; Qiu et al., 2019; Wu et al., 2021). Related research on medical data to build knowledge graphs is continually emerging. These studies focus on building knowledge graphs based on clinical medical record data and building knowledge graphs based on public medical health datasets (Jiang et al., 2017; Jiang et al., 2021).

Liu and Xu (2021) attempt to build a knowledge graph from real-world, "dirty" electronic medical records. In this study, after extracting "symptom-disease"-related data from clinical medical records, the medical record text itself is used to complete disambiguation based on similarity calculation and to construct a knowledge graph related to symptoms and diseases. The disease prediction calculation based on patient symptoms is completed based on the knowledge graph. Weng et al. (2017) (Weng et al., 2017)used traditional Chinese medicine (TCM) unstructured clinical text data, clinical protocol guidelines, medical textbooks, and other data to construct a TCM clinical knowledge graph based on the triad structure. This research describes an entity through the Resource Description Framework (RDF) and combines the relationship between TCM and human body parts to construct an entity with

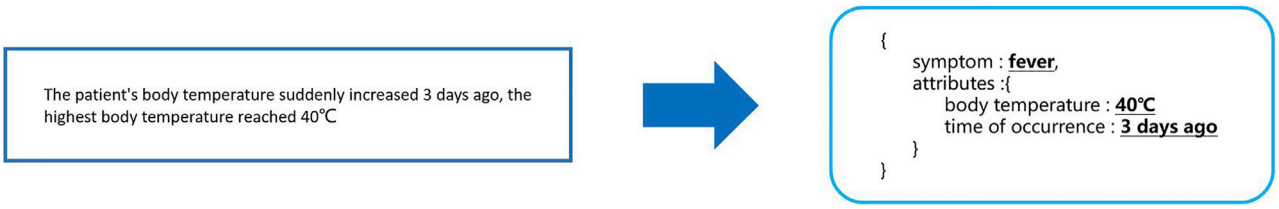


FIGURE 1 | Extract fever-related attributes from the fever description segment.

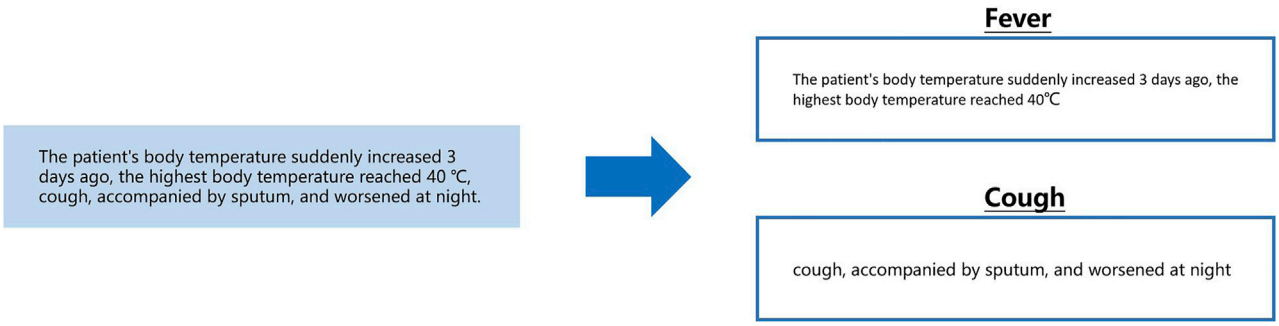


FIGURE 2 | From a sentence describing symptoms, separately extract segments describing fever and cough.

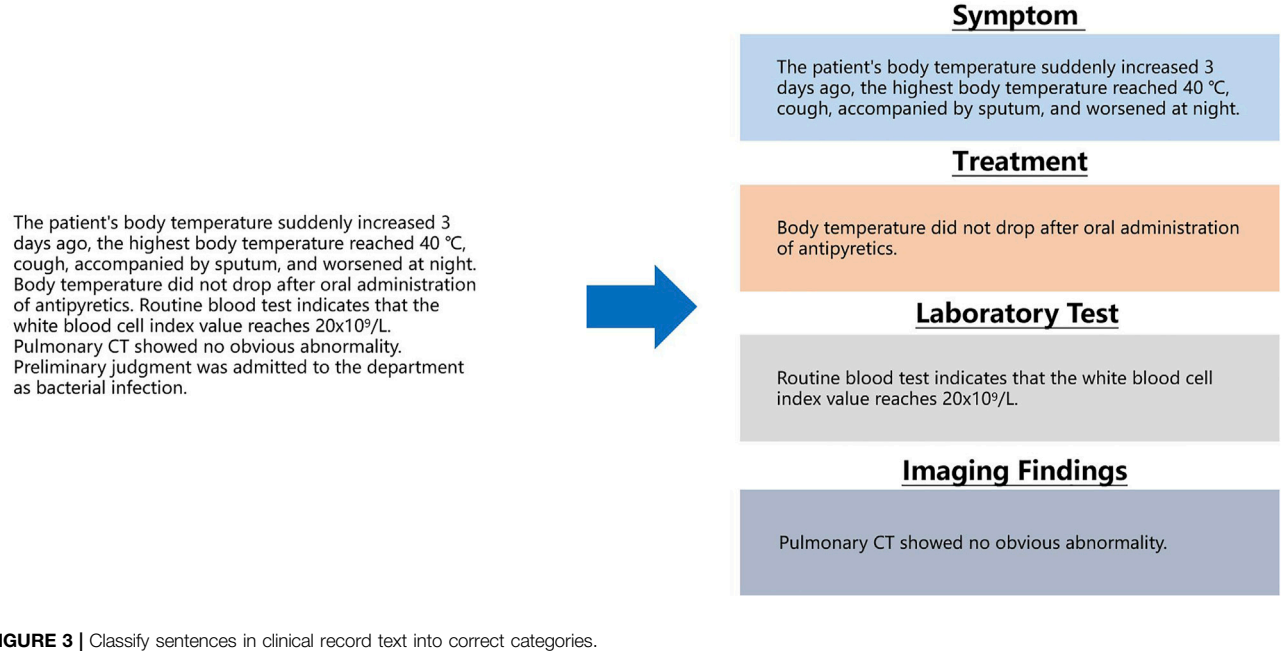


FIGURE 3 | Classify sentences in clinical record text into correct categories.

upper and lower relationships and forms a complex network of directed knowledge elements. This approach reflects the potential logical relationship between knowledge elements in TCM. Wu et al. (2021) used public medical quiz information and encyclopedia data on the Internet. The researchers proposed the co-training double word embedding conditioned bidirectional long short-term memory (CTD-BLSTM) computing architecture to improve the accuracy of medical

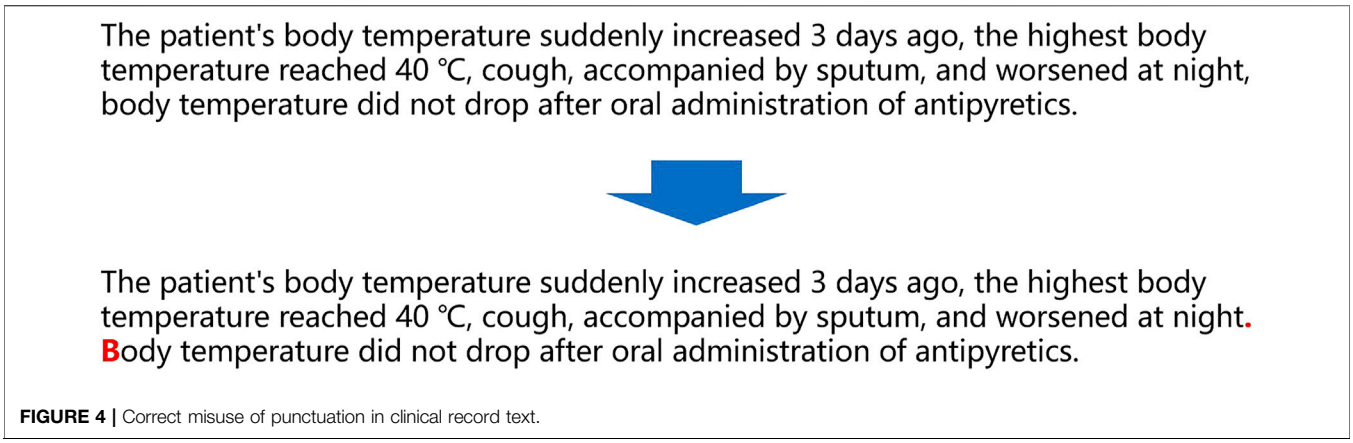


FIGURE 4 | Correct misuse of punctuation in clinical record text.

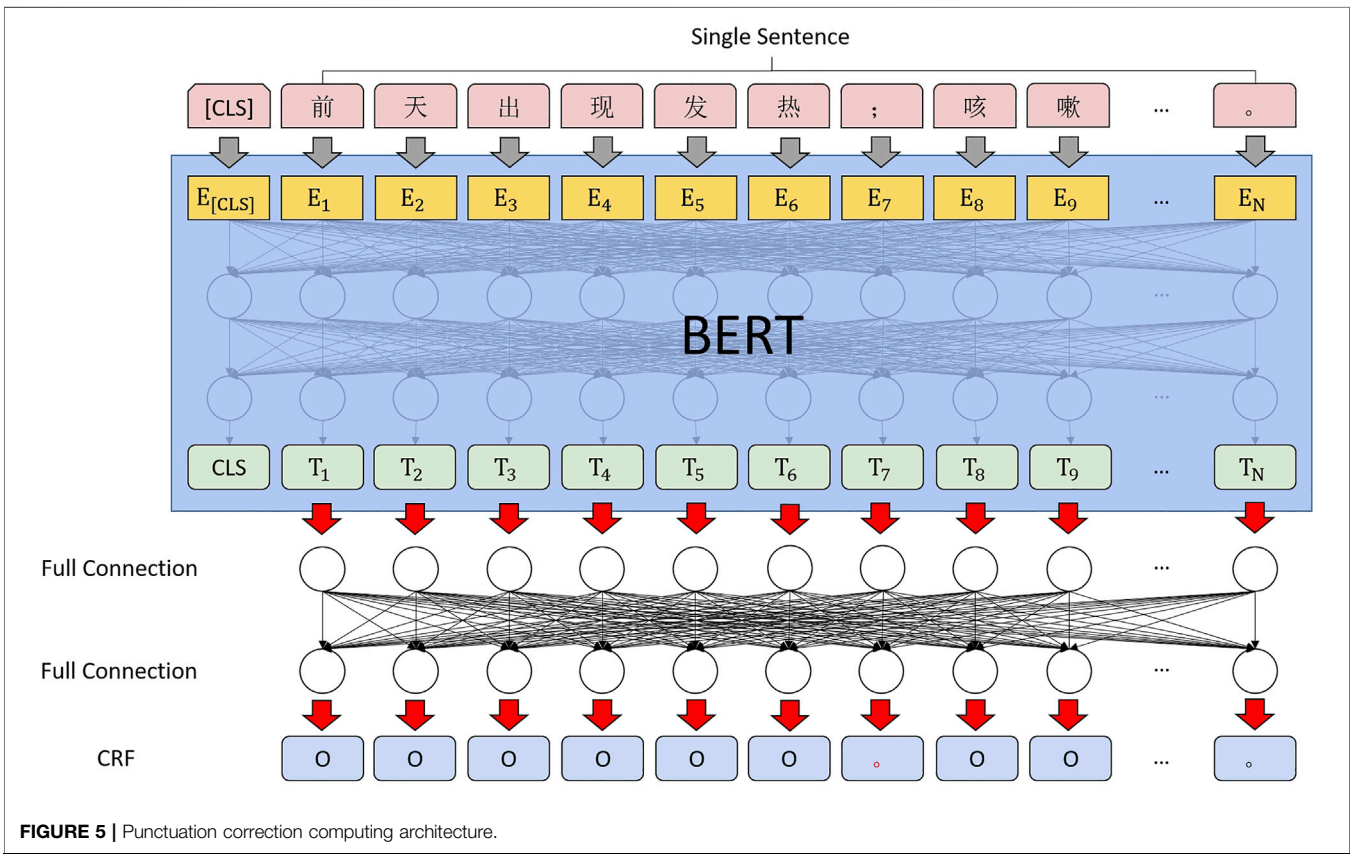


FIGURE 5 | Punctuation correction computing architecture.

named entities and entity relationships in the Chinese field and to provide better support for constructing a Chinese medical knowledge graph.

We have summarized and discussed the current related research on the construction of medical knowledge graphs and discovered that most researchers usually analyze the problem from the perspective of computer practitioners when conducting research. From the triad structure born from the knowledge graph until now, researchers in the industry have proposed the tuple data structure. These studies always use algorithms to achieve better computational accuracy and more diverse ways of reasoning. Just as doctors need to obtain multidimensional information in evidence-based medicine to diagnose diseases, medical entities also need multidimensional information to be fully expressed. We do not suggest that an ordinary triad can express the complete relationship between two medical entities. For example, the relationship between "fever-cough" and "fever-body temperature" or "fever-duration" are not in the same dimension. Building a knowledge graph from clinical data

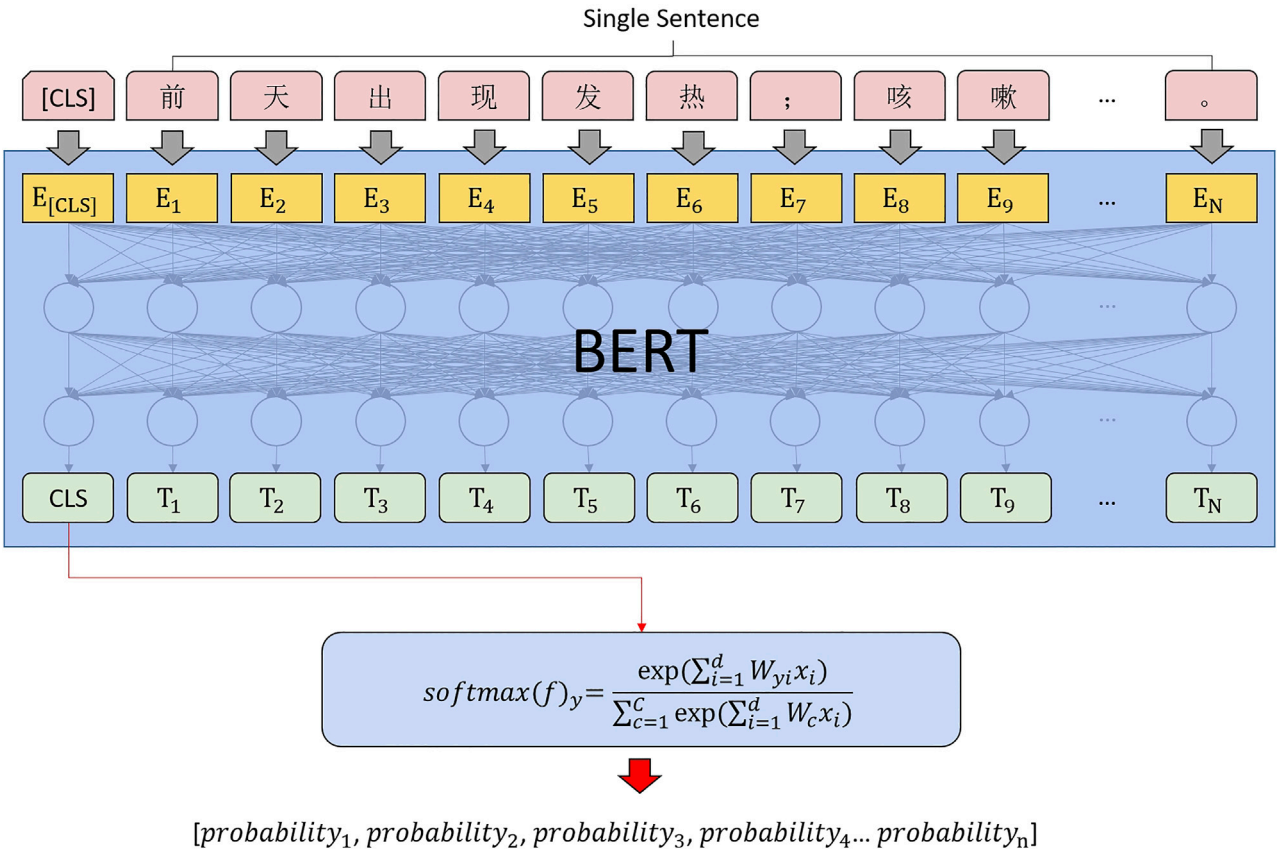


FIGURE 6 | Sentence classification computing architecture.

requires deeper data structures and computational architectures.

3 MATERIALS AND METHODS

The electronic clinical record covers the patient's condition and the diagnosis and treatment process. A point worthy of discussion is whether different doctors follow fixed rules when recording clinical records. Although we have not identified relevant rules and regulations in the medical industry, we have noticed that in the process of multidisciplinary treatment (MDT), clinicians from different departments, hospitals, and even countries can analyze a condition based on the same clinical record data. However, different clinicians can read the same clinical records, which also indicates that clinicians follow the rules of a fixed pattern in the medical industry when recording clinical records. Although this invisible rule should follow the basic logic of clinical diagnosis and treatment, it also standardizes the information presentation structure of clinicians when writing clinical records. This rule is the logic by which we extract medical entities from clinical records through algorithms.

By reviewing a numerous clinical records, we discovered that the logic of clinicians in writing clinical records is very clear. Consider the "Admission Record - Present Illness History", which records the patient's condition when they are admitted to the hospital as an example. Clinicians described the patient's symptoms, treatment methods, key indicators of laboratory examinations, and imaging findings in several sentences in the clinical record text. Proceeding to the next level of analysis, in the description of the patient's symptoms, the symptoms, degree, physical indicators associated with symptoms (such as recording body temperature during fever), cause of occurrence, time of occurrence, duration, aggravating factors, and mitigating factors. When describing the treatment method, for operation treatment, the type and date of the operation will be recorded; for medication treatment, the name of the drug, the dose, and the number of times will be recorded. A recording laboratory test will record the names and values of important indicators. The type of imaging examination, examination site, and abnormal findings will be recorded for imaging examination. These records can almost be the record rules that any hospital, department, and clinician will follow. The logical structure of these records is the same entity structure employed when we extract information. To extract medical entities from such clinical records, we can split them into the following process:

We want to extract medical entities that need to conform to medical logic and have an "object-attribute" structure. Therefore,

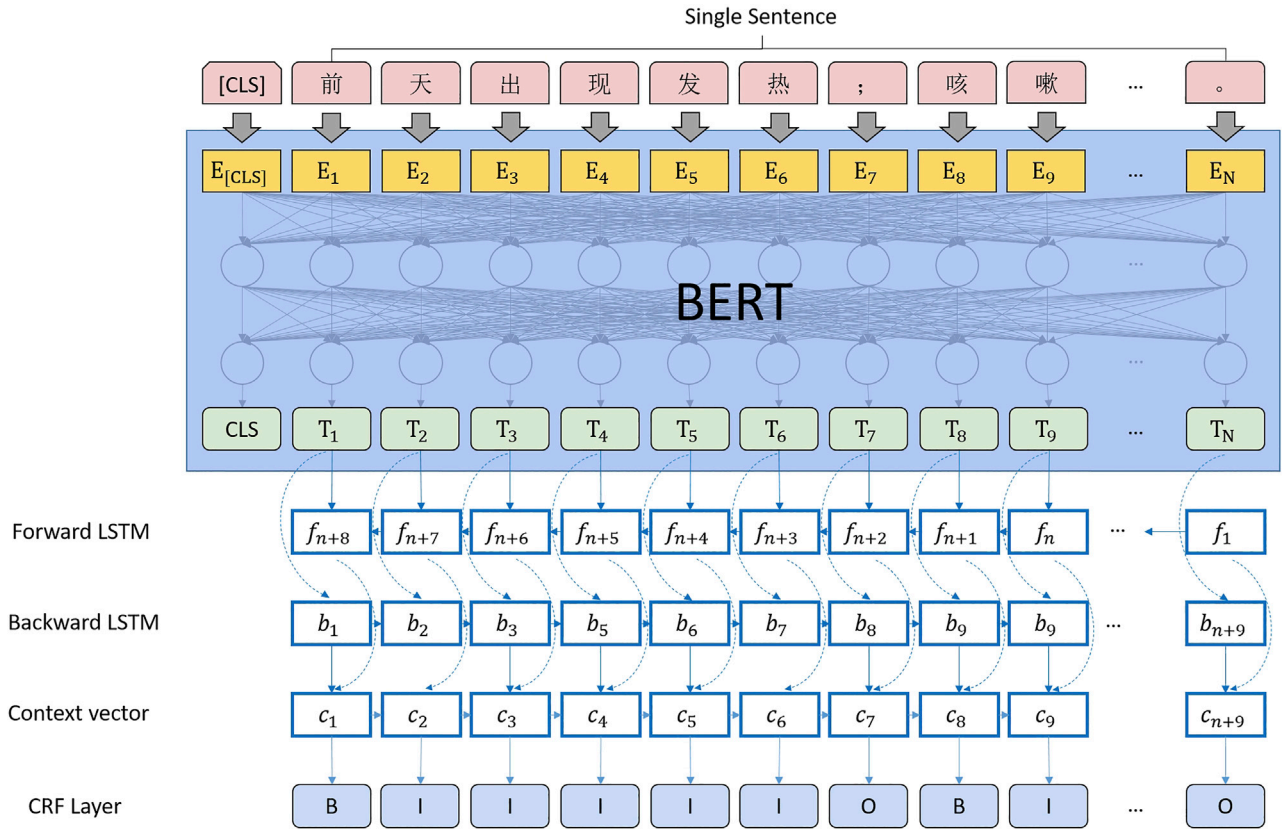


FIGURE 7 | Entity extraction computing architecture.

we have to extract the entity's attributes from the description related to each medical entity, as shown below in **Figure 1**.

In the above example, the text on the left is a segment from the clinical record text that describes the patient's fever. Extracting "body temperature" and "occurrence time" from this segment can be performed by a sequence labeling algorithm. However, note that "body temperature" is unique to the symptom "fever". When extracting this kind of information, it is necessary to know in advance that the current segment describes "fever". When doctors describe patients' symptoms, they usually make a centralized record in the same sentence. To obtain the fever description segment in the clinical records required for the above calculation, we designed a calculation as shown in the following **Figure 2**.

The content shown in the above figure can be understood as the need to segment the description of fever and cough from a sentence describing a patient's symptoms and to give corresponding symptom labels. This process can be conducted by long entity recognition in sequence labeling computation. The next problem then becomes that we need to classify the sentences in the text clinical records into a known category. As previously described, when recording the basic condition, clinicians usually use several fixed sentence patterns (symptoms, treatment methods, key indicators of laboratory examinations, and imaging findings). Using text classification computing to complete this task is a good choice as shown in the following **Figure 3**.

As shown in the above figure, as long as the sentences in the text clinical records are calculated through the classification calculation, the corresponding categories of the sentences are obtained, and entity recognition and entity attribute recognition can be performed. However, in actual work, we discovered an easily overlooked detail. When Chinese clinicians write clinical records, punctuation is irregular, and even the entire clinical records are separated by commas. For this kind of irregularity, there is no hospital or relevant department to supervise. Although this irregularity does not affect human reading, for computers, this irregularity will produce low-precision classification calculations due to unclear sentence boundaries. To solve this problem, a punctuation correction calculation needs to be prepended before the clinical record sentence classification calculation as shown in the following **Figure 4**.

The above content describes the researcher's final plan to use four steps to extract medical entities after analyzing the logic in the text clinical records. The four steps are arranged in positive order based on data processing, namely, "punctuation correction", "sentence classification", "medical entity extraction", and "entity object attribute extraction".

3.1 Punctuation Correction

We obtained a random sample of 500 medical records from the EMRs of hospital departments. The count revealed that a total of

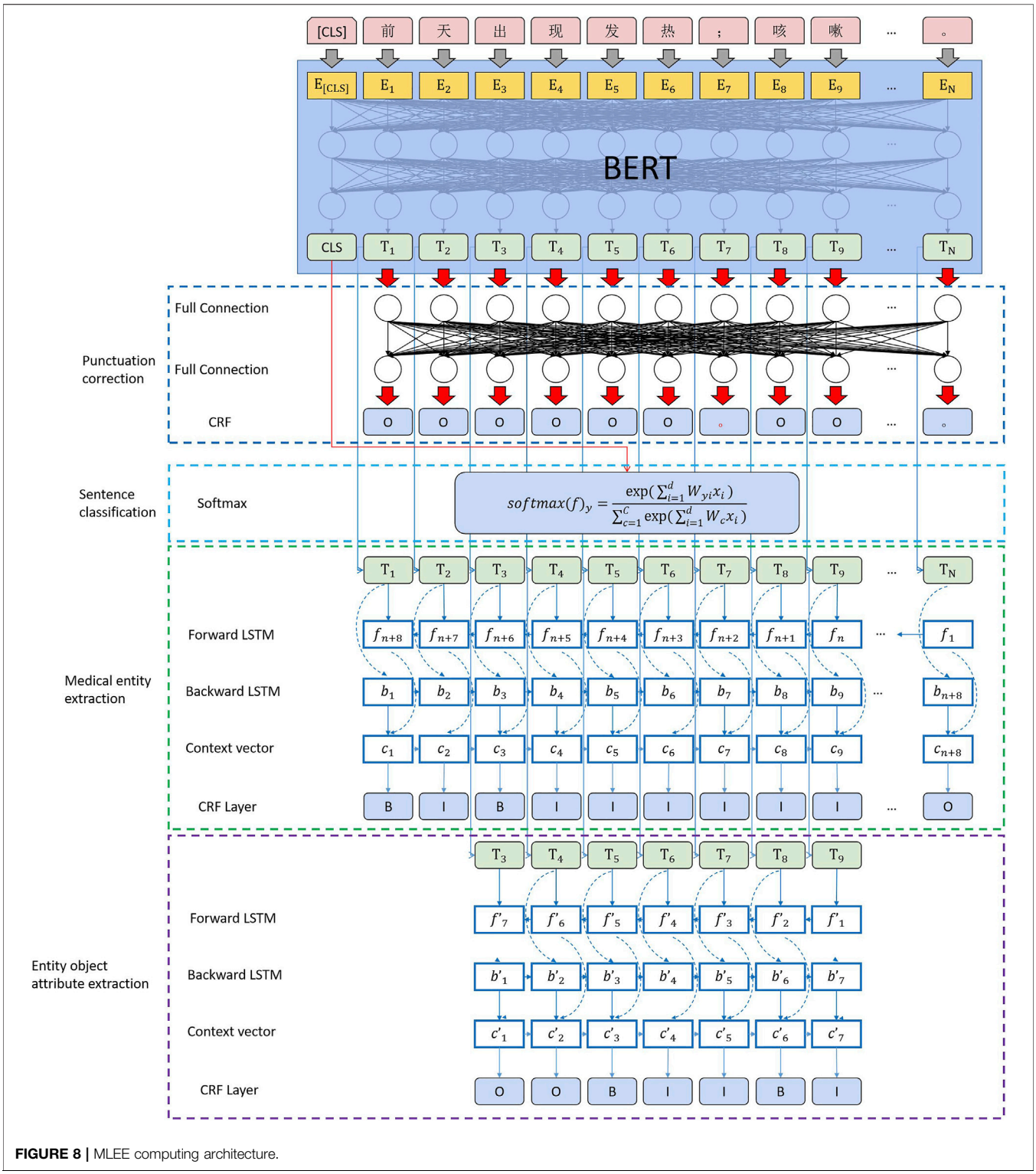


FIGURE 8 | MLEE computing architecture.

16,764 punctuation marks were utilized in these cases. According to the rules, we manually confirm the existing punctuation in the clinical medical record and correct the incorrect punctuation in the medical record. If manual correction was employed as the standard, the punctuation correctness rate for clinicians writing medical records was only 16.4%.

Based on this manually modified database, we plan to build a sequence annotation model. An elementary and effective

TABLE 1 | Medical knowledge graph schema label for information extraction.

Entity Type	Entity	Attributes
Symptom	Fever	Body Temperature
		Occurrence
		Duration
	Cough	Occurrence
		Duration
		Aggravating Factor
		Relieving Factor
		Cough Frequency
		Situation
Treatment	Medication Treatment	Drug name
		Drug dose
		Duration of course of treatment
	Operation	Type of operation
		Date of operation
		Adverse reactions
Laboratory Test	Laboratory Test Entity	Test item
		Value
Imaging	Computed Tomography	Body part
		Abnormal seen
	Magnetic Resonance Imaging	Body part
		Abnormal seen
		T1WI
Other		T2WI

neural network was constructed to accommodate the punctuation correction and subsequent information extraction. In the embedding layer, we chose to use the Bidirectional Encoder Representations from Transformers (BERT) model (Devlin et al., 2018). Although we initially tried to use Word2Vec for embedding based on a large amount of data, the results were approximately 4% lower than those based on BERT.

In building the actual sequence annotation, we made some changes to the original BERT, which processed tokens by slicing most characters. For example, we discovered that slicing words could sometimes significantly impact the meaning of Chinese expressions (Névéol et al., 2018). We therefore reworked the token in BERT to slice and dice by any individual character.

We tried to discard the long short-term memory (LSTM) (Greff et al., 2016) during the calculation of the sequence annotation of the correction markers. The transformer performs much better than the recurrent neural network (RNN) in many tasks. As Chinese words are stitched together from multiple characters, the profession usually uses the transformer's output at the last encoder layer in BERT as input for subsequent docking of bidirectional LSTM with a conditional random field (Bi-LSTM + CRF) (Huang et al., 2015). However, since the sequence information in the transformer itself is sufficient, obtaining the sequence information of the context by using RNN (LSTM) again is unnecessary (Feng et al., 2018). We also wanted to give the neural network as much information as possible by appending a CRF after the last fully connected CRF. The computing architecture is shown in **Figure 5**.

TABLE 2 | Effect of each calculation step of MLEE.

Computational Procedure	Precision	Recall	F1 value
Punctuation correction	0.9874	0.9529	0.9698
Sentence classification	0.9812		
Medical entity extraction	0.9611	0.9438	0.9524
Entity object attribute extraction	0.9638	0.9611	0.9624

3.2 Sentence Classification

According to the information obtained through the EMR system, the actual patient will generate 27 subcategories of clinical records.

After considering all types of clinical records, we discovered that the same types of sentences occur in many different types of medical record types. Treatment-related descriptions appear in the "past history", "treatment plan", "discharge instructions and rehabilitation instructions" and other types of medical records. If one follows this pattern, there must be a range of sentence types that can cover the semantic content of all types of medical records (Frunza and Inkpen, 2010). The clustering of all statements in the clinical records was calculated using the clustering calculation (Rodriguez and Laio, 2014), and the validity of the current clustering results was verified using the silhouette coefficient.

We then manually observed the clustering results, and after merging the two smaller clusters based on the semantics of the clinical history statements, we obtained 18 clusters. Afterward, the content of the utterances in each cluster was again manually and semantically confirmed, and medical semantic description labels were associated with each of these 18 clusters. This labelling includes a description of symptoms, treatment, signs and symptoms, specialist examination, examination information, etc.

TABLE 3 | Labels for flat transformation using the schema.

Entity type	Entity	Attributes	NER Label
Symptom	Fever	Body Temperature	**Fever-Body Temperature**
		Occurrence	**Fever-Occurrence**
		Duration	**Fever-Duration**
	Cough	Occurrence	**Cough-Occurrence**
		Duration	**Cough-Duration**
		Aggravating Factor	**Cough-Aggravating Factor**
		Relieving Factor	**Cough-Relieving Factor**
		Cough Frequency	**Cough-Cough Frequency**
		Situation	**Cough-Situation**
Treatment	Medication Treatment	Drug name	**Medication Treatment-Drug name**
		Drug dose	**Medication Treatment-Drug dose**
		Duration of course of treatment	**Medication Treatment-Duration of course of treatment**
	Operation	Type of operation	**Operation-Type of operation**
		Date of operation	**Operation-Date of operation**
		Adverse reactions	**Operation-Adverse reactions**
Laboratory Test	Laboratory Test Entity	Test item	**Laboratory Test Entity-Test item**
		Value	**Laboratory Test Entity-Value**
Imaging	Computed Tomography	Body part	**Computed Tomography-Body part**
		Abnormal seen	**Computed Tomography-Abnormal seen**
	Magnetic Resonance Imaging	Body part	**Magnetic Resonance Imaging-Body part**
		Abnormal seen	**Magnetic Resonance Imaging-Abnormal seen**
		T1WI	**Magnetic Resonance Imaging-T1WI**
		T2WI	**Magnetic Resonance Imaging-T2WI**

The bold values indicate NER label, it represents the label used to annotation the real data.

We constructed a text classifier based on BERT + FC + Softmax (Kim, 2014) as shown in **Figure 6**; the model was validated in multiple rounds by cross-validation.

3.3 Medical Entity Extraction and entity Object Attribute Extraction

After completing punctuation correction and sentence classification, the final entity description segment extraction and entity attribute extraction process can be understood as a short text sequence annotation.

The semantic scope of entities and attributes in the medical field is relatively small, and the semantic space of the text to be extracted has been fixed through the above two steps, which is a very simple calculation scenario for sequence labeling.

Since the entire computing architecture needs to be merged to ensure the consistency of feature extraction, BERT + Bi-LSTM + CRF is selected for sequence annotation, as shown in **Figure 7**.

3.4 Computing Architecture

We built the computing architecture, as shown in **Figure 8**. After using BERT to complete the feature conversion of text data, we realize the extraction and calculation of medical entities by connecting four downstream tasks. The detailed process is presented as follows:

1) Complete the punctuation correction calculation using a fully connected layer and conditional random fields.
2) Use the CLS vector generated by BERT for the sentence and complete the sentence classification through softmax.

TABLE 4 | Comparison of MLEE information extraction and traditional sequence labeling.

Method	F1 value
Bert + BiLSTM + CRF	0.9367
MLEE	**0.9624**

The bold values indicate experiment results of the method proposed in this paper.

3) Sequence annotation of medical entity segments using bidirectional LSTM and CRF.
4) Perform the final medical entity attribute extraction using bidirectional LSTM and CRF.

In this computing architecture, it is necessary to explain the change in the loss function of BERT in the upstream computing process in the multi-downstream task scenario.

$$\mathrm{Loss}(\theta, \widetilde{\theta_1}, \theta_2) = \mathrm{Loss}(\theta, \widetilde{\theta_1}) + \mathrm{Loss}(\theta, \theta_2) \quad (1)$$

where θ represents the parameters of the Encoder part in BERT. $\widetilde{\theta_1}$ in the original BERT paper represents the parameters in the output layer connected to the encoder in the masked-language modeling (LM) task. This study represents the parameter combination of three sequence annotations after being output by the encoder. θ_2 The original paper represents the classifier parameters connected to the encoder in the sentence prediction task. This study represents the classifier parameters in the classification calculation of text medical record sentences. Details are presented as follows:

$$\mathrm{Loss}(\theta, \theta_{11}) = -\Sigma_{i=1}^{M} \log \mathrm{P}(m = m_i | \theta, \theta_{11}), m_i \in [1, 2 \ldots, |\mathrm{Punctuation\ Set}|] \quad (2)$$

where θ_{11} represents the parameters in the output layer connected to the encoder in the punctuation correction sequence labeling task.

$$\mathrm{Loss}(\theta, \theta_{12}) = -\Sigma_{j=1}^{N} \log P\left(n = n_j | \theta, \theta_{12}\right), n_j \in [1, 2 \ldots, |\text{Medical Entity Set}|] \quad (3)$$

θ_{12} represents the parameters in the output layer connected to the encoder in the medical entity description segment sequence labeling task.

$$\mathrm{Loss}(\theta, \theta_{13}) = -\Sigma_{k=1}^{N} \log P(o = o_k | \theta, \theta_{13}), o_k \in [1, 2 \ldots, |\text{Entity Attribute Set}|] \quad (4)$$

θ_{13} represents the parameters in the output layer connected to the encoder in the medical entity attribute sequence labeling task.

$$\mathrm{Loss}(\theta, \widetilde{\theta_1}) = \mathrm{Loss}(\theta, \theta_{11}) + \mathrm{Loss}(\theta, \theta_{12}) + \mathrm{Loss}(\theta, \theta_{13}) \quad (5)$$

The loss of the three downstream sequence labeling tasks is added to obtain $\mathrm{Loss}(\theta, \widetilde{\theta_1})$.

$$\mathrm{Loss}(\theta, \theta_2) = -\Sigma_{l=1}^{H} \log P(h = h_l | \theta, \theta_2), h_l \in [\mathrm{label}_1, \mathrm{label}_2 \ldots, \mathrm{label}_x] \quad (6)$$

In the second part, $\mathrm{Loss}(\theta, \theta_2)$ is the loss function of the sentence classification task.

4 EXPERIMENT

This chapter introduces the experiment in three parts. The first part concerns data sources, the definition of medical entities in the schema, and data annotation. In the second, we introduce the extraction of medical entities based on the computational architecture proposed in this study. Since there is currently no open-source text clinical record dataset in the Chinese field and based on the diseases involved in the current clinical records (pediatric respiratory diseases), there is no unified knowledge map schema standard. This paper temporarily evaluates the effect based on the data extraction accuracy of the in-hospital data based on the data standard jointly constructed by the author and the clinicians of Shengjing Hospital of China Medical University. In the third part, we test all the entity attributes of the custom schema by flattening to test whether the computational architecture proposed in this study has an accuracy loss comparable with the general sequence annotation.

4.1 Data Preparation

We randomly selected the current illness histories of 1,000 patients from the inpatient clinical records at Shengjing Hospital of China Medical University. We discussed them with clinicians and learned about their concerns about writing and reading clinical records. Combined with the definition of medical fields in the Snomed CT International Edition, the medical entities and attribute labels in the schema are sorted, as shown **Table 1**.

Based on the above labels, we use "entity type" as the classification calculation label of medical record sentences, "entity" as the sequence annotation label of medical entity segments, and "attribute" as the sequence annotation label of medical entity attributes. In the process of punctuation correction, the "period" is corrected to ensure that these sentences can be correctly split. The data were labeled according to the table by clinicians and used as the gold standard.

According to the above rules, we manually marked 7,029 sentences (3,418 punctuation points were manually corrected, and the error rate of punctuation used by doctors reached 48.6%), 10,467 medical entities, and 29,478 medical attributes based on the clinical medical records of 1,000 patients. entities with 2.82 attributes).

4.2 Description of Effect

The above data and the entity labels defined in schema model training and effect verification are carried out based on the computing architecture introduced in the previous chapter. The calculation effect of all steps is presented as **Table 2**.

The experimental results exceeded our expectations, and we subsequently analyzed the calculation results by decomposing steps. Most of the miscalculated punctuation is concentrated in the over segmentation of symptom-related descriptions in the punctuation correction step. For example, "fever" and "cough", which should be listed in the same sentence, are divided into two sentences. Such errors do not cause error propagation in subsequent computations. In the sentence classification step, because we built an "Other" category to carry some content in the clinical record about the patient's general condition before admission, the patient's body temperature, mental state, appetite, and other related information may be included. Some of these sentences are divided into "symptom" labels for the last two sequence annotation computations. Although the input of the final entity attribute sequence annotation labeling is the output of the previous layer of medical entity segment sequence annotation labeling, the error propagation will be critical. However, the results indicate that the accuracy of the lower layer calculation is higher than that of the upper layer calculation. The researchers determined that when calculating the medical entity segment, precision and recall may decrease due to the error of one character before or after. However, as long as it contains all the characters required for the lower-level sequence annotation labeling, the correct result can still be obtained in the final entity attribute calculation.

4.3 Calculate Loss Assessment

To evaluate whether the superimposed computing architecture of this study will lose accuracy through error transmission, we compare the accuracy by flattening the labels in the schema. The sequence annotation labels used for testing are shown in the last column of **Table 3**.

The final comparison accuracy is shown as **Table 4**.

This conclusion also confirms that the method proposed in this study improves the information extraction accuracy compared with general sequence annotation and better expresses medical entities through the "object-attribute" structure. This finding provides a good data foundation for

constructing medical knowledge graphs and reasoning computations based on knowledge graphs.

5 CONCLUSION

In this paper, we propose a method for extracting medical entities using real Chinese clinical medical records. A medical knowledge graph based on clinical data can be constructed on this basis. We discovered that the same medical record data, simply based on entity co-occurrence, can be used as a high-quality relational to connect entities. If many cases, the data can be utilized as the research object, even directed probability edges can be obtained, which is the follow-up research direction of the research team.

DATA AVAILABILITY STATEMENT

The data analyzed in this study is subject to the following licenses/restrictions The data used in this study came from the hospital's electronic medical record system. All data used in the experiment did not involve any personal information of patients, and all data experiments were carried out in the hospital. Requests to access these datasets should be directed to Zhiying Zhao, zhaozy@sj-hospital.org.

AUTHOR CONTRIBUTIONS

GZ designed the study, performed measurements, designed the analysis, and wrote the manuscript. WG designed the analysis. WC designed the analysis. ZZ designed the medical schema and the data labeling standards. XZ designed the study and the analysis. JL designed the study and the analysis. All authors contributed to the article and approved the submitted version.

FUNDING

This research is supported by the National Key Research and Development Program of China No. 2020AAA0109400 and the Shenyang Science and Technology Plan Fund (No. 20-201-4-10).

REFERENCES

Abhyuday, J., Feifan, L., Weisong, L., and Hong, Y. (2020). Overview of the first natural language processing challenge for extracting medication, indication, and adverse drug events from electronic health record notes (made 1.0). *Drug Saf.* 42 (1), 99–111. doi:10.1007/s40264-018-0762-z

Cheng, M., Li, L. M., Ren, Y., Lou, Y., and Gao., J. (2019). A hybrid method to extract clinical information from Chinese electronic medical records. *IEEE Access* 7, 70624–70633. doi:10.1109/ACCESS.2019.2919121

Devlin, J., Chang, M. W., Lee, K., and Toutanova, K. (2018). *Bert: pre-training of deep bidirectional transformers for language understanding*. arXiv preprint arXiv: 1810.04805,2018.

Feng, Y. H., Hong, Y. U., Sun, G., and Sun, J. J. (2018). Named entity recognition method based on blstm. *Comput. Sci.* 45 (2), 261–268. doi:10.11896/j.issn.1002-137X.2018.02.045

Frunza, O., and Inkpen, D. (2010). "Extraction of Disease-Treatment Semantic Relations from Biomedical Sentences," in Proceedings of the 2010 Workshop on Biomedical Natural Language Processing, Uppsala, Sweden (Association for Computational Linguistics), 91–98.

Greff, K., Srivastava, R., Koutník, J., Steunebrink, B., and Schmidhuber, J. (2016). Lstm: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* 28 (10), 2222–2232. doi:10.1109/TNNLS.2016.2582924

Huang, Z., Wei, X., and Kai, Y. (2015). Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv: 1508.01991,2015.

Jiang, J., Zhao, C., Guan, Y., and Yu, Q. (2017). Learning and inference in knowledge-based probabilistic model for medical diagnosis. *Knowledge-Based Syst.* 138, 58–68. doi:10.1016/j.knosys.2017.09.030

Jiang, Z., Chi, C., and Zhan, Y. (2021). Research on medical question answering system based on knowledge graph. *IEEE Access* 9, 21094–21101. doi:10.1109/access.2021.3055371

Jonnagaddala, J., Liaw, S. T., Ray, P., Kumar, M., Chang, N. W., Dai, H. J., et al. (2015). Coronary artery disease risk assessment from unstructured electronic health records using text mining. *J. Biomed. Inf.* 58 (Suppl. l), S203–S210. doi:10.1016/j.jbi.2015.08.003

Kang, T., Zhang, S., Xu, N., Wen, D., Zhang, X., and Lie, J. (2017). Detecting negation and scope in Chinese clinical notes using character and word embedding. *Comput. Methods Programs Biomed.* 140, 53–59. doi:10.1016/j.cmpb.2016.11.009

Kim, Y. (2014). "Convolutional Neural Networks for Sentence Classification," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar (Association for Computational Linguistics), 1746–1751.

Li, L., Wang, P., Yan, J., Wang, Y., Liu, Y., Jiang, J., et al. (2020). Real-world data medical knowledge graph: Construction and applications. *Artif. Intell. Med.* 103 (19), 101817. doi:10.1016/j.artmed.2020.101817

Li, X., Liu, H., Zhao, X., Zhang, G., and Xing, C. (2020). Automatic approach for constructing a knowledge graph of knee osteoarthritis in Chinese. *Health Inf. Sci. Syst.* 8 (1), 12. doi:10.1007/s13755-020-0102-4

Liu, X., and Xu, L. (2021). "Knowledge graph building from real-world multisource "dirty" clinical electronic medical records for intelligent consultation applications," in 2021 IEEE International Conference on Digital Health (ICDH), Chicago, IL, USA, 05-10 Sep. 2021, 260–265.

Louise, D., Cyril, G., and Pierre, Z. (2010). Extracting medical information from narrative patient records: The case of medication-related information. *J. Am. Med. Inf. Assoc.* 17 (5), 555–558. doi:10.1136/jamia.2010.003962

Mykowiecka, A., Marciniak, M., and Kup, A. (2009). Rule-based information extraction from patients' clinical data. *J. Biomed. Inf.* 42 (5), 923–936. doi:10.1016/j.jbi.2009.07.007

Névéol, A., Dalianis, H., Velupillai, S., Savova, G., and Zweigenbaum, P. (2018). Clinical Natural Language processing in languages other than English: Opportunities and challenges. *J. Biomed. Semant.* 9 (1), 12. doi:10.1186/s13326-018-0179-8

Qiu, J., Zhou, Y., Wang, Q., Ruan, T., and Gao, J. (2019). Chinese clinical named entity recognition using residual dilated convolutional neural network with conditional random field. *IEEE Trans. Nanobioscience* 18, 306–315. doi:10.1109/TNB.2019.2908678

Rodriguez, A., and Laio, A. (2014). Machine learning. Clustering by fast search and find of density peaks. *Science* 344 (6191), 1492–1496. doi:10.1126/science.1242072

Shi, L., Li, S., Yang, X., Qi, J., Pan, G., and Zhou, B. (2017). Semantic health knowledge graph: Semantic integration of heterogeneous medical knowledge and services. *Biomed. Res. Int.*, 2017, 1–12. doi:10.1155/2017/2858423

Uzuner, Ö., Solti, I., and Cadag, E. (2010). Extracting medication information from clinical text. *J. Am. Med. Inf. Assoc.* 17, 514–518. doi:10.1136/jamia.2010.003947

Weng, H., Liu, Z., Yan, S., Fan, M., Ou, A., Chen, D., et al. (2017). A Framework for Automated Knowledge Graph Construction Towards Traditional Chinese Medicine. *Health Information Science, HIS 2017*. Lecture Notes in Computer Science (Cham: Springer) 10594, 170–181.

Wu, Y., Zhu, X., and Zhu, Y. (2021). An improved approach to the construction of Chinese medical knowledge graph based on CTD-BLSTM model. *IEEE Access* 9, 74969–74976. doi:10.1109/access.2021.3079962

Zhao, C., Jiang, J., Xu, Z., and Guan, Y. (2017). A study of emr-based medical knowledge network and its applications. *Comput. Methods Programs Biomed.* 143, 13–23. doi:10.1016/j.cmpb.2017.02.016

Zheng, L., Liu, S., Song, Z., and Dou, F. (2021). Diversity-aware entity exploration on knowledge graph. *IEEE Access* 9, 118782–118793. doi:10.1109/access.2021.3107732

Zhu, L., Gao, H., Lili, X., Tong, Y., Shun, X., Xu, L., et al. (2017). Knowledge graph for TCM health preservation: Design, construction, and applications. *Artif. Intell. Med.* 77, 48–52. doi:10.1016/j.artmed.2017.04.001

Conflict of Interest: JL was employed by Neusoft Corporation.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers | Frontiers in Genetics

TYPE Original Research
PUBLISHED 26 July 2022
DOI 10.3389/fgene.2022.958069

OPEN ACCESS

EDITED BY
Yucong Duan,
Hainan University, China

REVIEWED BY
Souvik Roy,
Indian Institute of Technology Kanpur, India
Jiangan Xie,
Chongqing University of Posts and Telecommunications, China
Teijiro Isokawa,
University of Hyogo, Japan

*CORRESPONDENCE
Chunrong Wu,
crwu@cqu.edu.cn

SPECIALTY SECTION
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

RECEIVED 31 May 2022
ACCEPTED 27 June 2022
PUBLISHED 26 July 2022

CITATION
Xu W, Wu C, Peng Q, Lee J, Xia Y and Kawasaki S (2022), Enhancing the diversity of self-replicating structures using active self-adapting mechanisms. *Front. Genet.* 13:958069. doi: 10.3389/fgene.2022.958069

Enhancing the diversity of self-replicating structures using active self-adapting mechanisms

Wenli Xu[1], Chunrong Wu[1]*, Qinglan Peng[1], Jia Lee[1,2], Yunni Xia[1,2] and Shuji Kawasaki[3]

[1]College of Computer Science, Chongqing University, Chongqing, China, [2]Chongqing Key Laboratory of Software Theory and Technology, Chongqing, China, [3]Faculty of Science and Engineering, Iwate University, Morioka, Japan

Numerous varieties of life forms have filled the earth throughout evolution. Evolution consists of two processes: self-replication and interaction with the physical environment and other living things around it. Initiated by von Neumann et al. studies on self-replication in cellular automata have attracted much attention, which aim to explore the logical mechanism underlying the replication of living things. In nature, competition is a common and spontaneous resource to drive self-replications, whereas most cellular-automaton-based models merely focus on some self-protection mechanisms that may deprive the rights of other artificial life (loops) to live. Especially, Huang et al. designed a self-adaptive, self-replicating model using a greedy selection mechanism, which can increase the ability of loops to survive through an occasionally abandoning part of their own structural information, for the sake of adapting to the restricted environment. Though this passive adaptation can improve diversity, it is always limited by the loop's original structure and is unable to evolve or mutate new genes in a way that is consistent with the adaptive evolution of natural life. Furthermore, it is essential to implement more complex self-adaptive evolutionary mechanisms not at the cost of increasing the complexity of cellular automata. To this end, this article proposes new self-adaptive mechanisms, which can change the information of structural genes and actively adapt to the environment when the arm of a self-replicating loop encounters obstacles, thereby increasing the chance of replication. Meanwhile, our mechanisms can also actively add a proper orientation to the current construction arm for the sake of breaking through the deadlock situation. Our new mechanisms enable active self-adaptations in comparison with the passive mechanism in the work of Huang et al. which is achieved by including a few rules without increasing the number of cell states as compared to the latter. Experiments demonstrate that this active self-adaptability can bring more diversity than the previous mechanism, whereby it may facilitate the emergence of various levels in self-replicating structures.

KEYWORDS

self-replication, self-adaption, cellular automaton, gene mutation, biological resources

1 Introduction

A cellular automaton (CA) is a discrete dynamical system that consists of a huge number of identical finite-state automata (Abou-Jaoudé et al., 2016; Xiao et al., 2020). Self-replication is a fundamental feature of life in biological resources, and it is a process of biosynthesis in which the original structure is replicated in the exact same structure (Cea et al., 2015; Baris et al., 2022; Gemble et al., 2022). Research of self-replication on CAs was founded by von Neumann (1966) and was viewed as one of the origins of artificial life research (Marchal, 1998; Gindin et al., 2014). In addition to reproducing offsprings with identical structures, attempts at including self-adapting mechanisms into the self-replicating models have been done (Suzuki and Ikegami, 2003; Sayama, 2004; Huang et al., 2013). In particular, Huang et al. (2013) designed a self-adaptive, self-replicating model using a greedy selection mechanism, which can increase the ability of the loops to survive through an occasionally abandoning part of their own structural information, for the sake of adapting to the restricted environment. Although the greedy mechanism is straightforward and sounds natural, it seems too passive. In addition to the self-adaptation which helps organisms survive (Williams and Burt, 1997), evolution and mutation are also inherent abilities of living things for adapting to environments in more active ways (Agrawal, 2001; Wilke et al., 2001; Miles et al., 2020; Moore et al., 2021; Monroe et al., 2022; Sasani et al., 2022), like the RNA virus (Domingo and Holland, 1997).

Likewise, identification of multiple adaptive mutations turns out to be essential for studying adaptation (Aminetzach et al., 2005; Scott, 2013; Lawson et al., 2020; Zuko et al., 2021). And, point mutations including insertions and replacements can help perform edits in human cells, thereby, in principle, correcting up to most of the known genetic variants associated with human diseases (Poduri et al., 2013; Anzalone et al., 2019; Buisson et al., 2019). Especially, changes in the self-replicating structure and behavior are controlled via their genetic memory (Bilotta and Pantano, 2006; Sha et al., 2020). As the living environment becomes more and more hostile, living organisms may have to change their own structures to survive. Self-adaptation through gene mutation, therefore, provides a spontaneous drive for natural life to survive against crucial competition with other living things and evolve into more advanced forms (Bilotta and Pantano, 2006; Sha et al., 2020). Moreover, self-adaptation has gained much attention in other fields such as knowledge architecture discovering (Edwards et al., 2009; Duan, 2019; Lei and Duan, 2021; Li et al., 2021) and edge computing (Xia et al., 2015; Song et al., 2018), due to its promise of more sophisticated and flexible computational paradigms (Duan et al., 2019a,b).

Inspired by the gene mutation-based self-adaptability in nature, this article endows two active mechanisms to the self-replicating loops which can facilitate the dynamical adaption of their structures to limited cellular regions. The new active mechanisms only need to change some rules in the passive model Huang et al. (2013), without increasing the number of cell states. The self-replication progress also contains two stages. In the first stage, the shape-encoding scheme is utilized to generate genetic information (construction signals), and the constructed arm receives the genetic codes to stretch forward, rightward, or leftward. During this period, collisions may occur at any moment and it seems urgently necessary to find a way out of a stalemate. Similar to the gene mutation process, we propose two solutions to resolve the collision. One mechanism generates, rather than waiting , a genetic code which resembles the insert mutation from single point mutation (Bargmann et al., 1986; Shenhav and Zeevi, 2020). Especially, the insertion of a transposable element can increase *Drosophila*'s resistance to an organophosphate pesticide (Aminetzach et al., 2005), which helps *Drosophila* to survive. In order to simplify the rules Huang et al. (2013), we randomly change the direction of the construction arms' head. Another mechanism will choose to change following the genetic code from the mother loop next to the construction arm, which is similar to replace mutation (Vogel, 1972). The method of replacing genetic codes is used in suppression of tumorigenicity of human prostate carcinoma cells (Bookstein et al., 1990). After finishing the first extension stage of the construction arm, the mother loop will send a validation signal to the arm for the sake of confirming whether there is a closed loop or not. If it succeeds, the signal will cut off the link between the child loop and mother loop; otherwise, the construction arm will be drawn back. Finally, several typical and initial configurations are selected for the numerical experiments, which demonstrate that our new active mechanisms can obtain more types of variation loops, thereby increasing the opportunities of the organisms' survival and expanding biodiversity (Klimentidis, 2012; Becerra-Rodríguez et al., 2021).

This article is organized as follows: Section 2 reviews related works. Section 3 gives an overview of the self-timed cellular automata and describes self-replicating loops with two active mechanisms which are capable of self-adapting their structures when the space is not enough to replicate themselves completely. Detailed comparison experiments are done in Section 4, followed by discussions given in Section 5.

2 Related works

Self-reproduction is one of the fundamental features in nature. Von Neumann was able to exhibit a universal Turing machine embedded in a cellular space using 29-states per cell and the 5-cell neighborhood. After that, many studies were done to reduce the complexity of the machine (Codd, 2014), re-mold signal-crossing organs (Buckley and Mukherjee, 2005), and

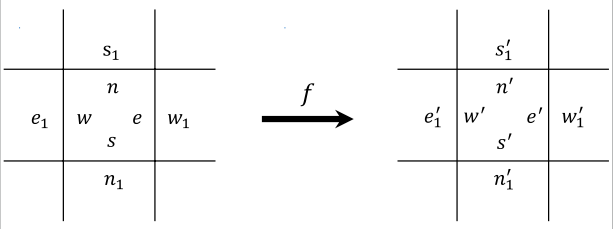


FIGURE 1
A transition rule according to the function *f*.

realize self-replicating in the hardware (Merkle, 1992; Pesavento, 1995; Tempesti et al., 1998).

After ignoring the universality in computations, Langton (1984) proposed a simple self-replicating loop based on the periodic emitter (Codd, 2014) in a two-dimensional cellular space. Langton's loop uses 8-states and 5-cell neighborhood (von Neumann neighborhood). After that, Langton' loop attracts much attention and various attempts have been done, such as deleting the external sheath (Tempesti, 1995) or the inner sheath (Byl, 1989), producing unsheathed loops with less states (Reggia et al., 1993), and considering self-replication on asynchronous cellular automata (Nehaniv, 2002). Likewise, Ibáñez et al. (1995) introduced the ability of self-inspection, which allows the genome to dynamically construct concomitantly with its interpretation. Making full of the self-inspection ability, Morita and Imai (1996b) proposed a shape-encoding mechanism that depends on genetic codes from the loops' phenotypical pattern to self-replication. Afterward, there were many studies in two-dimensional (Morita and Imai, 1996a) or three-dimensional reversible cellular space (Imai et al., 2002). In addition to self-replication, interacting between different loops has been conjectured, including self-protection with shielding, deflecting, and poisoning (Sayama, 2004), settling collisions with inroad, counter, defensive, and cancel methods (Suzuki and Ikegami, 2003). Such actions always harm the right of others to live.

All the aforementioned self-replicating models are based on synchronous CAs, in which all the cells are iterated to undergo state transitions simultaneously at every discrete time step. In nature, living systems are characterized by asynchronous timing modes, whereby studying self-replication on asynchronous cellular automata (ACAs) turns out to be crucial for a deeper understanding of the underlying mechanisms Huang et al. (2013). In an ACA,

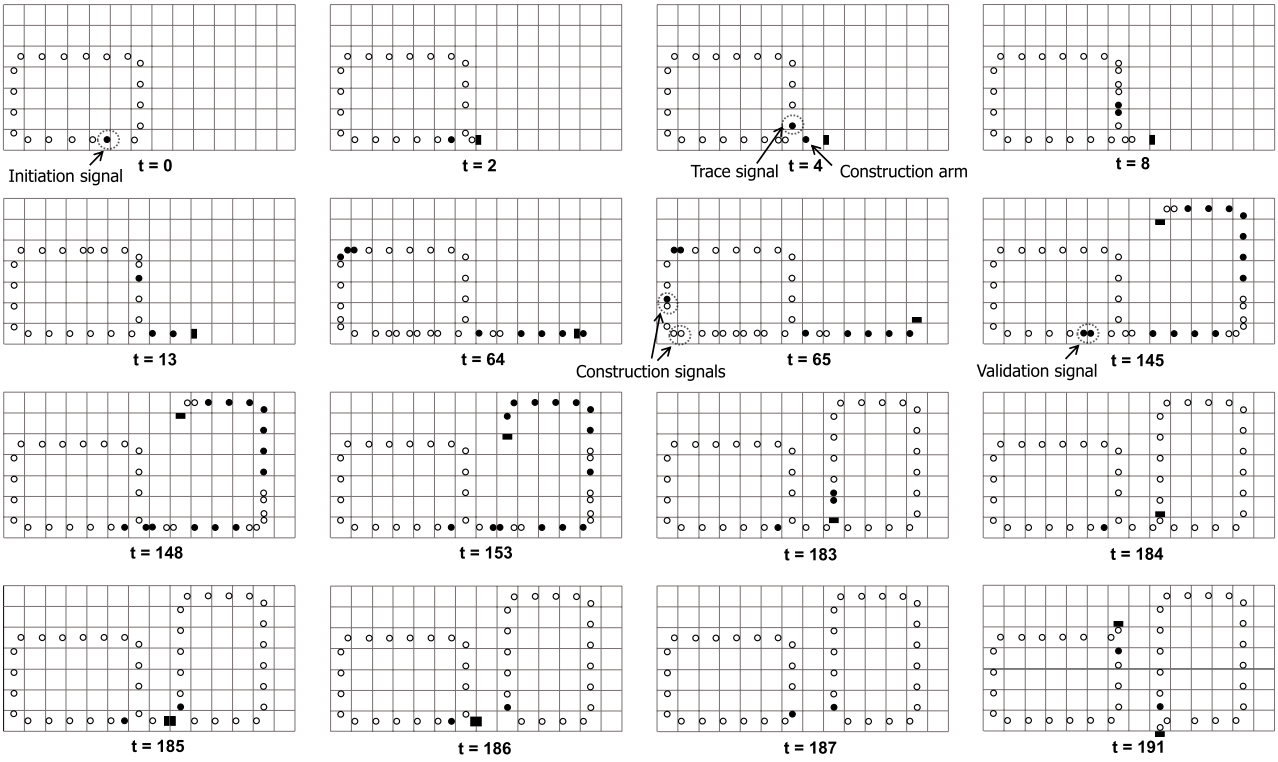


FIGURE 2
The normal process of self-replicating.

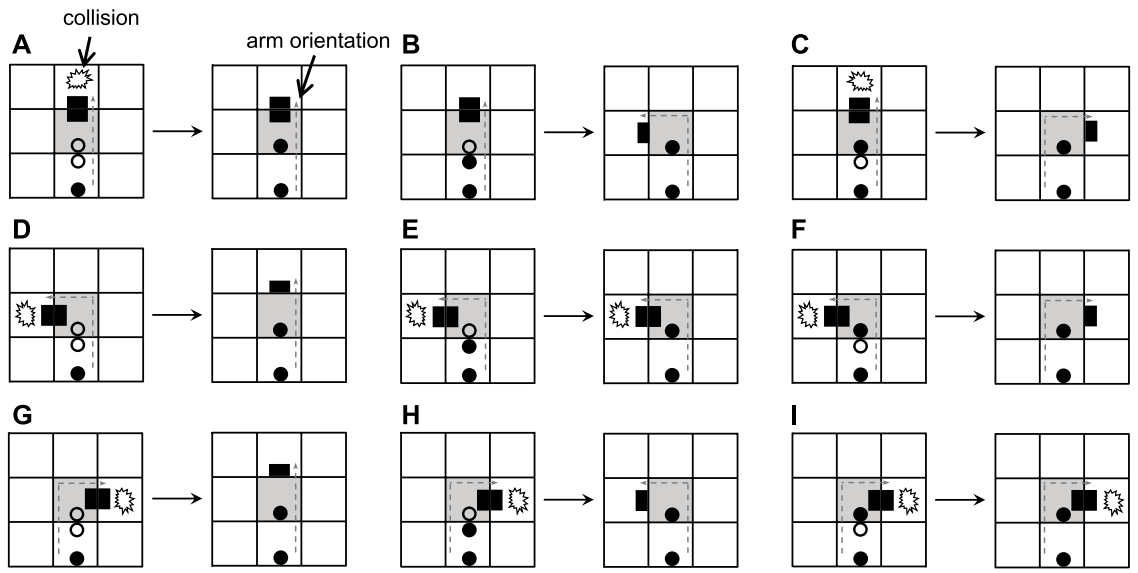


FIGURE 3
Transition rules of the greedy selection mechanism.

cells are updated at random timings independently from other cells, not needing a central clock signal to be distributed to all cells at any time. On the other hand, the unpredictable updating order of cells tends to bring more difficulty into the construction and self-reproduction on ACAs than on synchronous CAs. Nevertheless, Takada et al. (2007) designed a self-replicating loop based on the self-timed cellular automaton, which can self-reproduce parallelly and cope with the deadlock caused by collisions between self-replicating loops due to the asynchronous updating sequence. Especially, they used a simple mechanism that permits two colliding arms to fall back simultaneously. Huang et al. (2013) endowed a self-adaptive ability to the model, which allows two loops to not retract their arms but continue to accomplish self-replication when a collision occurs on occasion. In this case, the dead head will wait for a construction signal that can move the head into a direction away from the collision. More specifically, the choice of using which signal is made locally at the moment when the end of the constructing arm runs into an obstacle, and hence, such a selection is greedy. As a result, the passive self-adaptation can work in many situations where the normal reproduction of a loop is disturbed by some external constrain, thereby enabling the loop to survive and reproduce in a wide variety of regions (Huang et al., 2013).

3 Materials and methods

3.1 Self-timed cellular automata

Our self-replicating loops are implemented on a self-timed cellular automaton (Peper et al., 2002; Takada et al., 2007), which comprises of a two-dimensional asynchronous cellular array of identical cells. Each cell is partitioned into

TABLE 1 The list of functions about various signals.

Name	Pattern	Function
Initiation signal	● ♯	Initiate self-replicating
Trace signal	♯ ●	Trace the shape of a mother loop
Validation signal	● ●	Validate whether the offspring and construction signals are replicated successfully
Construction signals	◦ ◦	Advance construction arm straight forward
	◦ ●	Advance construction arm leftward
	● ◦	Advance construction arm rightward

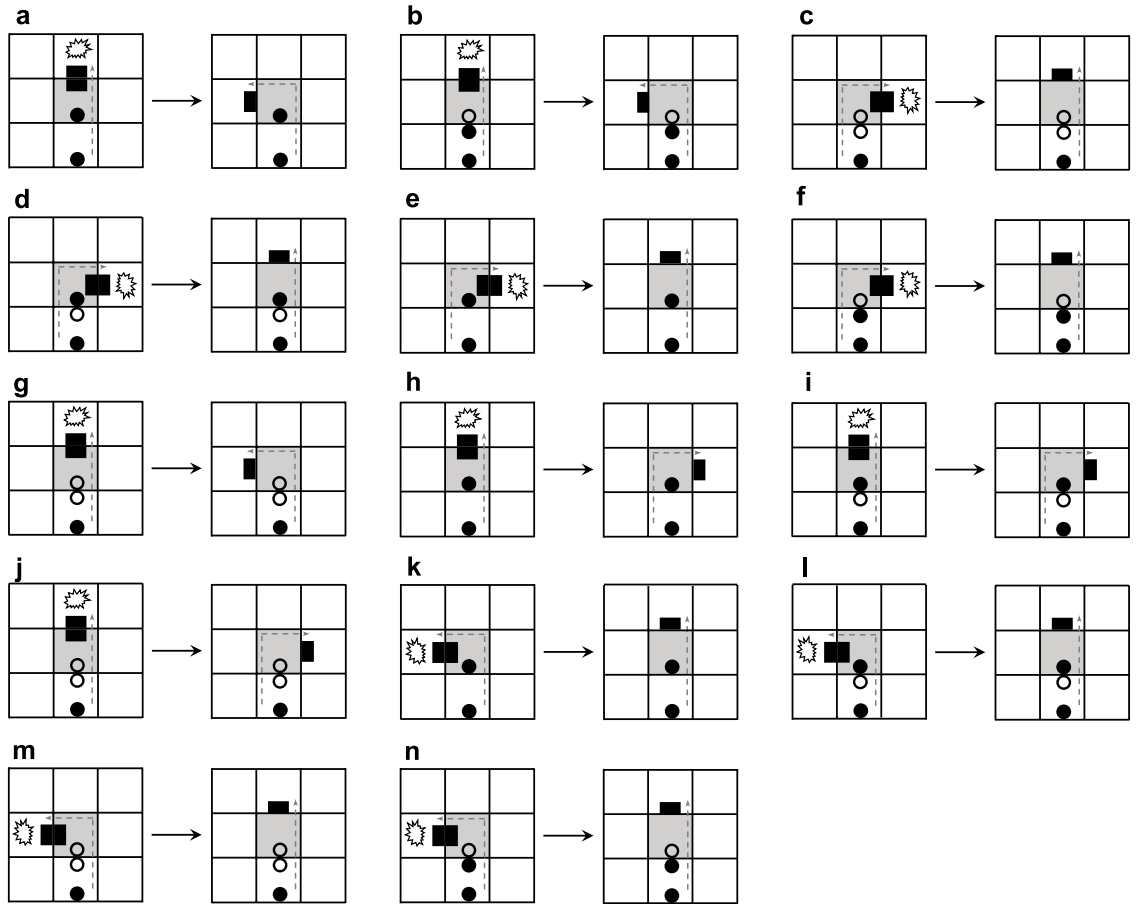


FIGURE 4
Transition rules of the adding mechanism.

four parts in a one-to-one correspondence with its neighboring cells, and each part has a state taken from a finite set of states at a time. Thus, a STCA may be deemed to a partitioned cellular automaton (Imai et al., 2002). Each cell undergoes transitions according to a transition function f that operates on the four parts of the cell and the nearest part of its four neighbors. The transition function f is defined as follows:

$$f(n, w, s, e, s_1, e_1, n_1, w_1) = (n', w', s', e', s_1', e_1', n_1', w_1'), \quad (1)$$

where each value in parentheses denotes the new state of a partition after updating (see Figure 1).

Also, transition rules of an STCA are rotation symmetric, such that rotating both the left-hand side and the right-hand side of a rule in a multiple of 90° simultaneously give rise to equivalent rules of the original one. The transitions of cells in an STCA occur randomly and are independent of each other, i.e., an ACA. Because the update of a cell may change the nearest sub-cells of its neighboring cells, to prevent a write–conflict situation from occurring, we assume that all neighboring cells never undergo transitions at the same time. To this end, an effective scheme that can be used to iterate the STCA's global transition is called random choice, by which at a time, only one cell is randomly selected with uniform probability to undergo a transition.

3.2 Self-replicating loops with active self-adaptability

Different from sheathed self-replicating loops in Suzuki and Ikegami (2003), a self-replicating loop implemented on our STCA model is unsheathed and needs the same number of states as the passive model in Huang et al. (2013). Four-cell states are used for each part of any cell, denoted by ♯, ◦, • and ■, respectively. The state ♯ is often shown blank in the figures for convenience. A cell is quiescent if all of its four sub-cells are in the state ♯. Transition rules are listed in

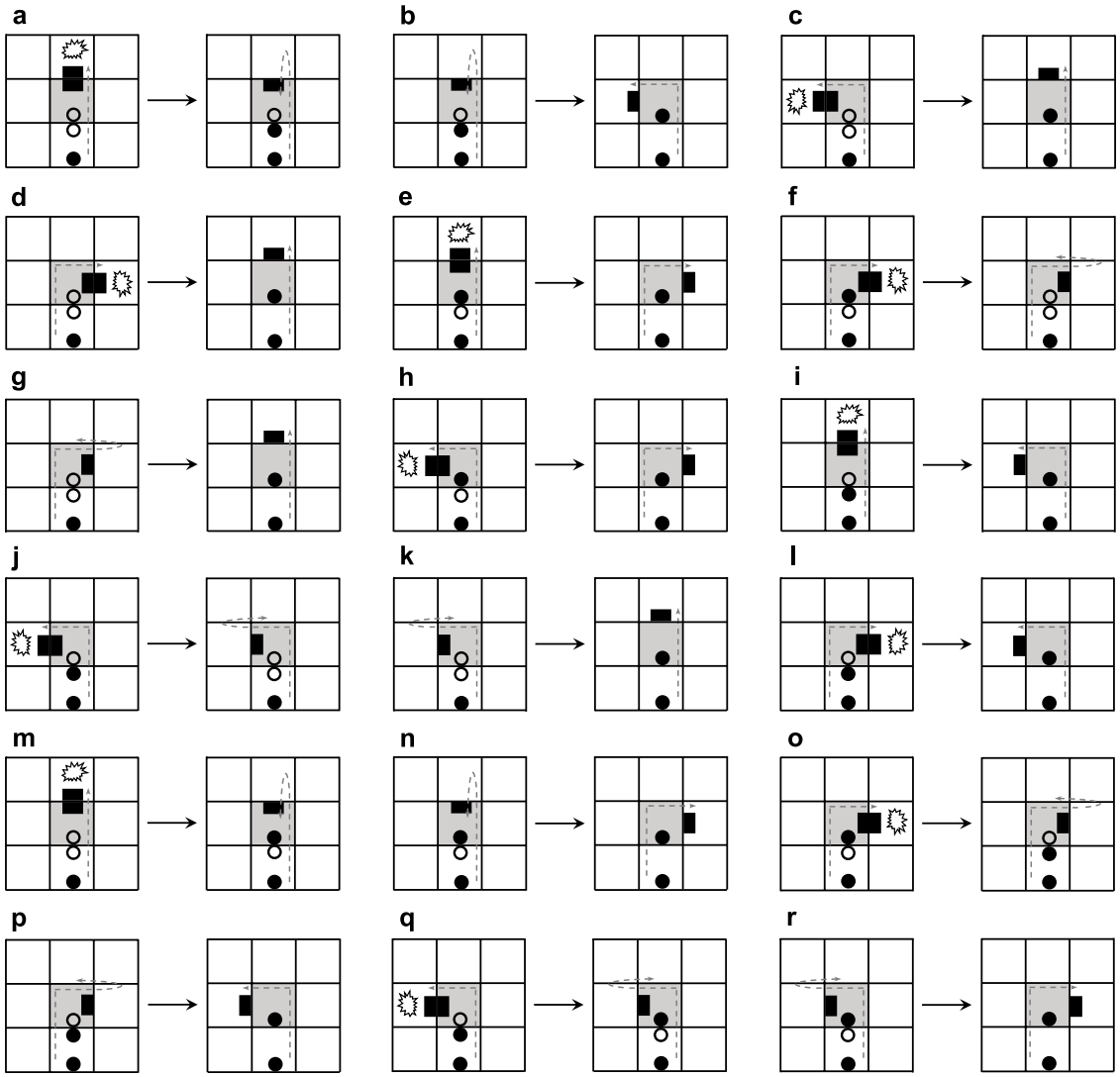


FIGURE 5
Transition rules of the changing mechanism.

Supplementary Appendix A, excluding the rotational symmetry equivalents.

3.2.1 Normal self-replicating based on shape-encoding mechanism

When enough space is left, a loop can normally replicate itself in the cell region. Several signals listed in Table 1 are used to fulfill the self-replication according to the shape-encoding mechanism.

Figure 2 illustrates a typical self-replicating process of a loop, which is similar to Huang et al. (2013). An initiation signal will transmit counterclockwise before the replication starts. When the initiation signal arrives at a left-turn corner of the loop, it generates an initial construct arm stretching out from the corner, as well as an inspection head to trace the shape of the mother loop. The inspection head •• will sequentially encodes each cell into an appropriate construction signals including going straight, turning right, and turning left. The signals from the mother loop are continuously transmitted to the head of the construct arm and are decoded into the corresponding part. Moreover, as soon as the shape-encoding process finishes, a validation signal is generated to verify whether the sub loop is constructed. If self-replicating succeeds, the signal will cut off the umbilical cord between the mother and the child, whereby both loops can start further replications individually.

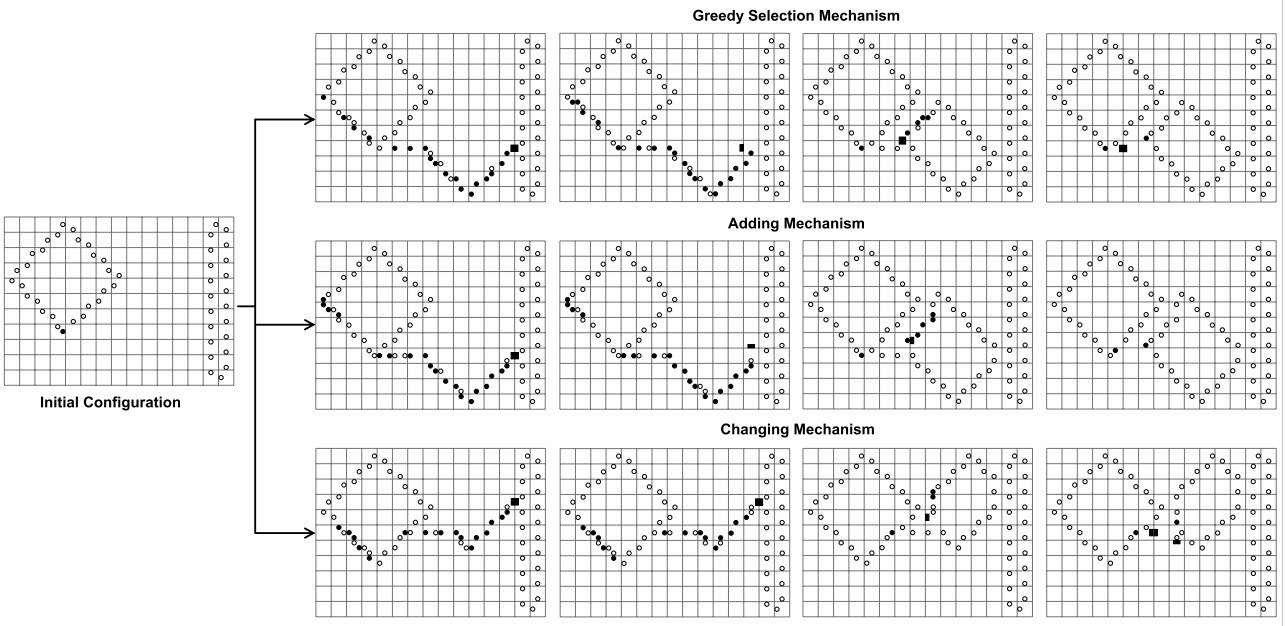


FIGURE 6
The different results of the greedy selection mechanism, adding mechanism, and changing mechanism starting from the same initial configuration where normal replication is limited by space.

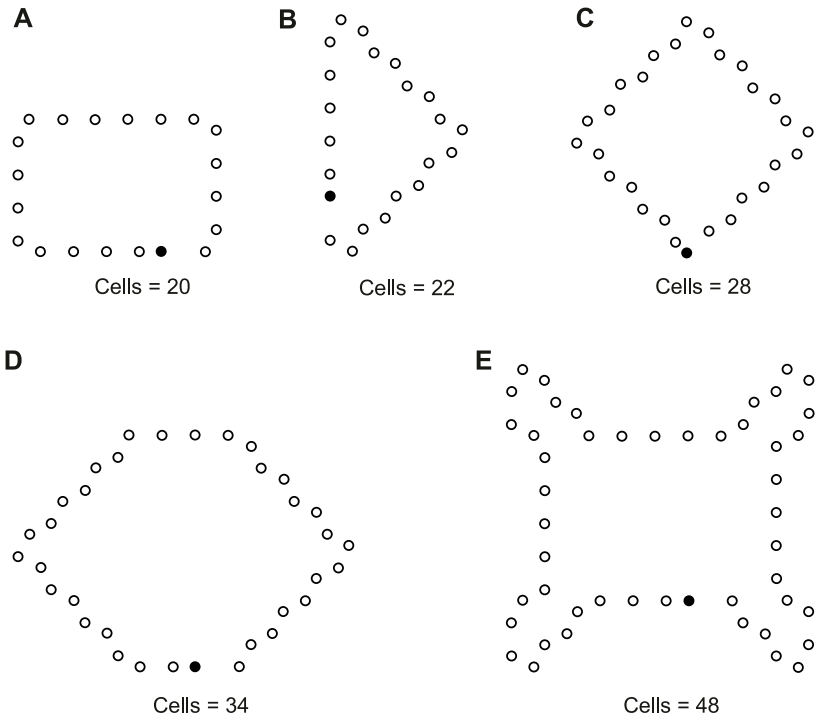


FIGURE 7
Different initial configurations.

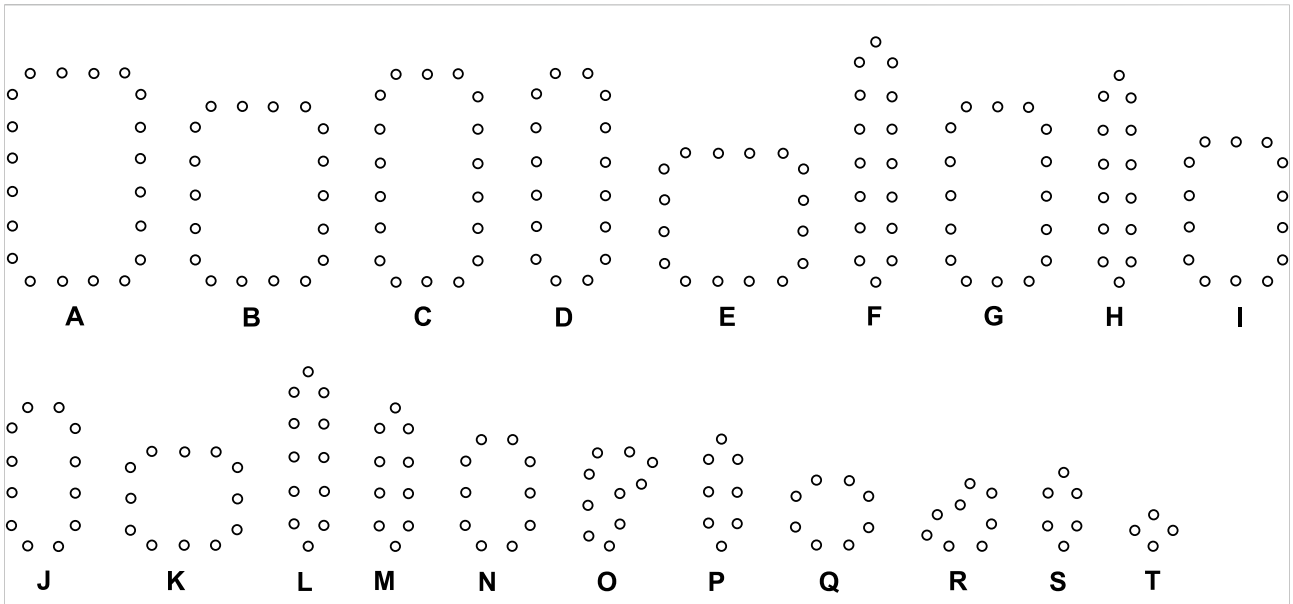


FIGURE 8
All final loop structures starting from the configuration in Figure 7A by the greedy selection, adding, and changing mechanisms.

TABLE 2 Statistical numbers of the loops with various structures for the greedy selection mechanism on different cellular spaces starting from the initial configuration in Figure 7A.

Loop Size	Shape\Amount\Space	60*60	80*65	100*65	Loop Size	Shape\Amount\Space	60*60	80*65	100*65
20 cells	Figure 8A	55	63	104	10 cells	Figure 8M	1	42	1
16 cells	Figure 8D	0	1	1		Figure 8N	2	4	20
	Figure 8E	13	7	2	8 cells	Figure 8P	4	0	3
14 cells	Figure 8H	0	13	0		Figure 8Q	3	5	11
	Figure 8I	6	6	3	6 cells	Figure 8S	2	33	16
12 cells	Figure 8J	9	6	9	4 cells	Figure 8T	2	19	0
	Figure 8K	2	0	0					
	Value of H	0.68547	0.83363	0.59182					

TABLE 3 Statistical numbers of the loops with various structures for the adding mechanism on different cellular spaces starting from the initial configuration in Figure 7A.

Loop Size	Shape\Amount\Space	60*60	80*65	100*65	Loop Size	Shape\Amount\Space	60*60	80*65	100*65
20 cells	Figure 8A	38	44	49	10 cells	Figure 8N	0	1	0
18 cells	Figure 8B	11	10	12		Figure 8O	0	0	2
	Figure 8C	0	0	1	8 cells	Figure 8P	12	18	19
16 cells	Figure 8F	1	1	1		Figure 8R	1	2	5
	Figure 8G	2	1	0	6 cells	Figure 8S	63	163	212
12 cells	Figure 8L	0	5	2	4 cells	Figure 8T	58	82	103
10 cells	Figure 8M	17	22	41					
	Value of H	0.72329	0.66301	0.65614					

TABLE 4 Statistical numbers of the loops with various structures for the changing mechanism on different cellular spaces starting from the initial configuration in Figure 7A.

Loop Size	Shape\Amount\Space	60*60	80*65	100*65	Loop Size	Shape\Amount\Space	60*60	80*65	100*65
20 cells	Figure 8A	32	35	36	12 cells	Figure 8K	0	1	0
18 cells	Figure 8B	0	0	1		Figure 8L	3	0	3
16 cells	Figure 8D	2	0	3	10 cells	Figure 8M	4	1	9
	Figure 8E	0	1	6		Figure 8N	2	2	4
	Figure 8F	0	1	0	8 cells	Figure 8P	32	6	54
	Figure 8G	11	12	15		Figure 8Q	11	12	20
14 cells	Figure 8I	2	3	5	6 cells	Figure 8S	29	115	137
12 cells	Figure 8J	23	14	22	4 cells	Figure 8T	34	139	77
	Value of H	0.91211	0.66984	0.85149					

TABLE 5 Statistical numbers of the loops with various structures for the greedy selection mechanism on different cellular spaces starting from the initial configuration in Figure 7B.

Loop Size	Shape\Amount\Space	60*60	80*65	85*65	Loop Size	Shape\Amount\Space	60*60	80*65	85*65
22 cells	Figure 9A	66	36	60	8 cells	Figure 9Z	8	108	3
16 cells	Figure 9E	7	0	2	6 cells	Figure 9AC	0	0	5
10 cells	Figure 9U	8	0	47	4 cells	Figure 9AD	29	52	51
	Value of H	0.52218	0.43068	0.57119					

TABLE 6 Statistical numbers of the loops with various structures for the adding mechanism on different cellular spaces starting from the initial configuration in Figure 7B.

Loop Size	Shape\Amount\Space	60*60	80*65	85*65	Loop Size	Shape\Amount\Space	60*60	80*65	85*65
22 cells	Figure 9A	38	52	43	12 cells	Figure 9M	3	0	0
20 cells	Figure 9B	0	4	1		Figure 9N	35	0	1
18 cells	Figure 9C	0	1	1		Figure 9O	0	1	5
16 cells	Figure 9E	0	1	0		Figure 9P	0	0	3
	Figure 9F	0	1	0	10 cells	Figure 9V	1	0	26
	Figure 9G	0	0	1		Figure 9W	4	0	0
14 cells	Figure 9I	1	0	1	8 cells	Figure 9Z	1	16	2
	Figure 9J	2	1	0		Figure 9A	6	0	8
	Figure 9K	1	0	0	6 cells	Figure 9AC	53	14	18
	Figure 9L	0	1	0	4 cells	Figure 9AD	103	106	142
	Value of H	0.69327	0.57124	0.62362					

3.2.2 Adaptive self-replication with mutations

What will happen if there is no extra space for normal self-replication of a loop or if the space is taken up by the arms of other loops? Huang et al. (2013) considered a greedy selection mechanism to deal with the situation, which means only useful information is retained during self-replication. And the details are shown in Figure 3. After a collision occurs, the construction arm's head becomes a dead head waiting for the construction signals coming from its mother. If the signal can work, then use it and change the direction of the construction arm. Otherwise,

TABLE 7 Statistical numbers of the loops with various structures for the changing mechanism on different cellular spaces starting from the initial configuration in Figure 7B.

Loop Size	Shape\Amount\Space	60*60	80*65	85*65	Loop Size	Shape\Amount\Space	60*60	80*65	85*65
22 cells	Figure 9A	47	55	41	10 cells	Figure 9V	1	0	0
16 cells	Figure 9H	1	1	0		Figure 9X	1	4	56
12 cells	Figure 9N	1	0	1		Figure 9Y	3	28	0
	Figure 9Q	2	5	35	8 cells	Figure 9Z	4	7	1
	Figure 9R	31	0	0		Figure 9AA	1	0	0
	Figure 9S	0	1	1		Figure 9AB	18	2	0
	Figure 9T	0	0	1	6 cells	Figure 9AC	9	6	13
10 cells	Figure 9U	1	0	1	4 cells	Figure 9AD	16	43	19
	Value of H	0.81213	0.71099	0.70811					

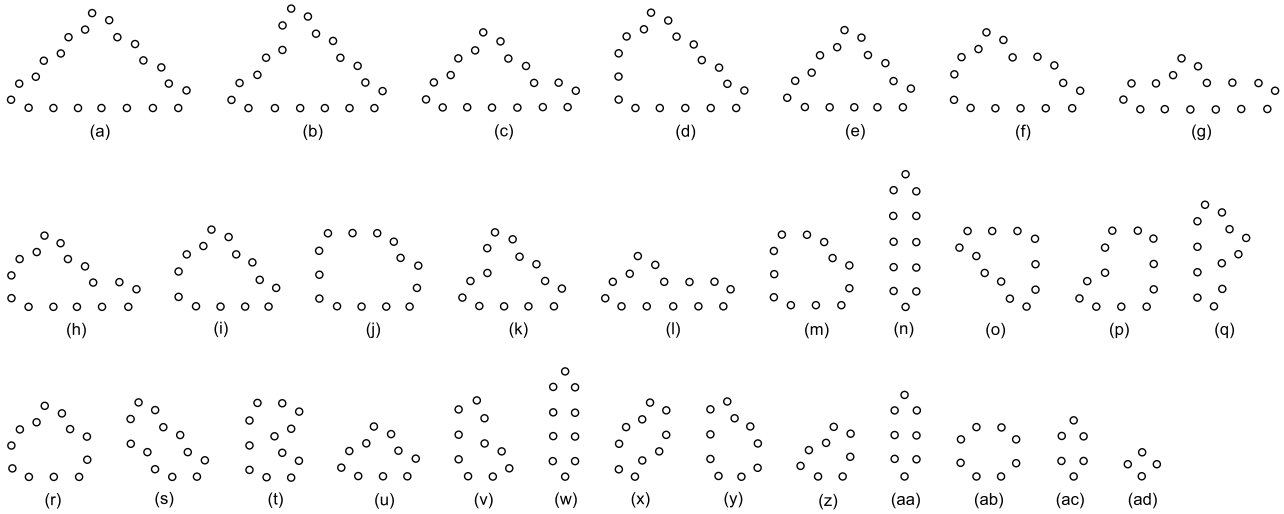


FIGURE 9
All final loop structures starting from the configuration in Figure 7B by the greedy selection, adding, and changing mechanisms.

TABLE 8 Statistical numbers of the loops with various structures for the greedy selection mechanism on different cellular spaces starting from the initial configuration in Figure 7C.

Loop Size	Shape\Amount\Space	60*60	80*80	100*100	Loop Size	Shape\Amount\Space	60*60	80*80	100*100
28 cells	Figure 10A	36	83	57	12 cells	Figure 10AA	59	2	104
24 cells	Figure 10C	0	6	25	8 cells	Figure 10AG	7	2	83
20 cells	Figure 10J	0	4	9	6 cells	Figure 10AI	0	0	5
16 cells	Figure 10Q	14	25	71	4 cells	Figure 10AJ	1	63	47
	Value of H	0.50862	0.55976	0.79217					

TABLE 9 Statistical numbers of the loops with various structures for the adding mechanism on different cellular spaces starting from the initial configuration in Figure 7C.

Loop Size	Shape\Amount\Space	60*60	80*80	100*100	Loop Size	Shape\Amount\Space	60*60	80*80	100*100
28 cells	Figure 10A	42	86	141	16 cells	Figure 10Q	2	0	0
26 cells	Figure 10B	0	1	4		Figure 10R	1	0	0
24 cells	Figure 10C	0	0	20		Figure 10S	1	0	0
	Figure 10D	14	0	0		Figure 10T	0	0	1
22 cells	Figure 10E	4	0	0	14 cells	Figure 10V	1	0	0
	Figure 10F	0	4	0	12 cells	Figure 10AA	0	2	0
	Figure 10G	0	0	2		Figure 10AB	1	1	0
	Figure 10H	0	0	3	10 cells	Figure 10AE	0	3	0
22 cells	Figure 10I	0	0	1		Figure 10AF	0	0	2
20 cells	Figure 10J	0	25	0	8 cells	Figure 10AG	5	1	1
	Figure 10K	0	1	0	6 cells	Figure 10AI	0	1	1
18 cells	Figure 10N	1	0	0	4 cells	Figure 10AJ	16	8	83
	Figure 10O	1	0	0					
	Value of H	0.71351	0.52249	0.50833					

TABLE 10 Statistical numbers of the loops with various structures for the changing mechanism on different cellular spaces starting from the initial configuration in Figure 7C.

Loop Size	Shape\Amount\Space	60*60	80*80	100*100	Loop Size	Shape\Amount\Space	60*60	80*80	100*100
28 cells	Figure 10A	43	72	138	12 cells	Figure 10AA	0	1	2
20 cells	Figure 10J	2	2	5		Figure 10AC	0	1	1
	Figure 10I	0	15	19		Figure 10AD	0	0	4
	Figure 10M	0	0	1	10 cells	Figure 10AF	1	69	0
18 cells	Figure 10P	19	3	0	8 cells	Figure 10AG	1	6	0
16 cells	Figure 10U	0	7	0		Figure 10AH	0	0	1
14 cells	Figure 10X	5	4	29	6 cells	Figure 10AI	3	1	26
	Figure 10Y	0	1	0	4 cells	Figure 10AJ	1	44	2
	Figure 10Z	0	1	0					
	Value of H	0.54088	0.74551	0.57765					

simply throw it away. Although such self-adaptation is simple and straightforward, it is passive and weak, resulting in much smaller child loops. In order to increase the adaptability and diversity of self-replicating models, we propose two novel mechanisms for active adaptation as follows:

Adding: add a different construction signal next to the head of the construction arm. For simplicity, the direction is directly changed at random.

Changing: change the construction signal following the head of the construction arm to other construction signals that are selected randomly.

Collisions are often inevitable due to the unpredictable nature of asynchronous updating. If the construction arm of a self-replicating loop perceives that the space is occupied, then it cannot extend furthermore and the state of the construction arm head will change from ♯■ to ■■ (called dead end). There are many situations when a collision occurs, such as an arm bumping into another loop's arm or an arm meeting the body of a loop.

Figure 4 elaborates the process of adding mechanisms for active adaptation. When the arm under going straight collides with an obstacle (Figures 4A,H), the current blocking state will be changed by randomly selecting one of the two orientations, namely turning left and turning right. Even a construction

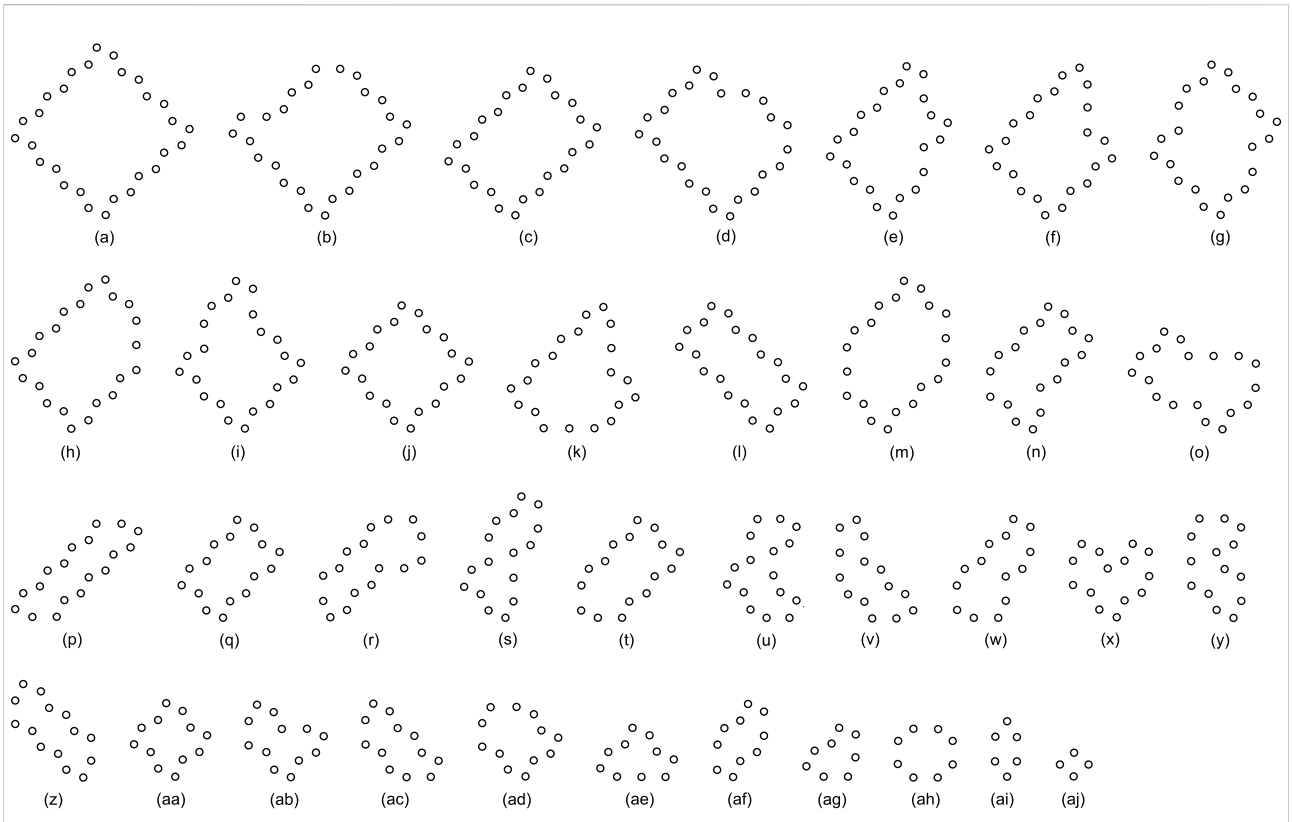


FIGURE 10
All final loop structures starting from the configuration in Figure 7C by the greedy selection, adding, and changing mechanisms.

signal behind the dead head is a straight-going signal; the mechanism will add a random direction (Figure 4G and Figure 4J). Especially if the construction signal behind the dead head is a left-turning signal, the dead head will turn left and become normal after going straight is blocked (Figure 4B). Similarly, if there is a right-turning signal, the head will turn right (Figure 4I). Whatever a construction signal is behind the dead head, if the head is blocked by turning left or right, then the head will go straight.

The content of the changing mechanism is presented in Figure 5. If an arm going straight meets an obstacle and the construction signal behind the dead head is a straight-going signal, then the straight-going signal will change to a left-turning signal (Figure 5A) or a right-turning signal (Figure 5M) and the head goes back. Such a state is not durable, and after which the arm will turn left (Figure 5B) or turn right (Figure 5N). If the construction signal behind the dead head can mitigate the collision, the original signal remains constant (Figures 5C–E, H, and I). When the arm is blocked to turn left and the construction signal following the dead head is a left-turning signal, the construction signal will randomly mutate to a right-turning signal (Figure 5Q) or straight-going signal (Figure 5J). Similarly, the aforementioned situation also happens on turning right.

We can see from Figure 6 that the greedy selection mechanism, adding mechanism, and changing mechanism can produce different sub-loops from the same initial configuration. Especially, the changing mechanism does not self-replicate at the beginning.

4 Experiments

In order to testify that active adaptation can produce more diversity of species than the previous passive adaptation, we set up various initial configurations and different boundary values to conduct the experiments. We used the trait distribution entropy from Sayama (2004) to characterize the diversity of the population, which shows as follows:

$$H = -\sum_i \left(\frac{n_i}{N} log \frac{n_i}{N}\right) = logN - \frac{1}{N}\sum_i (n_i * logn_i), \qquad (2)$$

where n_i is a quantity of loops that are made of i cells and N the number of loops in the current space. Moreover, the value of the

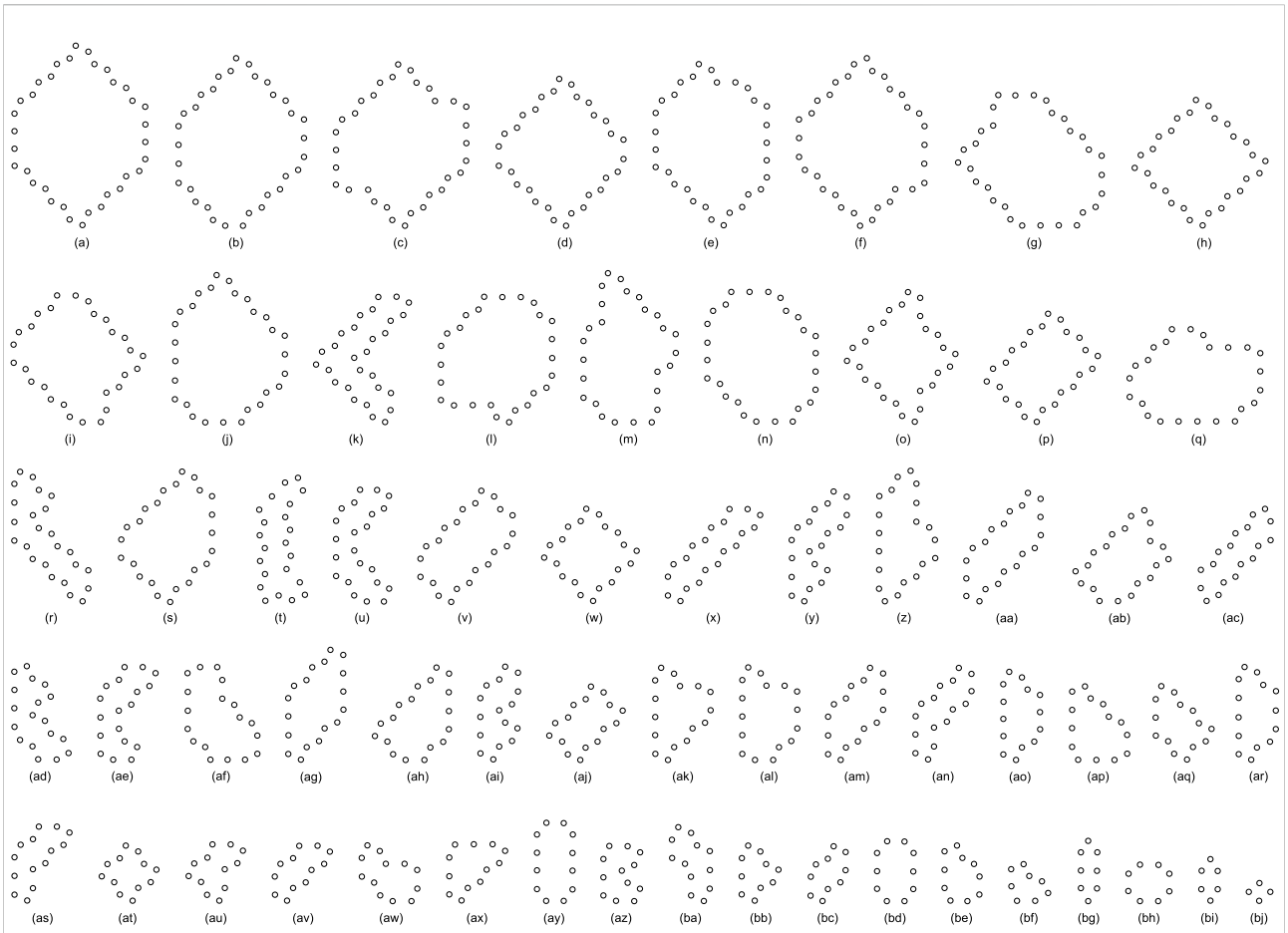


FIGURE 11
All final loop structures starting from the configuration in Figure 7D by the greedy selection, adding, and changing mechanisms.

trait distribution entropy ranges from 0 to *logN* and log function takes the logarithm base 10 instead of base *e*. $H = 0$ means that the space is filled with the same loop and $H = logN$ can be obtained when each loop in the current space differs from each other (i.e., the value of each n_i is 0 or 1 for all i). Especially, loops which posses different manifestations belong to different species even if the loops consist of the identical number of cells.

We use different initial configurations to do experiments as shown in Figure 7, in which, the first three are common shapes and the last two are irregular. For simplicity, all possible final structures of replicated sub-loops starting from the initial configuration in Figure 7A by either self-adaptation mechanism are listed in Figure 8. In addition, the quantities and distributions of each structure in the cellular spaces using greedy selection mechanism, adding mechanism, and changing mechanism are provided in Tables 2, 3, and 4, respectively. As a result, compared with the other two active mechanisms, the greedy selection (passive) mechanism has a highest value of H in 80*65 cellular space, because the space is not filled with one or two identical and abundant small loops. However, on the whole, the adding mechanism and changing mechanism have higher values of H than the greedy selection mechanism.

Likewise, Tables 5, 6, and 7 provide the self-replication results starting from the initial configuration in Figure 7B, along with all possible final sub-loops given in Figure 9. The value of H of the greedy selection mechanism is lower than that of adding mechanism and changing mechanism, which means that the adding mechanism and the changing mechanism can give rise to more diversity. Moreover, small loops appear later in the changing mechanism than in the adding mechanism, leaving more room for larger loops to self-replicate and bring more kinds of species. In addition, Tables 8, 9, and 10 demonstrate the results from the initial configuration in Figure 7C by each mechanism, in which the greedy selection mechanism can achieve the highest value of H in 100*100 cellular space. All possible loop structures are shown in Figure 10. Though the kinds of loops are the least for greedy selection mechanism, there is the maximum number of loops. Therefore, in the same biological environment, when the kinds of species are relatively small and the population is relatively large, the species also have a high diversity. Especially,

TABLE 11 Statistical numbers of the loops with various structures for the greedy selection mechanism on different cellular spaces starting from the initial configuration in Figure 7D.

Loop Size	Shape\Amount\Space	60*60	80*80	100*100	Loop Size	Shape\Amount\Space	60*60	80*80	100*100
34 cells	Figure 11A	30	34	43	12 cells	Figure 11AT	19	0	0
30 cells	Figure 11D	0	0	1		Figure 11AU	0	123	0
28 cells	Figure 11H	5	0	9	10 cells	Figure 11BB	4	0	0
24 cells	Figure 11P	0	0	1		Figure 11BC	0	0	249
20 cells	Figure 11W	0	2	0	8 cells	Figure 11BF	0	34	20
	Figure 11X	0	0	1		Figure 11BG	0	0	1
16 cells	Figure 11AI	0	1	22	6 cells	Figure 11BI	0	2	2
	Figure 11AJ	0	0	1	4 cells	Figure 11BJ	12	31	9
	Value of H	0.59562	0.55587	0.49319					

TABLE 12 Statistical numbers of the loops with various structures for the adding mechanism on different cellular spaces starting from the initial configuration in Figure 7D.

Loop Size	Shape\Amount\Space	60*60	80*80	100*100	Loop Size	Shape\Amount\Space	60*60	80*80	100*100
34 cells	Figure 11A	32	47	86	14 cells	Figure 11AO	0	1	0
32 cells	Figure 11B	2	1	0		Figure 11AP	0	1	0
	Figure 11C	0	0	1		Figure 11AQ	0	0	7
30 cells	Figure 11E	1	0	1		Figure 11AR	0	0	1
	Figure 11F	0	0	2	12 cells	Figure 11AT	0	4	0
	Figure 11G	0	0	1		Figure 11AV	1	0	1
28 cells	Figure 11J	4	0	2		Figure 11AW	1	0	0
	Figure 11K	0	1	0		Figure 11AX	0	20	0
26 cells	Figure 11L	0	0	1		Figure 11AY	0	1	0
	Figure 11M	0	0	1		Figure 11AZ	0	2	0
24 cells	Figure 11Q	1	0	0		Figure 11AB	0	0	1
	Figure 11R	0	1	1	10 cells	Figure 11BC	0	1	0
20 cells	Figure 11Y	0	1	0		Figure 11BD	1	0	0
	Figure 11Z	0	0	1		Figure 11BE	0	3	0
	Figure 11AA	0	0	3	8 cells	Figure 11BF	5	3	17
18 cells	Figure 11AC	1	0	0		Figure 11BG	3	0	1
	Figure 11AD	0	0	4		Figure 11BH	0	1	0
16 cells	Figure 11AK	0	9	0	6 cells	Figure 11BI	5	51	29
	Figure 11AL	0	1	0	4 cells	Figure 11BJ	4	33	30
	Figure 11AM	0	0	1					
	Value of H	0.76886	0.85201	0.79910					

the adding mechanism can produce many loops with complete quantity and different sizes.

All replicating results of the loop structures from the configuration in Figure 7D are given in Figure 11. In this case, the values of H using the adding mechanism and the changing mechanism in Tables 12, 13, respectively are obviously higher than that of the greedy selection mechanism in Table 11. Furthermore, self-replications starting from the irregular and symmetric shapes in Figure 7E are elaborated in Tables 14, 15, and 16 with various types of sub-loops shown in Figure 12. It can be verified that the loop that is the same as the initial configuration quickly takes up the entire space, leaving

TABLE 13 Statistical numbers of the loops with various structures for the changing mechanism on different cellular spaces starting from the initial configuration in Figure 7D.

Loop Size	Shape\Amount\Space	60*60	80*80	100*100	Loop Size	Shape\Amount\Space	60*60	80*80	100*100
34 cells	Figure 11A	26	40	64	16 cells	Figure 11AN	0	0	1
26 cells	Figure 11N	0	0	1	14 cells	Figure 11AS	0	3	0
	Figure 11O	0	0	3	12 cells	Figure 11AU	13	39	2
24 cells	Figure 11S	0	0	2		Figure 11AV	0	0	3
22 cells	Figure 11T	2	22	1	10 cells	Figure 11BB	31	13	77
	Figure 11U	2	0	0		Figure 11BC	10	0	0
	Figure 11V	0	0	1		Figure 11BE	0	1	0
20 cells	Figure 11AB	1	0	0	8 cells	Figure 11BF	0	20	2
18 cells	Figure 11AE	1	0	0		Figure 11BG	0	35	2
	Figure 11AF	0	2	0	6 cells	Figure 11BI	6	5	32
	Figure 11AG	0	0	1	4 cells	Figure 11BJ	3	14	187
	Figure 11AH	0	0	1					
	Value of H	0.76923	0.88685	0.63472					

TABLE 14 Statistical numbers of the loops with various structures for the greedy selection mechanism on different cellular spaces starting from the initial configuration in Figure 7E.

Loop Size	Shape\Amount\Space	60*46	80*65	85*65	Loop Size	Shape\Amount\Space	60*46	80*65	85*65
48 cells	Figure 12A	19	27	22	14 cells	Figure 12W	0	0	4
46 cells	Figure 12B	1	0	0	10 cells	Figure 12AD	9	10	4
28 cells	Figure 12E	1	0	0	8 cells	Figure 12AH	2	4	0
22 cells	Figure 12J	0	0	6	6 cells	Figure 12AK	0	0	25
16 cells	Figure 12R	0	4	0	4 cells	Figure 12AL	1	8	0
	Figure 12S	0	0	1					
	Value of H	0.50377	0.57922	0.59937					

TABLE 15 Statistical numbers of the loops with various structures for the adding mechanism on different cellular spaces starting from the initial configuration in Figure 7E.

Loop Size	Shape\Amount\Space	60*46	80*65	85*65	Loop Size	Shape\Amount\Space	60*46	80*65	85*65
48 cells	Figure 12A	12	15	18	14 cells	Figure 12X	3	1	8
40 cells	Figure 12C	2	0	0	12 cells	Figure 12Z	0	2	3
34 cells	Figure 12D	0	2	2		Figure 12AA	0	2	1
24 cells	Figure 12G	1	0	0	10 cells	Figure 12AD	3	4	34
	Figure 12H	0	0	2		Figure 12AE	0	2	2
	Figure 12I	0	0	1	8 cells	Figure 12AH	0	0	1
20 cells	Figure 12K	1	0	0		Figure 12AI	3	1	0
	Figure 12L	0	0	11		Figure 12AJ	69	108	12
18 cells	Figure 12N	2	0	0	6 cells	Figure 12AK	5	18	6
	Figure 12O	0	1	0	4 cells	Figure 12AL	12	21	53
16 cells	Figure 12T	0	1	0					
	Value of H	0.62145	0.60756	0.85253					

TABLE 16 Statistical numbers of the loops with various structures for the changing mechanism on different cellular spaces starting from the initial configuration in Figure 7E.

Loop Size	Shape\Amount\Space	60*46	80*65	85*65	Loop Size	Shape\Amount\Space	60*46	80*65	85*65
48 cells	Figure 12A	11	16	12	12 cells	Figure 12AB	2	0	0
26 cells	Figure 12F	0	0	1		Figure 12AC	0	1	0
24 cells	Figure 12G	0	0	1	10 cells	Figure 12AE	1	0	1
20 cells	Figure 12M	0	1	0		Figure 12AF	11	47	50
18 cells	Figure 12P	5	0	0		Figure 12AG	0	1	0
	Figure 12Q	0	4	0	8 cells	Figure 12AI	0	1	0
16 cells	Figure 12U	24	0	1		Figure 12AJ	5	0	60
	Figure 12V	0	1	0	6 cells	Figure 12AK	11	44	4
14 cells	Figure 12X	4	6	10	4 cells	Figure 12AL	14	11	48
	Figure 12Y	0	0	2					
	Value of H	0.88156	0.70506	0.70877					

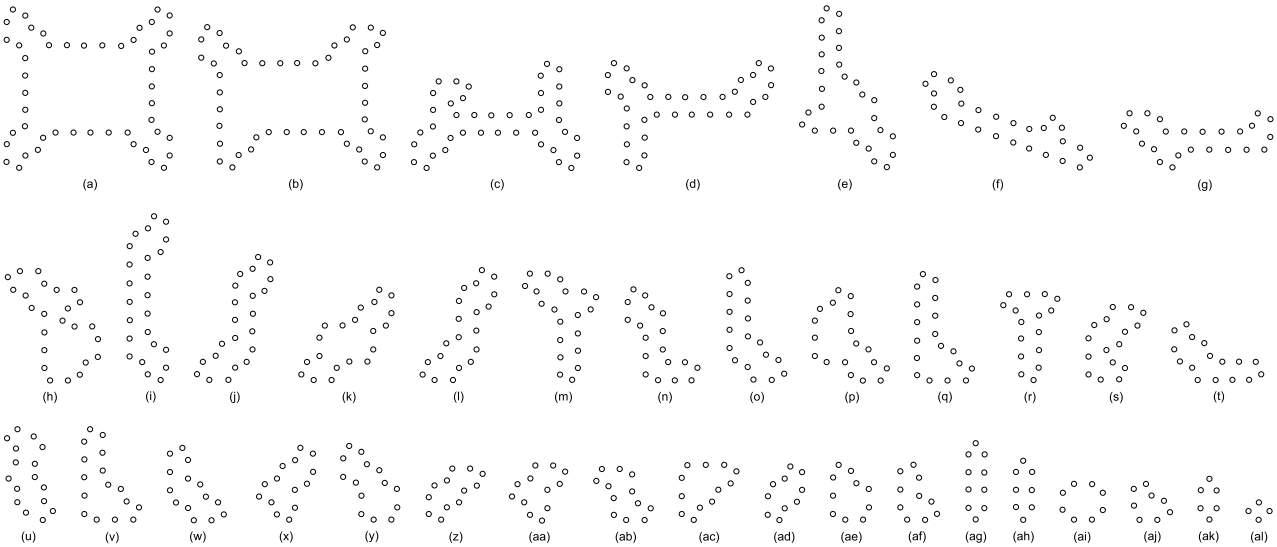


FIGURE 12
All final loop structures starting from the configuration in Figure 7E by the greedy selection, adding, and changing mechanisms.

little room for the smaller ones, which creates a smaller population of loops and owns the lowest diversity of species.

Therefore, the aforementioned experiments show that the adding mechanism and the changing mechanism can bring higher diversity than the greedy selection mechanism. Moreover, for those loops with the same number of cells, the adding mechanism and the changing mechanism can obtain more variable loops with different phenotypes. Phenotype change is a sufficient factor for achieving such a functional evolution Kampis and Gulyás (2008). In the process of self-replicating, once a minimal loop is created, the loop will quickly replicate itself, because the minimal loop can track its body much faster. As a result, the minimal loops will become the vast majority of the population after reaching saturation, thereby reducing the diversity. Such a tendency is similar to the basic orientation of the evolution paths in Sayama (2004).

Moreover, in order to further test the diversity that the active mechanisms can bring, we conducted experiments on the initial configuration in 7(d) with 60*60 cellular space using three mechanisms. From Figure 13, we can see that the greedy

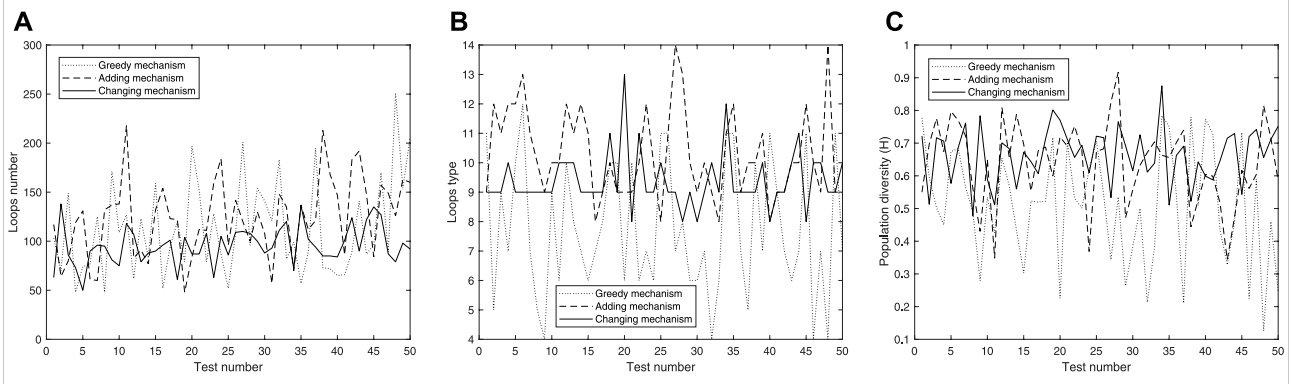


FIGURE 13
Further results on the initial configuration in Figure 7D with 60*60 cellular space using the three mechanisms.

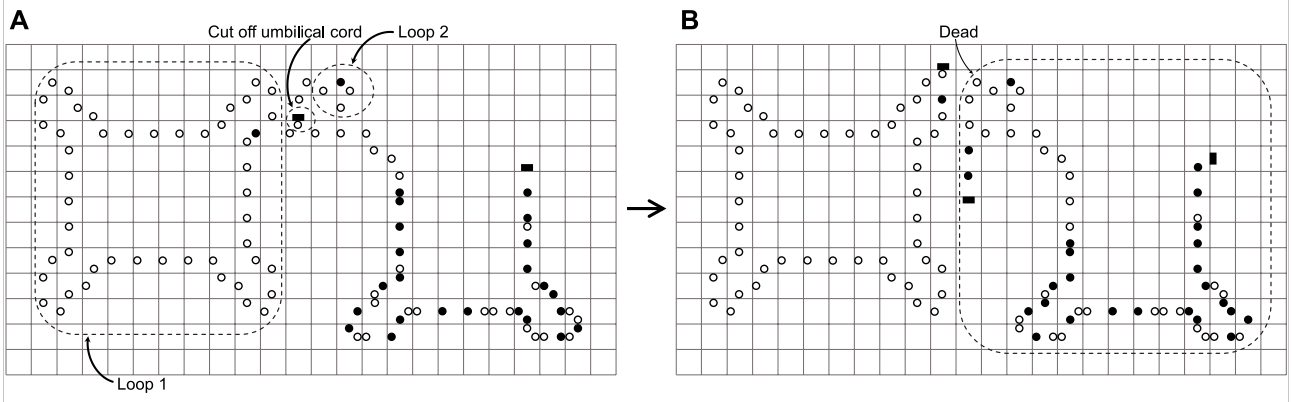


FIGURE 14
A dead loop caused by improperly cutting off an umbilical cord.

mechanism mostly can obtain the highest value on the total quantity of loops, but significantly lower than the active mechanisms in terms of species and diversity, which may imply that the greedy mechanism tends to produce smaller loops. Generally speaking, smaller loops can replicate themselves rapidly and be more likely to survive.

However, mistakes may occur in the process of self-replication and the details are shown in Figure 14. There are several conditions for the error to occur (see also Huang et al. (2013)): 1) Loop 1 is on the inner side of the arm of the loop 2 in Figure 14A; 2) The arm of loop 1 contains no construction code, which means the head of the arm is in the state ◦■; 3)The construction arm of loop 2 has been scanned by a validation signal, which means the state about the part of the arm turns state ● to state ◦. Especially, there is a parallel arm that is made up of state ◦ shown in Figure 14B. However, this error seldom happens. Under these conditions, loop 2 may have an erroneous cognition that it thinks of the arm of loop 2 as its own; thereby it will cut off the umbilical cord at the arm head. Fortunately, loop 1 is unaffected by this error and goes on self-replicating. Loop 2, however, is not so lucky, and dies. What is worse, the dead loop 2 and the discarded arm of loop 1 waste many spaces. Nevertheless, enhancing the function of a validation signal may seem reasonable to avoid erroneous cognition. On the plus side, an erroneous cognition may possibly be regarded as some non-trivial co-action between loops Sayama (1999). Moreover, an erroneous cognition may create an offspring the size of which is bigger than the mother loop Salzberg (2003).

Furthermore, from Figure 15, we can see that Loop 2 takes up the space thanks to the faster replication capability during the process of generating Loop 1, and Loop 1 exactly forms a closed

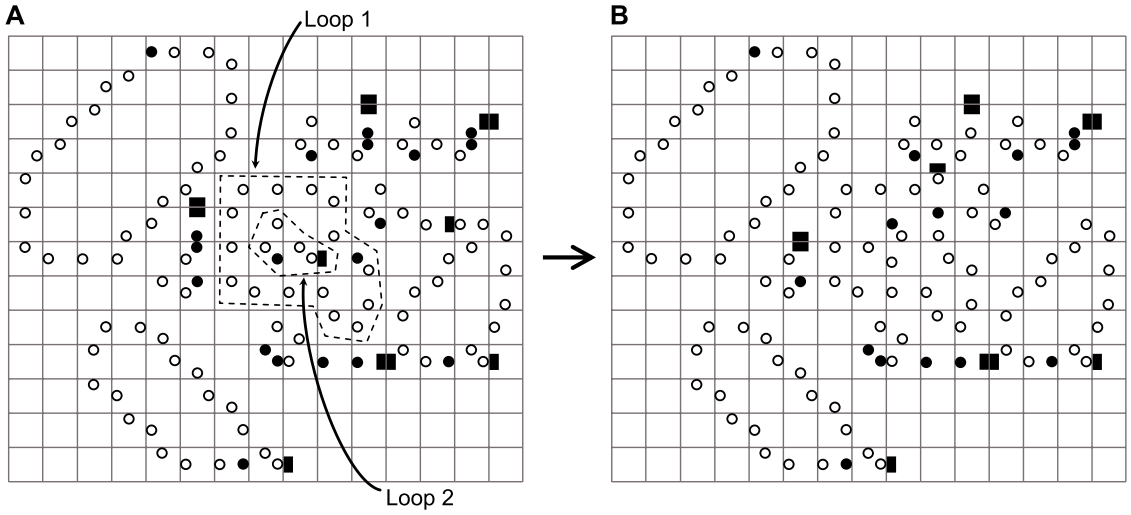


FIGURE 15
A phagocytosis situation because of different self-replicating speeds.

loop that wraps around Loop 2. This situation is similar to the phagocytosis of immune cell Stossel (1974). Luckily, Loop 1 and Loop 2 are alive. Thus, if there are enough spaces, the loops can self-replicate.

5 Discussion

Many studies have considered the self-replication on various cellular automata to simulate the process of biological self-replication, including the reversible cellular automata (Morita and Imai, 1996b), polymorphic cellular automata (Sekanina and Komenda, 2011), and graph automata (Tomita et al., 2002). Moreover, self-replication on cellular automata has been applied to several fields, such as worm propagation in smartphones (Peng et al., 2013), artificial chemistry (Hutton, 2007), and image processing (Sahin et al., 2015). In this article, we provided a different approach to enhance the diversity of artificial self-replicating structures, instead of abandoning partial structural information or destroying the whole loop. In order to obtain these effects better, on the basis of existing ordinary self-replication, we change a greedy selection mechanism to two active mechanisms when dealing with collision, which add an orientation and change the construction signal under the dead head. Experiments showed that active adaptations using our schemes can actually improve the possibility of survival and replication of any self-replicating structure in a wide variety of environments than the passive one. In particular, the changing mechanism involves abandoning one building-block from the original structure of a mother loop when every collision happens, even though the mechanism changes the construction signal. Also, the adding mechanism does not seem to lose the block of information coming from the parent, while some constructional information is left for the offspring to complete the replication. This may result in the shrinkage of both shape and size of the offspring.

Although the adding and changing mechanisms enable more active self-adaptation than the greedy selection mechanism, they still look somewhat passive in the sense that the adaptation can only be activated when collision occurs. In living organisms, mutation on genes will occur in a probabilistic manner. As with self-adaptation, self-recovery or self-healing is also an interesting feature of organisms. In the future work, we will consider how to endow self-replicating loops with a self-repairing ability (Tempesti et al., 1998), use random inputs (Griffith et al., 2005) to generate interesting patterns, and genetic algorithms to automatically discover rules (Lohn and Reggia, 1997).

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material; further inquiries can be directed to the corresponding author.

Author contributions

WX: methodology and writing—original draft; CW: conceptualization and writing—original draft; QP: software

and formal analysis; JL: conceptualization and supervision; YX: methodology and validation; and SK: formal analysis.

Acknowledgments

The authors are grateful to the reviewers for their careful reading and valuable comments.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2022.958069/full#supplementary-material

References

Abou-Jaoudé, W., Traynard, P., Monteiro, P. T., Saez-Rodriguez, J., Helikar, T., Thieffry, D., et al. (2016). Logical modeling and dynamical analysis of cellular networks. *Front. Genet.* 7, 94. doi:10.3389/fgene.2016.00094

Agrawal, A. A. (2001). Phenotypic plasticity in the interactions and evolution of species. *Science* 294, 321–326. doi:10.1126/science.1060701

Aminetzach, Y. T., Macpherson, J. M., and Petrov, D. A. (2005). Pesticide resistance via transposition-mediated adaptive gene truncation in Drosophila. *Science* 309, 764–767. doi:10.1126/science.1112699

Anzalone, A. V., Randolph, P. B., Davis, J. R., Sousa, A. A., Koblan, L. W., Levy, J. M., et al. (2019). Search-and-replace genome editing without double-strand breaks or donor DNA. *Nature* 576, 149–157. doi:10.1038/s41586-019-1711-4

Bargmann, C. I., Hung, M.-C., and Weinberg, R. A. (1986). Multiple independent activations of the neu oncogene by a point mutation altering the transmembrane domain of p185. *Cell.* 45, 649–657. doi:10.1016/0092-8674(86)90779-8

Baris, Y., Taylor, M. R. G., Aria, V., and Yeeles, J. T. P. (2022). Fast and efficient dna replication with purified human proteins. *Nature* 606, 204–210. doi:10.1038/s41586-022-04759-1

Becerra-Rodríguez, C., Marsit, S., and Galeote, V. (2021). Diversity of oligopeptide transport in yeast and its impact on adaptation to winemaking conditions. *Front. Genet.* 11. doi:10.3389/fgene.2020.00602

Bilotta, E., and Pantano, P. (2006). Structural and functional growth in self-reproducing cellular automata. *Complexity* 11, 12–29. doi:10.1002/cplx.20131

Bookstein, R., Shew, J.-Y., Chen, P.-L., Scully, P., and Lee, W.-H. (1990). Suppression of tumorigenicity of human prostate carcinoma cells by replacing a mutated RB gene. *Science* 247, 712–715. doi:10.1126/science.2300823

Buckley, W. R., and Mukherjee, A. (2005). "Constructibility of signal-crossing solutions in von neumann 29-state cellular automata," in *International conference on computational science* (Atlanta, GA: Springer), 395–403.

Buisson, R., Langenbucher, A., Bowen, D., Kwan, E. E., Benes, C. H., Zou, L., et al. (2019). Passenger hotspot mutations in cancer driven by apobec3a and mesoscale genomic features. *Science* 364, eaaw2872. doi:10.1126/science.aaw2872

Byl, J. (1989). Self-reproduction in small cellular automata. *Phys. D. Nonlinear Phenom.* 34, 295–299. doi:10.1016/0167-2789(89)90242-x

Cea, V., Cipolla, L., and Sabbioneda, S. (2015). Replication of structured dna and its implication in epigenetic stability. *Front. Genet.* 6, 209. doi:10.3389/fgene.2015.00209

Codd, E. F. (2014). *Cellular automata*. Orlando, FL: Academic Press.

Domingo, E., and Holland, J. J. (1997). RNA virus mutations and fitness for survival. *Annu. Rev. Microbiol.* 51, 151–178. doi:10.1146/annurev.micro.51.1.151

Duan, Y., Lu, Z., Zhou, Z., Sun, X., and Wu, J. (2019a). Data privacy protection for edge computing of smart city in a dikw architecture. *Eng. Appl. Artif. Intell.* 81, 323–335. doi:10.1016/j.engappai.2019.03.002

Duan, Y., Sun, X., Che, H., Cao, C., Li, Z., Yang, X., et al. (2019b). Modeling data, information and knowledge for security protection of hybrid iot and edge resources. *IEEE Access* 7, 99161–99176. doi:10.1109/access.2019.2931365

Duan, Y. (2019). "Towards a periodic table of conceptualization and formalization on state, style, structure, pattern, framework, architecture, service and so on," in 2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Toyama, Japan, 08-11 July 2019, 133–138.

Edwards, G., Garcia, J., Tajalli, H., Popescu, D., Medvidovic, N., Sukhatme, G., et al. (2009). "Architecture-driven self-adaptation and self-management in robotics systems," in 2009 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems, Vancouver, BC, Canada, 18-19 May 2009, 142–151.

Gemble, S., Wardenaar, R., Keuper, K., Srivastava, N., Nano, M., Macé, A.-S., et al. (2022). Genetic instability from a single s phase after whole-genome duplication. *Nature* 604, 146–151. doi:10.1038/s41586-022-04578-4

Gindin, Y., Meltzer, P. S., and Bilke, S. (2014). Replicon: A software to accurately predict dna replication timing in metazoan cells. *Front. Genet.* 5, 378. doi:10.3389/fgene.2014.00378

Griffith, S., Goldwater, D., and Jacobson, J. M. (2005). Robotics: Self-replication from random parts. *Nature* 437, 636. doi:10.1038/437636a

Huang, X., Lee, J., Sun, T.-H., and Peper, F. (2013). Self-adaptive self-reproductions in cellular automata. *Phys. D. Nonlinear Phenom.* 263, 11–20. doi:10.1016/j.physd.2013.07.012

Hutton, T. J. (2007). Evolvable self-reproducing cells in a two-dimensional artificial chemistry. *Artif. Life* 13, 11–30. doi:10.1162/artl.2007.13.1.11

Ibáñez, J., Anabitarte, D., Azpeitia, I., Barrera, O., Barrutieta, A., Blanco, H., et al. (1995). "Self-inspection based reproduction in cellular automata," in *European conference on artificial life* (Berlin: Springer), 564–576.

Imai, K., Hori, T., and Morita, K. (2002). Self-reproduction in three-dimensional reversible cellular space. *Artif. Life* 8, 155–174. doi:10.1162/106454602320184220

Kampis, G., and Gulyás, L. (2008). Full body: The importance of the phenotype in evolution. *Artif. Life* 14, 375–386. doi:10.1162/artl.2008.14.3.14310

Klimentidis, Y. (2012). On the limits of diversity. *Front. Genet.* 3. doi:10.3389/fgene.2012.00136

Langton, C. G. (1984). Self-reproduction in cellular automata. *Phys. D. Nonlinear Phenom.* 10, 135–144. doi:10.1016/0167-2789(84)90256-2

Lawson, A. R. J., Abascal, F., Coorens, T. H. H., Hooks, Y., O'Neill, L., Latimer, C., et al. (2020). Extensive heterogeneity in somatic mutation and selection in the human bladder. *Science* 370, 75–82. doi:10.1126/science.aba8347

Lei, Y., and Duan, Y. (2021). Trusted service provider discovery based on data, information, knowledge, and wisdom. *Int. J. Soft. Eng. Knowl. Eng.* 31, 3–19. doi:10.1142/s0218194021400015

Li, Y., Duan, Y., Maamar, Z., Che, H., Spulber, N.-B., Fuentes, S., et al. (2021). Swarm differential privacy for purpose-driven data-information-knowledge-wisdom architecture. *Mob. Inf. Syst.* 6671628, 1–15. doi:10.1155/2021/6671628

Lohn, J. D., and Reggia, J. A. (1997). Automatic discovery of self-replicating structures in cellular automata. *IEEE Trans. Evol. Comput.* 1, 165–178. doi:10.1109/4235.661547

Marchal, P. (1998). John von Neumann: The founding father of artificial life. *Artif. Life* 4, 229–235. doi:10.1162/106454698568567

Merkle, R. (1992). Self replicating systems and molecular manufacturing. *Br. Interplanet. Soc. J.* 45, 407–413.

Miles, L. A., Bowman, R. L., Merlinsky, T. R., Csete, I. S., Ooi, A. T., Durruthy-Durruthy, R., et al. (2020). Single-cell mutation analysis of clonal evolution in myeloid malignancies. *Nature* 587, 477–482. doi:10.1038/s41586-020-2864-x

Monroe, J. G., Srikant, T., Carbonell-Bejerano, P., Becker, C., Lensink, M., Exposito-Alonso, M., et al. (2022). Mutation bias reflects natural selection in arabidopsis thaliana. *Nature* 602, 101–105. doi:10.1038/s41586-021-04269-6

Moore, L., Cagan, A., Coorens, T. H. H., Neville, M. D. C., Sanghvi, R., Sanders, M. A., et al. (2021). The mutational landscape of human somatic and germline cells. *Nature* 597, 381–386. doi:10.1038/s41586-021-03822-7

Morita, K., and Imai, K. (1996a). "Logical universality and self-reproduction in reversible cellular automata," in *International conference on evolvable systems* (Berlin: Springer), 152–166.

Morita, K., and Imai, K. (1996b). Self-reproduction in a reversible cellular space. *Theor. Comput. Sci.* 168, 337–366. doi:10.1016/s0304-3975(96)00083-7

Nehaniv, C. L. (2002). "Self-reproduction in asynchronous cellular automata," in *Proceedings 2002 NASA/DoD conference on evolvable hardware* (Alexandria, VA: IEEE), 201–209.

Peng, S., Wang, G., and Yu, S. (2013). Modeling the dynamics of worm propagation using two-dimensional cellular automata in smartphones. *J. Comput. Syst. Sci.* 79, 586–595. doi:10.1016/j.jcss.2012.11.007

Peper, F., Isokawa, T., Kouda, N., and Matsui, N. (2002). Self-timed cellular automata and their computational ability. *Future Gener. Comput. Syst.* 18, 893–904. doi:10.1016/s0167-739x(02)00069-9

Pesavento, U. (1995). An implementation of von Neumann's self-reproducing machine. *Artif. Life* 2, 337–354. doi:10.1162/artl.1995.2.4.337

Poduri, A., Evrony, G. D., Cai, X., and Walsh, C. A. (2013). Somatic mutation, genomic variation, and neurological disease. *Sci. (New York, N.Y.)* 341, 1237758. doi:10.1126/science.1237758

Reggia, J. A., Chou, H.-H., Armentrout, S. L., and Peng, Y. (1993). Minimizing complexity in cellular automata models of self-replication. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 1, 337–344.

Sahin, U., Uguz, S., Akın, H., and Siap, I. (2015). Three-state von Neumann cellular automata and pattern generation. *Appl. Math. Model.* 39, 2003–2024. doi:10.1016/j.apm.2014.10.025

Salzberg, C. (2003). *Emergent evolutionary dynamics of self-reproducing cellular automata* (Amsterdam, The Netherlands: Universiteit van Amsterdam). Master's thesis.

Sasani, T. A., Ashbrook, D. G., Beichman, A. C., Lu, L., Palmer, A. A., Williams, R. W., et al. (2022). A natural mutator allele shapes mutation spectrum variation in mice. *Nature* 605, 497–502. doi:10.1038/s41586-022-04701-5

Sayama, H. (2004). Self-protection and diversity in self-replicating cellular automata. *Artif. Life* 10, 83–98. doi:10.1162/106454604322875922

Sayama, H. (1999). A new structurally dissolvable self-reproducing loop evolving in a simple cellular automata space. *Artif. Life* 5, 343–365. doi:10.1162/106454699568818

Scott, E. (2013). This i believe: We need to understand evolution, adaptation, and phenotype. *Front. Genet.* 3, 303. doi:10.3389/fgene.2012.00303

Sekanina, L., and Komenda, T. (2011). Global control in polymorphic cellular automata. *J. Cell. Autom.* 6, 301–321. doi:10.1093/imrn/rnq280

Sha, Y., Wang, S., Bocci, F., Zhou, P., and Nie, Q. (2020). Inference of intercellular communications and multilayer gene-regulations of epithelial–mesenchymal transition from single-cell transcriptomic data. *Front. Genet.* 11, 604585. doi:10.3389/fgene.2020.604585

Shenhav, L., and Zeevi, D. (2020). Resource conservation manifests in the genetic code. *Science* 370, 683–687. doi:10.1126/science.aaz9642

Song, Z., Duan, Y., Wan, S., Sun, X., Zou, Q., Gao, H., et al. (2018). Processing optimization of typed resources with synchronized storage and computation adaptation in fog computing. *Wirel. Commun. Mob. Comput.* 3794175, 1–13. doi:10.1155/2018/3794175

Stossel, T. P. (1974). Phagocytosis (first of three parts). *N. Engl. J. Med.* 290, 717–723. doi:10.1056/NEJM197403282901306

Suzuki, K., and Ikegami, T. (2003). "Interaction based evolution of self-replicating loop structures," in *European conference on artificial life* (Dortmund, Germany: Springer), 89–96.

Takada, Y., Isokawa, T., Peper, F., and Matsui, N. (2007). Asynchronous self-reproducing loops with arbitration capability. *Phys. D. Nonlinear Phenom.* 227, 26–35. doi:10.1016/j.physd.2006.12.011

Tempesti, G. (1995). "A new self-reproducing cellular automaton capable of construction and computation," in *European conference on artificial life* (Berlin: Springer), 555–563.

Tempesti, G., Mange, D., and Stauffer, A. (1998). Self-replicating and self-repairing multicellular automata. *Artif. Life* 4, 259–282. doi:10.1162/106454698568585

Tomita, K., Kurokawa, H., and Murata, S. (2002). Graph automata: Natural expression of self-reproduction. *Phys. D. Nonlinear Phenom.* 171, 197–210. doi:10.1016/s0167-2789(02)00601-2

Vogel, F. (1972). Non-randomness of base replacement in point mutation. *J. Mol. Evol.* 1, 334–367. doi:10.1007/BF01653962

von Neumann, J. (1966). *Theory of self-reproducing automata*. Champaign, IL: University of Illinois Press.

Wilke, C. O., Wang, J. L., Ofria, C., Lenski, R. E., and Adami, C. (2001). Evolution of digital organisms at high mutation rates leads to survival of the flattest. *Nature* 412, 331–333. doi:10.1038/35085569

Williams, G. C., and Burt, A. (1997). *Adaptation and natural selection*. Princeton, NJ: Princeton University Press.

Xia, Y., Zhou, M., Luo, X., Zhu, Q., Li, J., Huang, Y., et al. (2015). Stochastic modeling and quality evaluation of infrastructure-as-a-service clouds. *IEEE Trans. Autom. Sci. Eng.* 12, 162–170. doi:10.1109/tase.2013.2276477

Xiao, X., Xue, G.-F., Stamatovic, B., and Qiu, W.-R. (2020). Using cellular automata to simulate domain evolution in proteins. *Front. Genet.* 11, 515. doi:10.3389/fgene.2020.00515

Zuko, A., Mallik, M., Thompson, R., Spaulding, E. L., Wienand, A. R., Been, M., et al. (2021). Trna overexpression rescues peripheral neuropathy caused by mutations in trna synthetase. *Science* 373, 1161–1166. doi:10.1126/science.abb3356

frontiers | Frontiers in Genetics

TYPE Original Research
PUBLISHED 26 August 2022
DOI 10.3389/fgene.2022.941996

OPEN ACCESS

EDITED BY
Yucong Duan,
Hainan University, China

REVIEWED BY
Zhuofeng Zhao,
North China University of Technology, China
Buqing Cao,
Hunan University of Science and Technology, China

*CORRESPONDENCE
Zhangbing Zhou,
zbzhou@cugb.edu.cn

SPECIALTY SECTION
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

RECEIVED 12 May 2022
ACCEPTED 29 June 2022
PUBLISHED 26 August 2022

CITATION
Diao J, Zhou Z, Xue X, Zhao D and Chen S (2022), Bioinformatic workflow fragment discovery leveraging the social-aware knowledge graph. *Front. Genet.* 13:941996. doi: 10.3389/fgene.2022.941996

Bioinformatic workflow fragment discovery leveraging the social-aware knowledge graph

Jin Diao[1], Zhangbing Zhou[1,2]*, Xiao Xue[3], Deng Zhao[1] and Shengpeng Chen[4]

[1]School of Information Engineering, China University of Geosciences (Beijing), Beijing, China, [2]Computer Science Department, TELECOM SudParis, Evry, France, [3]School of Computer Software, College of Intelligence and Computing, Tianjin University, Tianjin, China, [4]Wuda Geoinformatics Co., Ltd., Wuhan, China

Constructing a novel bioinformatic workflow by reusing and repurposing fragments crossing workflows is regarded as an error-avoiding and effort-saving strategy. Traditional techniques have been proposed to discover scientific workflow fragments leveraging their profiles and historical usages of their activities (or services). However, social relations of workflows, including relations between services and their developers have not been explored extensively. In fact, current techniques describe invoking relations between services, mostly, and they can hardly reveal implicit relations between services. To address this challenge, we propose a social-aware scientific workflow knowledge graph (S^2KG) to capture common types of entities and various types of relations by analyzing relevant information about bioinformatic workflows and their developers recorded in repositories. Using attributes of entities such as credit and creation time, the union impact of several positive and negative links in S^2KG is identified, to evaluate the feasibility of workflow fragment construction. To facilitate the discovery of single services, a service invoking network is extracted form S^2KG, and service communities are constructed accordingly. A bioinformatic workflow fragment discovery mechanism based on Yen's method is developed to discover appropriate fragments with respect to certain user's requirements. Extensive experiments are conducted, where bioinformatic workflows publicly accessible at the myExperiment repository are adopted. Evaluation results show that our technique performs better than the state-of-the-art techniques in terms of the precision, recall, and *F1*.

KEYWORDS

bioinformatic workflow, fragment discovery, social relations, knowledge graph, scientific workflow

1 Introduction

With the wide-adoption of web service technology, recurring data and computational resources are increasingly encapsulated as web services or mashup APIs and assembled as scientific workflows (Fischer et al., 2021; Coleman et al., 2022). Online repositories, such as *myExperiment*[1], are publicly accessible for publishing and sharing of scientific workflows constructed by scientists from various disciplines (Gkortzis et al., 2021). Bioinformatics, for example, has seen a spectacular rise in the availability of distributed services (Brandt et al., 2021) and allows rapid and accurate analysis using bioinformatic workflows. Examples of bioinformatic workflows from myExperiment are illustrated in Figure 1. With an increasing number of bioinformatic workflows available online, scientists can reuse and repurpose legacy workflows, rather than developing from scratch, to satisfy novel requirements which are examined to be completely or partially satisfiable by legacy workflows in repositories (Brandt et al., 2021; Rosa et al., 2021). As shown in Figure 1B, the workflow "BiomartAndEMBOSSDisease" retrieves all genes on human chromosome 22, which are associated with a disease, and aligns upstream regions with mouse and rat homologues. This workflow can be reused to reduce the cost when a scientist is willing to design a similar experiment. In fact, considering knowledge-intensiveness and error-proneness for constructing a novel bioinformatic workflow, reusing or repurposing current workflows has been evidenced as an error-avoiding and effort-saving strategy for conducting reproducible bioinformatics experiments (Ren and Wang, 2018; Almarimi et al., 2019). To facilitate the reuse and repurposing of bioinformatic workflows, techniques for discovering and recommending the most relevant fragments of current workflows are fundamental (Yao et al., 2021).

Current techniques have been developed to support the discovery of workflow fragments with similarity assessment. Traditionally, these works evaluate structural similarities between workflows (Bai et al., 2017; Zhang et al., 2018; Zhou et al., 2018), where partial-ordering relations specified upon services are concerned. Although the structure can well-represent the execution semantics of individual workflow fragments, semantic mismatches exist, due to domain differences of workflow developers. To mitigate this problem, annotation-based similarity computation techniques are proposed to complement the structural similarity assessment. Annotations are typically provided by developers to prescribe the category and essential functionalities of certain workflows (Ni et al., 2015; Zhong et al., 2016; Hao et al., 2017). Since workflows may not be accompanied with annotations in certain scenarios (Starlinger et al., 2014), annotation-based strategies with inaccurate similarity calculations may not work as expected. As a result, it may hardly recommend suitable fragments when performing certain scientific experiments.

Considering the fact that developers themselves, who prescribe the annotations, may provide insights about the execution relations between workflows, this study proposes to explore social relations between developers to facilitate recommending appropriate workflow fragments. Figure 1 shows a motivating example of two similar bioinformatic workflows, which are built by two developers who are actually friends. Therefore, incorporating the social relations of developers is promising to further improve the recommendation performance. Discovering fragments from bioinformatic workflows that are assembled by developers in social relations is a promising research challenge. While workflow repositories, such as myExperiment, have been constructed for decades, there still have insufficient socially relevant data on developers. As a result, current techniques focus on gathering and applying certain social information, such as developer reputation, to facilitate the discovery accuracy of appropriate workflows and services (Qiao et al., 2019; Khelloufi et al., 2021; Zhu et al., 2021). In fact, more relations between services (Herbold et al., 2021), and their positive or negative links on workflow fragments discovery and recommendation, have not been explored extensively. Therefore, considering social relevance between developers and services, for facilitating the reuse and repurposing of current workflow fragments, is a challenge to be explored further.

To address this challenge, this study proposes a novel workflow fragment discovery mechanism, by exploring social relations of developers and services that are formed in a knowledge graph. Major contributions presented in this article are summarized as follows:

- We constructed a social-aware scientific workflow knowledge graph (S^2KG) from the myExperiment repository, where services and developers of bioinformatic workflows are encapsulated as entities, and relevant attributes of entities, such as topic, reputation, and domain, are obtained. In addition, multiple relations between entities, including (i) invocation relations between services, (ii) developer relations between services and their developers, and (iii) friend relations between developers, are captured.
- We proposed a novel bioinformatic workflow fragment discovery mechanism leveraging S^2KG. Specifically, positive or negative links between services are identified by analyzing their credits, co-invocation possibilities, and co-developer relations (Ni et al., 2015). A service invoking network (SINet) is formed based on invocation relations between services in S^2KG. Service communities are generated from SINet using the fast unfolding method (Blondel et al., 2008), to facilitate individual candidate services discovery from a functional perspective. Thereafter, services are pairwisely connected through query operations upon S^2KG. The Yen's method (Yen,

1 https://www.myexperiment.org/workflows

1971) is adopted to construct and recommend appropriate workflow fragments to satisfy user's requirements.

Bioinformatic workflows in myExperiment are adopted as the data set in our experiments, where social relations between services and developers are discovered. Extensive experiments are conducted, and evaluation results show that our technique, which complements social relations, outperforms the state-of-the-art counterparts in terms of the precision, recall, and *F1*.

This study is organized as follows. Section 2 introduces relevant concepts of S^2KG and the attributes of entities. Section 3 presents the process of workflow recommendation based on S^2KG. Section 4 evaluates our method and makes a comparison with state-of-the-art techniques. Section 5 discusses related works. Section 6 concludes this study.

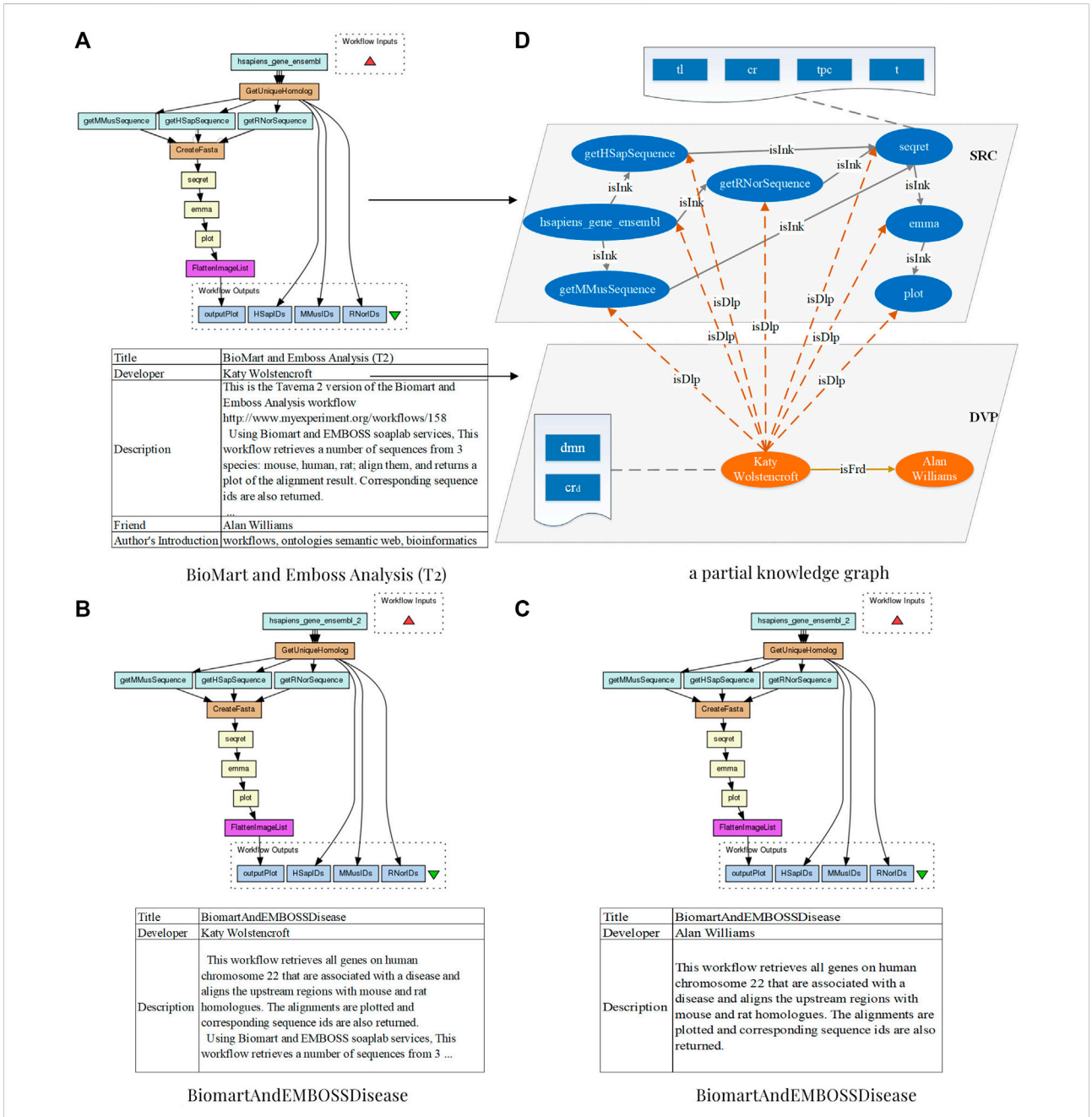


FIGURE 1
Three bioinformatic workflows from *Taverna* 2 of the myExperiment repository with the title "*BioMart and Emboss Analysis (T2)*", "*BiomartAndEMBOSSDisease*", and "*BiomartAndEMBOSSDisease*", respectively, and a partial knowledge graph of the *BioMart and Emboss Analysis (T2)* in S^2KG.

2 S^2KG construction

This section introduces relevant concepts and presents the construction procedure of S^2KG.

2.1 Concepts of S^2KG

myExperiment is an online research environment that supports social sharing of developers' workflows (Goble et al., 2010), which consists of several services. According to these characteristics, the knowledge graph constructed on this repository in this study includes two types of entities, that is, services and developers, as well as three types of relations between them. The workflow is used to reflect the invocation relation between services, so it is not used as a separate entity. The specific definitions are as follows.

A service in S^2KG is defined as follows:

Definition 1 *(Service). A service is a tuple src = (tl, tpc, cr, t), where:*

- *tl is the title of src;*
- *tpc is the topic vector that represents its functions;*
- *cr is its credit, which is calculated based on workflows containing this src;*
- *t represents the created time of src.*

A developer in S^2KG is defined as follows:

Definition 2 *(Developer). A developer is a tuple dvp = (dmn, cr_d), where:*

- *dmn is the topic vector representing his research domains;*
- *cr_d is the reputation calculated by his rating and the credit of his workflows.*

A social-aware scientific workflow knowledge graph (S^2KG) is defined as follows:

Definition 3 *(S^2KG). S^2KG is a tuple (V, LNK), where:*

- *$V = SRC \cup DVP$ is a set of entities for services, SRC, and a set of developers, DVP;*
- *LNK is a set of directed links which specify three kinds of relations: (i) services and services (isInk), (ii) services and developers (isDvp), and (iii) developers and developers (isFrd).*

A scientific workflow in S^2KG is defined as follows:

Definition 4 *(Scientific Workflow). A scientific workflow is a tuple wkf = (cr_w, SRC_w, LNK_w, dsc_w, dvp_w, TG_w), where:*

- *cr_w is the credit calculated upon its download times, viewing times, and rating;*
- *$SRC_w \subset SRC$ is a set of services in wkf;*
- *$LNK_w \subset LNK$ is a set of data links connecting services in SRC_w;*
- *dsc_w is the text description in the profile of wkf;*
- *$dvp_w \subset DVP$ is the developer of wkf;*
- *TG_w is a set of tags provided by dvp_w.*

Figure 1D shows a snippet of S^2KG, which includes several services represented by blue ovals, developers represented by orange ovals, and their relations are represented by arrows with different colors. Specifically, for scientific workflow *BioMart and Emboss Analysis (T2)* in myExperiment, which is a bioinformatic workflow, as shown in Figure 1A, its developer Katy Wolstencroft is represented by orange ovals. Its services are represented by blue ovals; for example, the service *hsapiens_gene_ensembl*. Blue rectangles in wavy rectangles describe the properties of entities, such as the *dmn* and cr_d of Katy Wolstencroft, and the *tl*, *tpc*, *cr*, and *t* of *hsapiens_gene_ensembl*. According to the workflow specification, relations between a developer and his services are extracted as *isDvp* and represented by the orange dotted line; for example, the relation between Katy Wolstencroft and his services *hsapiens_gene_ensembl*. Based on data links in workflows, relations between services are extracted as *isInk* and represented by the gray lines; for example, the relation between the service *hsapiens_gene_ensembl* and the service *getRNorSequence*. Specially, *GetUniqueHomolog* and *CreateFasta* are *beanshells* for cohesion, so they are not regarded as services. Finally, the relation between developers and their friends is extracted as *isFrd* and represented by the yellow arrow in this figure. For example, Katy Wolstencroft, the author of workflows shown in Figures 1A,B, and Alan Williams, the author of the workflow shown in Figure 1C, are friends, and their relation is represented by a yellow arrow and labeled as *isFrd*.

2.2 Topic of services

This section constructs topic vectors of services for representing their functions and domains. For a service, the title and text description in its profile prescribe its original functionality. However, since services are constantly being combined for new application scenarios, their profiles can hardly reflect their new application scenarios and functions. As is often the case, various workflow information sharing platforms provide rich descriptions to describe their domains and functions (Gu et al., 2021). Workflows can be regarded as a set of interdependent services that implement complex functions. Based on this observation, we argue that workflows can be considered as the domain of relevant services to provide their integrated functional description. For a more comprehensive representation of service topics,

these functions and domains are used to generate topic vectors for the corresponding services. In total, three sample scientific workflows are shown in Figure 1, and they contain similar services but have different descriptions to represent novel domain of services.

Require:
- SRC : a set of services
- WKF : a set of workflows

Ensure:
- DOC : a set of documents

1: **for** $\forall\ src_i \in SRC$ **do**
2: $doc_i \leftarrow src_i.tl \cup src_i.dsc$;
3: **for** $\forall\ wkf_j \in WKF$ **and** $src_i \in wfk_j.SRC_w$ **do**
4: **for** $\forall\ snt_i \in wfk_j.dsc_w$ **and** $\forall\ wd_{si} \in src_i.tl$ **do**
5: **if** $snt_i.contains(wd_{si})$ **then**
6: $doc_i \leftarrow snt_i \cup doc_i$
7: **end if**
8: **end for**
9: **for** $\forall\ wdt_i \in wfk_j.TG_w$ **and** $\forall\ wd_{si} \in src_i.tl$ **do**
10: **if** $wdt_i.contains(wd_{si})$ **then**
11: $doc_i \leftarrow wdt_i \cup doc_i$
12: **end if**
13: **end for**
14: **end for**
15: $DOC \leftarrow DOC \cup \{doc_i\}$
16: **end for**

Algorithm 1. Service corpus construction

Algorithm 1 presents the construction procedure of service corpus contained in workflows. To prescribe the functionality of each service src_i, its title $src_i.tl$ and text description $src_i.dsc$ are assembled as a document doc_i (line 2). To present the novel domain of src_i, the related description in $wfk_j.dsc_w$ and tags in $wfk_j.TG_w$ of each workflow wkf_i containing src_i are added to doc_i (lines 3–14), where contains () is a comparison function, snt_i is the ith sentence of $wfk_i.dsc_w$, wd_{si} is the ith word of $wrc_i.tl$, $src_i.tl$ is the title of src_i, and wdt_i is the ith tag in $wfk_j.TG_w$. All documents construct the corpus for generating topics for services (line 15). Note that doc_i contains several paragraphs, mostly, and could hardly be regarded as a short text, which usually contains less than five words or no more than 140 characters (Li et al., 2016). Therefore, considering the size of DOC, the Latent Dirichlet Allocation (LDA) model (Blei et al., 2003) is adopted to generate topics for service corpus. Generally, LDA is a bag-of-words model and widely used in general-scale long text classification, where stop words are removed during the model preprocessing phase.

The time complexity of Algorithm 1 is O ($|SRC|*|WKF|*|SRC_w|*|SNT|$), where $|SRC|$ is the number of services in the repository, $|WKF|$ is the number of workflows in the repository, $|SRC_w|$ is the number of services in the jth workflow, and $|SNT|$ is the number of sentences in the description of wkf_j. Note that line 9 should iterate fewer times than line 4, and thus, the time complexity of Algorithm 1 is determined by lines 1, 3, and 4.

Require:
- DOC : a set of documents generated by Algorithm 1
- α : a doc-topic Dirichlet prior parameter
- β : a topic-word Dirichlet prior parameter
- tp_n : the number of topics
- itt : the times of iterations
- vb : the size of vocabularies

Ensure:
- ϕ : parameters for topic-word distribution
- θ : parameters for doc-topic distribution

1: nkt, $nktS$, nmk, $nmkS$, $z \leftarrow$ iniModel(DOC)
2: **for** $itt \in [1, itt]$ **do**
3: **for** $tp \in [1, tp_n]$ **and** $tr \in [0, vb]$ **do**
4: $\phi_{kt} \leftarrow (nkt_{kt} + \beta) \div (nktS_k + vb \times \beta)$
5: **end for**
6: **for** $m \in [1, DOC.length]$ **and** $tp \in [1, tp_n]$ **do**
7: $\theta_{mk} \leftarrow (nmk_{mk} + \alpha) \div (nmkS_m + tp_n \times \alpha)$
8: **end for**
9: **for** $l \in [1, DOC.length]$ **and** $p \in [1, doc_l.length]$ **do**
10: $z \leftarrow$ smplTpcZ(l, p)
11: **end for**
12: **end for**

Algorithm 2. Service topic model construction

Leveraging DOC generated by Algorithm 1, Algorithm 2 introduces the service topic vector construction procedure. Specifically, the model is initialized and parameters are generated leveraging the set of documents DOC, where nkt is the count of a term for a certain topic, nmk is the count of a topic for a certain document, $nktS$ is the sum for the kth row in nkt, $nmkS$ is the sum for the mth row in nmk, and z is the generated topic label array (line 1). During each iteration itt, we continuously updated the parameters for topic–word distribution ϕ (lines 3–5), as well as the parameters for doc–topic distribution θ (lines 6–8), where tp_n is the number of topic; vb is the vocabulary; $DOC.length$ is the size of DOC; $doc_l.length$ is the size of document doc_l; and tp, tr, and m are local variables. The Gibbs sampling $smpleTpcZ$ () is adopted to update topic label array afterward, where l and p are local variables (lines 9–11). Please refer to (Blei et al, 2003) for the specific sampling process. The time complexity of Algorithm 2 is O ($itt*DOC.length*doc_l.length$). Note that lines 3 and 6 should iterate fewer times than line 9, and thus the time complexity of Algorithm 2 is determined by lines 2 and 9.

2.3 Reputation of services

This section constructs the credit of services through the collective perception of workflows containing these services. A service is applied in multiple workflows with some reputation information representing their popularity. As components of a workflow, the credit of every service contributes to an accurate partial-execution of this workflow, which indicates that users prefer to obtain a service with certain quality. To evaluate the quality of services, the method described in Yao et al. (2014) is used to calculate the credit (cr) of services leveraging the workflows information as follows.

Generally, the credit cr_w of a workflow $wkf_i.cr_w$ reflects the degree of adoption by developers, and it is calculated by three

factors including viewing times ($wkf_i.n_v$), download times ($wkf_i.n_d$), and rating ($wkf_i.n_{rt}$) by the following formula.

$$wkf_i.cr_w = f_{crd}(wkf_i.n_v, wkf_i.n_d, wkf_i.n_{rt}) \quad (1)$$

Where f_{crd} is a monotonic increasing function to ensure that the quality of a workflow is directly proportional to its popularity.

For each service in a workflow, its credit can be calculated by adopting a fair-share method, as presented in Nepal et al. (2009). Specifically, due to different importance, the credit of services in a workflow $wkf_i.V$ should be assigned according to its importance as follows.

$$wkf_i.V = v_1, v_2, \ldots, v_n \quad (2)$$

Where $v \in [0, 1]$ and $\sum v = 1$. Based on Eq. 2, the credit of each service $src_j.cr$ is computed using the formula below.

$$\forall src_j \in wkf_i.SRC_w \qquad src_j.cr = v_j wkf_i.cr_w \quad (3)$$

Since a service src_j may be adopted in several workflows, the average credit is regarded as the credit $src_j.cr$.

$$src_j.cr = \frac{\sum_{i=1}^{WN} wkf_i.src_j.cr}{WN} \quad (4)$$

Where WN is the number of workflows containing src_j, and wkf_i. $src_j.cr$ is the credit of src_j calculated by wkf_i.

2.4 Domain and reputation of developers

This section constructs the topic vector of developers for presenting their research domains which influence their services' and workflows' functionality and domains. Through examining the information about developers in myExperiment, a developer generally has four features describing his research domains, including his introduction, interests, tags, and field (or industry). These features are adopted to generate topic vectors of corresponding developers leveraging Algorithm 3.

Require:
- DVP : a set of developers

Ensure:
- ϕ : parameters for topic-word distribution
- θ : parameters for doc-topic distribution

1: **for** $\forall\ dvp_i \in DVP$ **do**
2: $\quad doc_i \leftarrow dvp_i.itd \cup dvp_i.itr \cup dvp_i.fld \cup dvp_i.TG_d$
3: $\quad DOC \leftarrow DOC \cup \{doc_i\}$
4: **end for**
5: $\phi, \theta \leftarrow DOC$.Algorithm 2

Algorithm 3. Developer topic model construction

Algorithm 3 shows the construction of topic vectors for developers. To prescribe the domain of each developer dvp_i (line 1), the introduction $dvp_i.itd$, interests $dvp_i.itr$, field $dvp_i.fld$, and tags $dvp_i.TG_d$ are assembled into a document doc_i (line 2), and these documents construct the corpus for generating topics of developers (line 3). Specifically, $dvp_i.itd$ and $dvp_i.itr$ are texts with several functional paragraphs, and $dvp_i.fld$ and $dvp_i.TG_d$ are some concise words. As mentioned in Section 2.2, doc_i of each developer is not a short text. Thus, the *LDA* model is adopted to generate topic vectors for developers (line 5). Note that the number of iterations in line 1 should be less than that in line 5, so the time complexity of Algorithm 3 is determined by Algorithm 2, where *DOC* is the corpus of developers involved in this algorithm.

The reputation is calculated to reflect the trust degree of a developer. We use the method proposed in Yao et al. (2014) to calculate this value using the developer's rating and his services' credit. Specifically, each developer in myExperiment has an average rating to reflect his contribution. Hence, the rating is considered as an important feature for calculating the reputation. In addition, the credit of his previously developed services is another feature that indicates his reputation. Therefore, the reputation cr_d of a developer is calculated leveraging the follow formula (Yao et al., 2014).

$$cr_d = f_{rp}(rt_{d_i}, \{p_{d_i}\}) \quad (5)$$

Where the function f_{rp} is a monotonic increasing function, which ensures that the reputation of a developer is high when his credit is high and the quality of his services is high as well. rt_{d_i} is the rating of a developer calculated by the platform. $\{p_{d_i}\}$ is the credit set of his services.

3 Bioinformatic workflow fragment discovery

This section presents the identification of positive and negative links between services to support bioinformatic workflow fragment discovery, involving the selection of candidate atomic services leveraging community detection, and the discovery of their fragments in S^2KG.

3.1 Union impact based on positive and negative links

There exists positive or negative links between pairs of services. Positive links specify correlations, collaborations, and complementary relations between services, whereas on the contrary for negative links. Based on S^2KG, four types of positive links are identified to guide service cooperation (Ni et al., 2015). A service src_i may compose with another src_j, when

1) src_j has a good credit,
2) src_j has a highly similar topic with src_i,
3) the developer of src_j is same as that of src_i, or
4) the developer of src_j is a friend with similar topics to the developer of src_i.

Specifically, the higher the credit of a service is, the higher the possibility that this service is selected to compose a novel workflow. Thus, the positive link Cr_{ij} between src_i and src_j is calculated to reflect the impact of their credit $src_i.cr$ and $src_j.cr$ as follows.

$$Cr_{ij} = src_i.cr \times src_j.cr \tag{6}$$

Where $src_i.cr$ and src_jcr are the credit of src_i and src_j calculated by Eq. 4.

- The second positive link Sim_{ij} is identified to calculate the similarity of src_i and src_j by the following formula Eq. 7 leveraging the services' topic vectors constructed in Section 2.2.

$$Sim_{ij} = \frac{\sum_{k=1}^{tp_n}\left(src_i.tpc_k \times src_j.tpc_k\right)}{\sqrt{\sum_{k=1}^{tp_n}\left(src_i.tpc_k\right)^2} \times \sqrt{\sum_{k=1}^{tp_n}\left(src_j.tpc_k\right)^2}} \tag{7}$$

Where $src_i.tpc_k$ and $src_j.tpc_k$ are the values of the kth feature in $src_i.tpc$ and $src_j.tpc.tp_n$ is the total number of topics. The higher the results are, the more similar the two topic vectors are. A threshold trd_t is prescribed to examine whether two services are similar. Intuitively, when $Sim_{ij} \geq trd_t$, the topic of two services are similar, and not otherwise.

- The third positive link Sd_{ij} is identified by Eq. 8 to examine whether the developers of src_i and src_j are same. Considering the stickiness of a developer's domain, his services should be similar in terms of his topics. These services may be easier to adapt from the perspective of structure, and their composition may match the functional requirements more appropriately.

$$Sd_{ij} = \begin{cases} 1 & \text{if } \exists\left(dvp_i,\ isDvp,\ src_j\right) \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Where dvp_i is the developer of src_i, and dvp_i, $isDvp$, and src_j means that the developer of src_j is also dvp_i. As shown in Figure 1A and Figure 1B, these two workflows are constructed by the same developer Katy Wolstencroft. Their structures are similar, but they are adopted in different domains and have different titles and introductions.

- The fourth positive link Sf_{ij} is identified, when two developers are friends, their domains and interests may be similar. Thus, the topic of services they developed should be similar.

$$Sf_{ij} = \begin{cases} 1 & \text{if } \exists\left(dvp_i,\ isFrd,\ dvp_j\right) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Where (dvp_i, $isFrd$, and dvp_j) means that the developer dvp_i of src_i is a friend of the developer dvp_j of src_j. As shown in Figure 1, the developers of Figure 1B and Figure 1C are friends. As we can see, they have constructed the similar workflows with the same title and different functional description.

Negative links indicate functionality uncorrelations, conflicts, or even competitions between services. Based on S^2KG, a union negative link TC_{ij} is identified leveraging Cr_{ij} and Tm_{ij}. Specifically, Tm_{ij} is a negative link specifying that services may cooperate with very low feasibility if they have not cooperated in the same workflow since their creation. Tm_{ij} is calculated as follows.

$$Tm_{ij} = now - \max\left(src_i.t, src_j.t\right) \tag{10}$$

Where now is the current time. The uncooperative duration of two services is determined by the latest service. The larger the value of Tm_{ij} is, the less likely that these two services are cooperated to construct a novel workflow.

Based on Tm_{ij}, TC_{ij} can be formed as follows to present that two services are unlikely to cooperate.

$$TC_{ij} = Tm_{ij} \times Cr_{ij} \tag{11}$$

Generally, the larger the value of TC_{ij} is, the lower the feasibility that src_i and src_j can be cooperated.

As mentioned before, given two services, we adopted the union impact U_{ij} through integrating positive and negative links to determine whether they can be cooperated, as follows.

$$U_{ij} = \alpha \times Sim_{ij} + \beta \times Cr_{ij} - \gamma \times TC_{ij}, \quad if\ Sd_{ij} = 1\ or\ Sf_{ij} = 1 \tag{12}$$

Where α, β, and γ are the importance of each influencing factor, and $\alpha + \beta + \gamma = 1$.

3.2 Service discovery leveraging community detection

Due to the different levels of users' expertise, a requirement in this study is composed of several sub-requirement descriptions in an effort to express the requirement more clearly. Generally, it can be formalized in terms of $Q = \{q_1, q_2, \ldots, q_m\}$. For each sub-requirement, an appropriate service is discovered accordingly. To facilitate single service discovery from the functional perspective, services and *isInk* relations are extracted from S^2KG and construct a Service Invoking Network (*SINet*). For example, the service *hsapiens_gene_ensembl* and the service *getRNorSequence* in the workflow *BioMart and Emboss Analysis (T2)* are divided into the same purple community because of similar application scenarios. The fast unfolding method (Blondel et al., 2008), which is heuristic based on modularity optimization, is adopted to divide *SINet* into several functional communities. This method adjusts the division of communities by continuously optimizing the modularity, where the modularity of a partition is a measure of the density of links within the community and the density of links between communities (Newman, 2006) as defined by Eq. 13.

$$CM = \frac{1}{2m} \times \sum_{i,j}\left[A_{ij} - \frac{k_i k_j}{2m}\right] \times \delta\left(c_i, c_j\right) \tag{13}$$

$$k_i = \sum_j A_{ij} \tag{14}$$

$$m = \frac{1}{2}\sum_{i,j} A_{ij} \tag{15}$$

$$\delta(c_i, c_j) = \begin{cases} 1 & c_i = c_j \\ 0 & others \end{cases} \tag{16}$$

Where A_{ij} represents the wight of the link between src_i and src_j, and the wight is Sim_{ij} as calculated by Eq. 7. k_i is the sum of wights of links which connect to src_i. m is the sum of link wights in SINet. c_i is the community to which src_i is assigned. $\delta(c_i, c_j)$ represents the fact that whether c_i and c_j are same. By dividing SINet into communities, the entire network is a set CM of communities, and each community c_i in CM is a tuple $c_i = \{c_t, CS\}$, where c_t is the topic vector of the representative service of c_i and CS is a set of services in c_i.

Require:
-$SINet$: a service invoking network
-Q : a set of a user's requirements
Ensure:
-S : a set of candidate services

1: $CM \leftarrow \text{Louvain}(SINet)$
2: $k \leftarrow 0$
3: **for** $\forall\ q_i \in Q$ **and** $\forall\ c_j \in CM$ **do**
4: **if** $Sim(q_i, c_j.c_t) \geq k$ **then**
5: $CT_i \leftarrow CT_i \cup \{c_j\}$
6: **end if**
7: **end for**
8: **for** $\forall\ c_i \in CT$ **and** $\forall\ src_j \in c_i.CS$ **do**
9: **if** $\text{Sim}(q_i, src_j.tpc) \geq trd_{sc}$ **then**
10: $S \leftarrow S \cup \{src_j\}$
11: **end if**
12: **end for**

Algorithm 4. Candidate service discovery

Based on CM, the most relevant communities and candidate services are discovered. Algorithm 4 represents candidate communities and services discovery procedure. First, SINet is divided into several communities leveraging the fast unfolding method (Blondel et al., 2008) according to service topics (line 1). A comparison variable k is set to 0 (line 2). For each sub-requirement q_i and each community c_j, the functional similarity between them is calculated by the comparison function $Sim()$ and compared with k, where q_i is vectorized by embedding. The community c_j with the most similar functionality to q_i is inserted into a set of candidate communities CT (lines 3–7). For each candidate community in CT, the similarity of each $src_j.tpc$ and q_i is calculated and compared with the pre-specified threshold trd_{sc}. If the similarity is larger than trd_{sc}, src_j is inserted into the set S as candidate services (lines 8–12). The time complexity of Algorithm 4 is O (|CT|*|CS|), where |CT| is the number of CT and |CS| is the number of services in the community c_i. Note that line 3 should iterate fewer times than line 8, and thus, the time complexity of Algorithm 4 is determined by line 8.

TABLE 1 Data set in *Taverna 2*.

Statistics	Value
# of service	2,870
# of workflow	1,058
# of developer	175
# of *isInk*	2,516
# of *isDvp*	2,870
# of *isFrd*	271

3.3 Bioinformatic workflow fragment discovery

Based on candidate services discovered by Algorithm 4, this section proposes to discover appropriate workflow fragments, where relations prescribed by S^2KG are obtained to connect candidate services for respective service stubs in the requirement. The Yen's method (Yen, 1971), which is a heuristic method widely used in graph traversal, is adopted to discover and compose relevant workflow fragments from various workflows.

Require:
-S^2KG : a services social knowledge graph
-S : a candidate services set
Ensure:
-MPS : a set of fragments

1: **for** $\forall\ src_i, src_j \in S$, where $i \neq j$ **do**
2: $PH_0 \leftarrow \text{Dijkstra}(U_{ij}, src_i, src_j)$
3: **for** $k \in [1, k_{\#}]$ **do**
4: **for** $i \in [0, \text{size}(PH_i) - 1]$ **do**
5: $src_{sp} \leftarrow PH_{k-1}.src_i$
6: $Ph_{rt} \leftarrow PH_{k-1}.srcs(0, \text{i})$
7: **for** $\forall\ p \in PH$ **do**
8: **if** $Ph_{rt} = p.srcs(0, \text{i})$ **then**
9: $S^2KG.LNK \leftarrow S^2KG.LNK - \{(p.edge(\text{i}, \text{i+1}))\}$
10: **end if**
11: **end for**
12: $Ph_{sp} \leftarrow \text{Dijkstra}(U_{spj}, src_{sp}, src_j)$
13: $BPH \leftarrow BPH \cup \{(Ph_{rt} + Ph_{sp})\}$
14: $S^2KG \leftarrow S^2KG.\text{rtEdg}()$
15: **end for**
16: **if** $BPH = \text{NULL}$ **then**
17: break
18: **end if**
19: $PH_k \leftarrow BPH.\text{MAX}()$
20: **end for**
21: $MPS \leftarrow MPS \cup PH$
22: **end for**

Algorithm 5. CFDY: Crossing-workflow fragment discovery using Yen's method

The Algorithm 5 (denoted *CFDY*) shows the procedure of discovering appropriate bioinformatic workflow fragments. First, the *Dijkstra* () is adopted to find the optimal combinatorial fragment PH_0 from the service src_i to the service src_j leveraging the union impact U_{ij} (lines 1,2). Based on PH_0, the kth

combinatorial fragment is found (lines 3–19). Above all, every deviated service is traversed (lines 4–15). Specifically, src_{sp} is retrieved from the (*k-1*)th best combinatorial fragment and Ph_{rt} records the sequence of services from src_i to src_{sp} (lines 4–6). The links that belong to part of the previous best combinatorial fragment of the same Ph_{rt} are removed from the S^2KG (lines 7–11). The combinatorial fragment from src_{sp} to src_j is found by the *Dijkstra* () and recorded to Ph_{sp} (line 12). Entire combinatorial fragment is made up of Ph_{rt} and Ph_{sp} and added to the set *BPH* (line 13). The links that were removed before are added back to S^2KG (line 14). If there are no other combinatorial fragments, the method ends (lines 16–18). The optimal combinatorial fragment in *BPH* is the *k*th combinatorial fragment PH_k (line 19). All paths in *PH* are added to the set *MPS* (line 21). The time complexity of Algorithm 5 is O ($|S|*k_{\#}*size(PH)*|PH|$), where $|S|$ is the number of candidate services, *size(PH)* is the value of *size* (PH_i) minus 1, and $|PH|$ is the number of path.

4 Implementation and evaluation

This section presents our experiments and evaluation results. Experiments are performed on a desktop computer with an Intel i7 6,700 processor at 3.40 and 3.41 GHz, 8.00 GB of RAM and a 64-bit Windows 10 operating system. The prototype is implemented by *Python* and Java.

4.1 Data set and preprocessing

This study adopts bioinformatic workflows in *myExperiment* for our experiments, where workflows in the *Taverna 2* category by May 2019 are crawled. For each service, its title, description, created time and developer are collected. For each workflow, its title, description, tags, publishing date, download times, viewing times, rating, developer, services and data links are collected, where the data links reflect the control flows between services (i.e., invocation relations). For each developer, his name, introduction, interests, field, rating and friends are collected. Note that services and workflows without a title or description are deleted. As a summary, the numbers of available services, workflows, developers and their relations are shown in Table 1.

The data cleaning procedure is conducted, where stop words are removed, and the stemming of words is extracted. Thereafter, entities and relationships are extracted, and their attributes are obtained by the techniques presented in Section 2. We adopt the graph database Neo4j (Robinson et al., 2015) to store these cleaned data.

To evaluate the efficiency of our technique, we have generated 40 crossing-workflow fragments leveraging legacy workflows as testing fragments based on S^2KG. According to the statistic reported in our previous work (Zhou et al., 2020), roughly 86% of workflows contains no more than 11 services. Therefore, 5 out of 40 testing fragments are set to contain over 11 services.

4.2 Measurement metrics

Three metrics are adopted to evaluate the accuracy and effectiveness of our technique as follows:

- *P*: The precision (denoted *P*) indicates the percentage of the number of correctly recommended services over the total number of recommended services.

$$P = \frac{|CS_{pt} \cap CS_{rc}|}{|CS_{rc}|} \tag{17}$$

- *R*: The recall (denoted *R*) refers to the percentage of the number of correctly recommended services over the total number of desired services.

$$R = \frac{|CS_{pt} \cap CS_{rc}|}{|CS_{pt}|} \tag{18}$$

- *F1*: The *F1* score is used for an overall evaluation based on *P* and *R*.

$$F1 = \frac{2 \times P \times R}{P + R} \tag{19}$$

Where CS_{pt} is the expected service set, and CS_{rc} is the set of recommended services.

4.3 Baseline techniques

In this section, the following four state-of-the-art techniques are chosen as baselines to evaluate the effectiveness of our technique:

- *CSBR* (Gu et al., 2021) is a semantics-based model to compose and recommend a set of complementary services for workflow construction. By applying this approach, we first construct a semantic service bundle repository using experimental data. Then a bundle of complementary services is recommended to fulfill the sophisticated requirements. Finally, a more suitable result is found using a greedy approximation method considering the time complexity.
- *ClstRec* (Conforti et al., 2016) is a modularized clustering algorithm to generate service clusters. We first identify target clusters for each service stub, find their services or fragments therein, and sort them into candidate services or fragments. Then a series of fragments are constructed

TABLE 2 *Prp* settings with various tp_n.

tp_{ns}	10	20	30	40	42	43	44	45	46
Prp_s	196.860	149.406	135.332	132.726	132.176	131.505	131.757	132.750	132.752
tp_{nd}	2	4	6	8	9	10	11	12	13
Prp_d	654.289	509.672	468.705	468.485	465.883	454.085	463.771	479.020	479.157

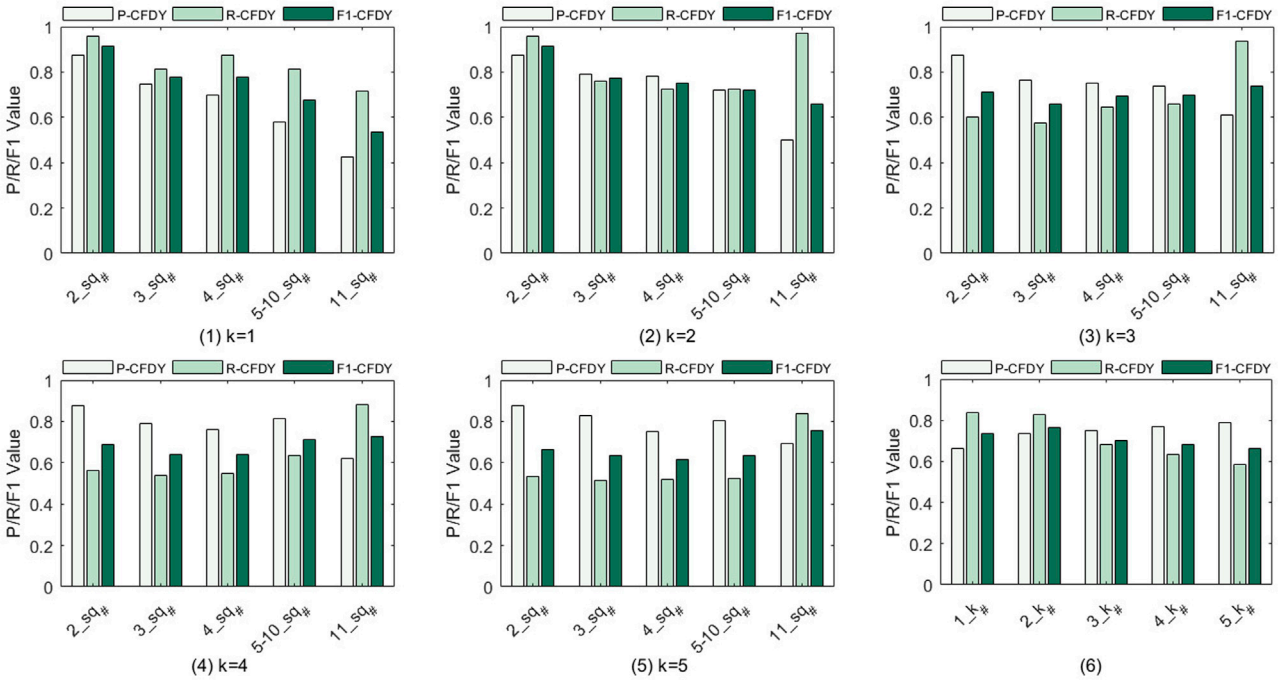


FIGURE 2
Precision, recall, and F1 value for the fragment discovery when $k_\#$ is set to 1, 2, 3, 4, and 5, respectively.

across workflows based on their relations. Based on their similarity, these fragments are identified, ranked, and recommended accordingly.

- *CDSR* (Xia et al., 2015) is a category-aware clustering and distributed service recommending method to automatically create fragments. First, we cluster the experimental data into various categories based on the similarity of functionality and popularity of their services. Then we map requirements to relevant categories to find candidate services. Finally, these candidate services in the most relevant categories construct cross-workflow fragments to fulfill the requirements.
- *Short Path* (denoted *SP*) method is a classical heuristic algorithm. First, we start to navigate from a service and select the neighbors with the highest relevance, which have a connection-aware relation with it, according to a given probability distribution. Then a similar operation is performed starting from that service to find a service fragment.

4.4 Evaluation results

In this section, we first optimize the algorithm *CFDY* by adjusting the following parameters tp_n and $k_\#$ and then use the parameters $sq_\#$ and trd_U to discuss the evaluation results of *CFDY* and baselines.

- tp_n: *The topic number.* The semantic description is susceptible to the topic number. Different number of topics should lead to different partitions of services and developers and recommend various results. Therefore, determining an appropriate tp_n is fundamental and crucial.

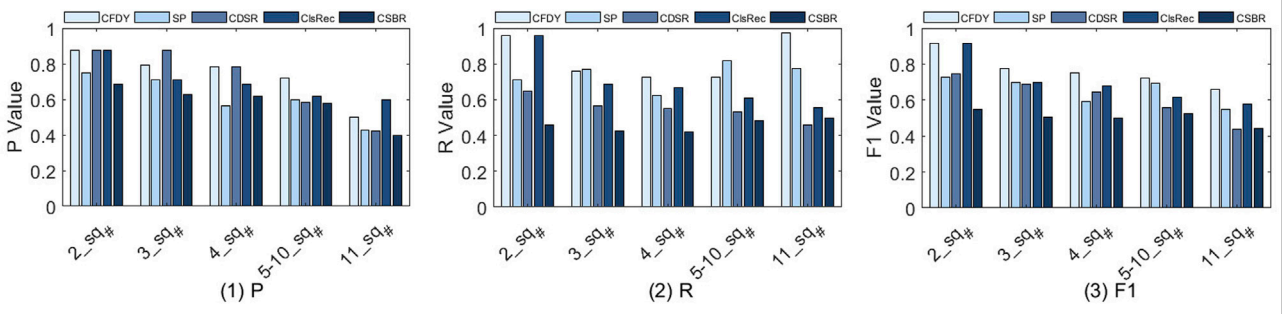


FIGURE 3
P, *R*, and *F1* value for the fragment discovery, when the sub-requirements number is set to 2, 3, 4, 5–10 and 11, respectively.

- $k_{\#}$: *The number of paths*. When it changes, it should affect the number of path searches of *CFDY*. Different $k_{\#}$ should affect the number of services in results, thereby affecting the efficiency of *CFDY*.
- $sq_{\#}$: *The number of sub-requirements*. With the increase of $sq_{\#}$, the number of service and the complexity of fragments should increase, thereby affecting the efficiency of the fragment discovery.
- trd_U: *The connection-aware threshold of two services*. It should influence the efficiency of our method by changing the scale of candidate services set.

4.4.1 Impact of tp_n

To select the optimal tp_n, a widespread perplexity is used to calculate the quality of the LDA model, as shown below, which describes the degree of uncertainty of the model about documents and their topic. Therefore, the lower the perplexity is, the better predictive effect.

$$Prp = \exp\left\{-\frac{\sum_{d=1}^{M} \log p(w_d)}{\sum_{d=1}^{M} N_d}\right\} \tag{20}$$

Where M is the number of *DOC*, N_d represents the number of words in a *doc*, and $p(w_d)$ is the probability of that the word w_d is contained in the *doc*.

As shown in Table 2, the perplexities of services' and developers' LDA models are calculated separately. For the LDA model of services, with the increasing of topic number (denoted as tp_{ns}), its perplexity (denoted as Prp_s) decreases. When Prp_s is 131.505, tp_{ns} is selected to the optimal value as 43. For the LDA model of developers, its perplexities (denoted as Prp_d) are calculated when its topic number (denoted as tp_{nd}) ranges from 2 to 13. When tp_{nd} is 10, Prp_d is the smallest value as 454.085. Therefore, 43 and 10 are determined as the tp_n of two LDA models.

4.4.2 Impact of $k_{\#}$

The influence of different $k_{\#}$ on *P*, *R*, and *F1* is shown in Figure 2 when $k_{\#}$ is set to 1, 2, 3, 4 and 5, respectively. tp_{ns} is set to 43, tp_{nd} is set to 10, and trd_U is set to 0.8. Considering that different $sq_{\#}$ will affect the complexity of fragments discovery, the efficiency of various $k_{\#}$ is evaluated with test examples containing 2–11 sub-requirements. Considering that the number of test examples containing 5–10 sub-requirements is small, these examples are grouped into one category.

- Figure 2(1) and Figure 2(2) show that, as $sq_{\#}$ gradually increases, *P*, *R* and *F1* as a whole gradually decrease. In particular, in Figure 2(2), when $sq_{\#}$ is 11, there is a sudden change in *R* that has a higher value. The increase of $sq_{\#}$ leads to the increase of the number of services contained in CS_{pt}. Therefore, the fragment structure becomes more complex. Generally, when $sq_{\#}$ is larger, the single path search can hardly fulfill the complexity requirement well. Compared with Figure 2(1), our technique increases the number of paths, and the number of services contained in CS_{rc}. Thus, *R* is increased to some extent.
- As shown in Figure 2(3)–2 (5), when $k_{\#}$ is larger than 2, their *P*, *R* and *F1* have similar tends. With the gradual increase in $sq_{\#}$, *P* and *F1* gradually decrease, but *R* has increased to a certain extent. In the same way, the decrease in *P* and *F1* is due to the increase in $sq_{\#}$ and the complexity of their structure. With the increase of $k_{\#}$, the number of services in CS_{rc} increases, which will lead to an increase in *R*. However, as $k_{\#}$ gets larger, too much exploration will lead to a decrease in *R* when $sq_{\#}$ is small. For example, compared with *R* in Figure 2(1), *R* in Figure 2(5) is significantly lower when $k_{\#}$ equal to 2.
- Figure 2(6) shows the average values of *P*, *R*, and *F1*, at each $k_{\#}$. Larger $k_{\#}$ means that more fragments are constructed. Due to the increase of fragments, more services are selected. Hence, this result may lead to an increase of *P*. However, since the number of expected services is fixed, blindly increasing the number of recommended services by adding too many paths may not maintain the increase in *R*. *F1* has a maximum value at 2. Therefore, 2 is finally selected as the value of $k_{\#}$ in subsequent experiments.

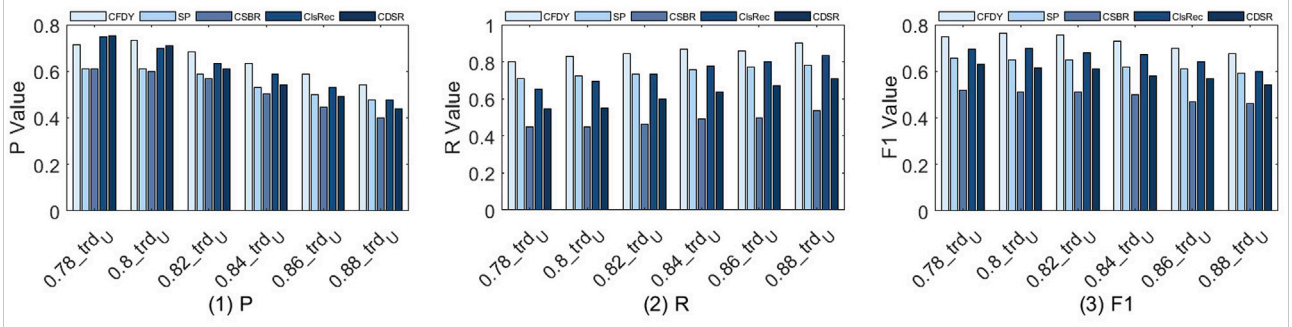


FIGURE 4
P, *R*, and *F1* value for the fragment discovery when the connection-aware threshold is set to 0.78, 0.80, 0.82, 0.84, 0.86, and 0.88, respectively.

4.4.3 Comparison on $sq_{\#}$

We compare five methods and estimate the influence of different $sq_{\#}$ for fragment discovery. The results of *P*, *R*, and *F1* are shown in Figure 3 when $sq_{\#}$ is set to 2, 3, 4, 5–10, and 11, respectively. tp_{ns} is set to 43, tp_{nd} is set to 10, $k_{\#}$ is set to 2, and trd_U is set to 0.8.

Figure 3(1) shows that *P* decreases with the increase of $sq_{\#}$, and the overall performance of *CFDY* is better than other methods.

- For CFDY, as $sq_{\#}$ increases, its p decreases. When $sq_{\#}$ is small, its structure is relatively simple, and the expected service can be selected more accurately by using the relationship between services. With the increase of $sq_{\#}$, the structure of the fragment becomes more complex and contains more branches. Therefore, the selection of candidate services brings certain difficulties, and thus the accuracy rate is reduced.
- Compared with *CFDY*, *SP* only explores a single path and lacks the exploration of branches. Its recommended fragment does not contain some of the expected services present in the branch. As a result, the number of expected services in its recommendation set is less than *CFDY*, which results in its *P* being lower than that of *CFDY*.
- In fact, *CSBR* pursues more semantic similarity matching, and the consideration of structural similarity is not a priority, which leads to the fact that most of the services it recommends are not the expected ones. Therefore, its overall performance is the lowest compared to others.
- *CDSR* uses category awareness to cluster services and considers the impact of service coexistence time on its relationship when considering historical combination information. By considering the functional similarity and relations, when $sq_{\#}$ is less than 5, its *P* is relatively high. But when $sq_{\#}$ is too large and the fragment structure is too complex, its consideration of semantics and structure can hardly fulfill the requirement.
- Similarly, *ClstRec* uses the description of services to cluster them, and selects candidate services from suitable clusters for each sub-requirement. However, this method does not pay too much attention to structural information and can hardly guarantee the rationality of service composition. This causes *P* to be lower when $sq_{\#}$ is large and the fragment structure is more complex.

Figure 3(2) shows *R* of five methods. Overall, the *R* of *CFDY* is higher than that of other methods.

- For *CFDY*, as $sq_{\#}$ increases, its *R* first decreases and then increases. On the whole, its *R* is the highest compared to other methods. When $sq_{\#}$ is small, its structure is relatively simple, and too many exploration branches will add some unexpected services to the recommended fragment. Therefore, its *R* decreases. As $sq_{\#}$ becomes larger, the structure of the fragment becomes more complex and contains more branches. Therefore, further exploration of branches will increase *R* to some extent.
- Similarly, since *CSBR* lacks consideration of structural similarity, most of the services contained in its recommended fragment are unexpected, so its *R* is the lowest.
- Since *SP* has not further explored branches, its *R* is overall lower than that of *CFDY*. When $sq_{\#}$ is at 5–10, its *R* is higher than that of *CFDY*. Because it only explores a single path, in the case of more branches, the number of services in the fragment it recommends is much smaller than *CFDY*, which causes its *R* to be higher.
- For *ClsRec*, when $sq_{\#}$ is smaller, it has a higher *R*. Because it focuses on the similarity of functions, when $sq_{\#}$ is small and the fragment structure is simple, it can relatively accurately find expected services. However, when the fragment becomes complicated, this method can hardly effectively find all expected functions, due to the lack of comparison of structural similarity.

- *CDSR* first finds candidate services according to functional category, then uses historical usages and the coexistence time of services to construct fragments. This method considers the structure of fragments to a certain extent, but can hardly guarantee the necessity of the recommended services. Therefore, the recommended fragments contain a large number of unexpected services, which leads to a lower *R*.

As shown in Figure 3(3), the *F1* of five method decreases as $sq_{\#}$ increases.

- Compared with other methods, *CFDY* has the highest *F1*. Because it considers the structural similarity of fragments while considering semantic similarity. As the $sq_{\#}$ increases, the requirement of a user becomes more and more sophisticated, which leads to more complex selection of candidate services, and more complex fragment discovery and recommendation. Its *F1* is the highest when $sq_{\#}$ is 2, indicating that its recommendation effect is the best when the recommended fragment contains two services and their relationships. This value does not reach 1, because the function descriptions of some services are too similar, resulting in too high similarity of their topic vectors, so that they can hardly be accurately distinguished when selecting candidate services. This is an inevitable problem of *LDA* model.
- For *SP*, due to its low applicability when the fragment structure is complex, its *F1* is lower than that of *CFDY*. The other three methods divide services into different categories, clusters or packages according to their functions, and use semantic similarity to select candidate services. They lack the comparison of structural similarity. In contrast, both *CDSR* and *ClsRec* use historical relations between services to calculate the similarity in fragment structure, while *CSBR* only considers the feasibility of combinations in terms of functional similarity, which leads to the lowest *F1*.

4.4.4 Comparison on trd_U

We estimate the influence of different trd_U for bioinformatic workflow fragment discovery and the results of *P*, *R* and *F1* are shown in Figure 4 when trd_U is set to 0.78, 0.80, 0.82, 0.84, 0.86 and 0.88, respectively. tp_{ns} is set to 43, tp_{nd} is set to 10 and $k_{\#}$ is set to 2. Since the test set contains various samples with different $sq_{\#}$, the final result is the average of all test results.

The results in Figure 4(1) show that *P* of *CFDY* is the highest overall compared to other methods.

- Similarly, when semantic similarity is considered, *CFDY* has more exploration branches compared to *SP*, so the recommended fragment contains more expected services. Specifically, when trd_U is higher, the number of candidate services that can be selected decreases, and the number of expected services that are missing in the recommended fragment increases. This results in *P* getting smaller and smaller as trd_U increases. Especially for *CFDY*, *P* at 0.8 is greater than that at 0.78, which is caused by the uneven distribution of service in S^2KG and the large difference in in-degree and out-degree of them.
- Since *CSBR* doesn't consider the structural similarity much, its *P* is the lowest among all methods. It only relies on the functional similarity between services to discover a crossing-workflow fragment. When trd_U is higher, the number of candidate services for selection decreases, which affects its recommendation effect. Compared with *CSBR*, although *SP* considers the similarity of the fragment structure to a certain extent, it does not further explore branches and its accuracy is only higher than that of *CSBR*. Compared with the above two methods, *ClsRec* and *CDSR* have higher *P*. Generally, they adopt the clustering and classification to compare the functional similarity of fragments, and also apply historical usages to evaluate the structural similarity of fragments. Therefore, they are more effective than the methods that only consider semantic similarity. However, lacking the exploration of social relations, they are not as effective as *CFDY*.

Figure 4(2) shows the comparison of *R*. Similarly, *R* of *CFDY* is the highest, while *R* of *CSBR* is much lower than the other four methods. The difference is that, compared with *P*, as trd_U increases, the *R* of five methods increases.

- As the threshold increases, the candidate services become more similar and these services are more likely to be expected services. Some unexpected services are filtered out and the expected services are more likely to be included in the recommended fragment. For *CFDY*, the variation of trd_U affects the selection of its candidate services. However, compared with other methods, the consideration of social information on the discovery of crossing-workflow fragments can ensure the functional similarity and structural rationality of fragments to a certain extent and a better effect can be obtained.
- For *CSBR*, since it pursues more semantic similarity without considering the fragment structure, and does not consider the structural matching between services, its recommended fragment contains more unexpected services than other methods, which leads to its *R* is the lowest. *CDSR* uses historical usages information to ensure the rationality cross-workflow fragment structure. As a result, the recommended fragment contains a relatively high number of expected services, which results in a higher *R* than that of *CSBR*.
- Similarly, because *SP* and *ClsRec* add the similarity evaluation of the recommended cross-workflow

fragment structure, their *R* are higher than that of *CSBR*. The difference in the recommended effects of *SP*, *ClsRec* and *CDSR* is caused by their different calculation methods of functional similarity. In addition, the fragment complexity recommended by *SP* is lower than other methods and its fragment contains a relatively small number of services, which is part of the reason for its high *R*.

Finally, the comparison results of *F1* of the five methods are shown in Figure 4(3). This figure shows that the *F1* of each method decreases according to the changes of *P* and *R*. In general, *CFDY* has the highest *F1* and *CSBR* has the lowest one.

- For *CFDY*, as the trd_U increases, there are fewer connections that can meet the requirements, which leads to some feasible solutions to be ignored, thereby reducing *P*. At the same time, the reduction of the candidate set can increase the possibility of selecting the expected services, so that *R* increases. However, in combination, the increase in *R* is less than the decrease in *P*, so *F1* decreases. Since the *P* of *CFDY* at 0.8 is greater than that at 0.78, the *F1* of *CFDY* at 0.80 is higher than that at 0.78.
- Due to the lack of comparison of structural similarity in *CSBR*, its *F1* is the lowest. It shows that structural similarity is an important factor in cross-workflow fragment discovery and recommendation. Blindly pursuing functional similarity while ignoring structural similarity cannot achieve better recommendation results. Compared with *CSBR*, the other three methods leverage some structural information, thereby obtaining better *F1*. But compared with *CFDY*, they lack the exploration of the social relations between services, so *F1* is lower.

A higher *F1* of *CFDY* indicates that reasonable social information can improve the effectiveness of cross-workflow fragment discovery and recommendation to some extent. In fact, the representation of the functional domain of a service can be enhanced by using social information. In addition, author information can be used to reveal the hidden relationships between services. Therefore, it has a positive impact on fragment discovery and recommendation.

5 Related works

5.1 Social-aware workflow fragment discovery

Workflow fragment recommendation is an important research problem in the field of service computing (Coleman et al., 2022). It can shorten development cycles and reduce the cost by recommending suitable services and workflow fragments for users (Almarimi et al., 2019) from an open, large-scale library of Web services (Modi and Garg, 2019). In the past, profiles of services and workflows were used as the only guide for users to discover workflow fragments. However, with the development of social network service (SNS), traditional service repositories have become increasingly social, and contain a wealth of social information reflecting the social connections of developers and services (Bastami et al., 2019). This social information can also have an impact on workflow fragment recommendation, whereas existing approaches did not take full advantage of this complex social information currently.

Authors (Gu et al., 2021) propose a service package recommendation model (*CSBR*) based on a semantic service package repository by mining existing workflows. Using the degree of service co-occurrence, the correlation between service and workflow is mined. Specifically, reusable service packages composed of multiple collaborative services are annotated with composite semantics instead of their original semantics. Based on the semantic service pack repository, *CSBR* can recommend service packs that cover the functional requirements of workflow fragments as completely as possible. However, this approach discusses only some social properties and lacks further exploration of social relations, making it difficult to reveal the implicit relations between services.

Xia et al. (2015) used the categories of services to construct workflow fragments. They propose a category-aware distributed service recommendation (CDSR) model based on a distributed machine learning framework. Experiments on real data sets prove that the proposed method not only achieves a significant improvement in accuracy, but also enhances the diversity of recommendation results. However, this method ignores the relations between services and can hardly guarantee the structural similarity of the recommended workflow fragments.

Yao et al. (2014) proposed a ReputationNet to facilitate the workflow fragment discovery. Based on the ReputationNet, the reputations of services and its developers are calculated and represented. According to this, the services and workflows that have better qualities can be recommended to users to satisfy their sophisticated and complicated business requirements. This method utilizes the social attribute reputation, which can improve the efficiency of fragment recommendation to a certain extent. However, many other social information, such as social relations which can promote users to mine latent knowledge, have not been considered.

Zhu et al. (2021) proposed a new model SRaSLR, which is a type of social-aware service label recommendation model. There are invocation and dependency relations between services, and these relations make services naturally constitute a service social network. The authors combine the textual information in service profiles and the social network relations between services. Based on the feature fusion of two perspectives, a model based on deep learning is constructed. Authors conduct a lot of experiments on real-world

Programmable Web data set, and the experimental results show that the use of social relations can improve the performance of recommendation.

Khelloufi et al. (2021) argued that combining users' social characteristics can improve the efficiency of services recommendation and help us provide context-aware services. Therefore, they exploit the social relationships defined in SIoT to build service recommendations among devices, and thus, to enhance service discovery and composition. They propose a SIoT-based service recommendation framework in which devices inherit social relationships from their owners to provide socially aware service recommendations. A boundary-based community detection algorithm is proposed to form a community of socially connected devices.

Kalaï et al. (2018) adopted the social information about the users and the profiles about services to build a social-aware graph for services recommendation. The widespread use of social media provides a large amount of social information for service repositories. Using social information, many user relationships can be extracted for capturing implicit relationships between services. For example, two users, who are friends with each other, may be interested in similar service features. Based on the interests of a user and his friends, personal service recommendations can be provided. However, workflow fragments that can accomplish complex requirements may be preferable to users than recommending a single service that can accomplish simple and specific tasks for them.

Liang et al. (2016) proposed a new framework to effectively discover appropriate services by combining social media information. Specifically, they propose different methods to measure the four social factors collected from Twitter that semantic similarity, popularity, activity and decay factors. Qiao et al. (2019) proposed a recommendation algorithm based on knowledge graph representation learning, which embeds the entities and relations of knowledge graph into a low-dimensional vector space. These methods consider some social attributes in service recommendation, reflecting the importance of social information in recommendation work. However, they mainly recommend a single service to users, and can hardly be used to discover workflow fragments to fulfill the complex requirements prescribed by certain users.

Based on the various types of data in service repositories, underlying logical relations among them can be found to facilitate workflow fragment discovery and recommendation (Wang et al., 2019). Authors propose a fine-grained knowledge graph (DUSKG) to represent the information about users, services and service value feature (VF) and their relations. Based on the DUSKG, the VFs that a service has, the VFs which a user is interested in, and the relations between users and services can be expressed intuitively. Leveraging the DUSKG, five methods are adopted to recommend reasonable single services. However, this method also ignores the importance of workflows which can accomplish complex tasks.

5.2 Semantics-based workflow fragment discovery

Techniques have been developed to recommend workflow fragments from a functional perspective (Hao et al., 2019). Conforti et al. (2016) proposed a technique for automatically discovering hierarchical workflow fragments containing interrupted and non-interrupted boundary services markers. This technique uses approximate functions and contains dependency discovery techniques to extract the process-subprocess hierarchy. Profiles and service invocation relations are used for workflow fragment discovery. However, this method has not yet considered the social information that has an impact on the workflow fragment recommendation, and the information in the repository is not considered comprehensively.

Since the profiles of services are static and the development process is iterative (Huang et al., 2012), Modi and Garg (2019) proposed a method to update the profile of a single service leveraging the description of related workflows. Supplementary information can update the application scenario of a service and optimize its profile. Using this approach, the accuracy of the functional description of a service can be improved and the available services can be recommended to the user. However due to the limitations of functionality, a single service may not accomplish sophisticated and complicated requirements. Wang et al. (2017) proposed a method to extract fine-grained service value features and distributions for personalization service recommendation. By analyzing comments, the most interest aspects of a user can be learned. According to them, the similarity of these features and the descriptions of services are calculated and services with high similarity will be recommended to the user. However, the application scenarios of a single service are limited, since a single service can hardly satisfy the user's requirement as well as implement the user's complex functions completely. This approach lacks to explore the impact of social information and social association on workflow fragment recommendation.

Zhong et al. (2016) and (Hao et al. (2017) extracted the valued information from workflow description to narrow gaps between developers and users. The application scenarios are adopted to supply the description of services to emphasize their functionalities. The *LDA* model is adopted to represent the semantic functions of services. Based on reconstructed descriptions of services, the similarity between services and queries can be improved. However, the similarity is not the only metric that should be considered. Other metrics (Sun et al., 2019), for example, the quality of services (Li et al., 2019), should also be considered in the workflow fragment discovery procedure, so as to guarantee the reliability of the workflow fragments.

Zhou et al. (2018) proposed a method for workflow fragment recommendation which both consider the semantic information of workflows and the hierarchical relations of services. The clustering approach is adopted to cluster the hierarchical structure according to

the semantic information, so that the services and workflows with similar functions are in the same group as much as possible. However, this method only considers the invoking relationship between services and does not consider the impact of social connections on workflow fragment recommendation. In fact, these social relations emerge in large numbers in the repositories and also affect the composition of services to a certain extent.

Many services can provide similar functionality, and it is difficult for users to find the service they want (Ren and Wang, 2018). In the workflow fragment recommendation, whether two services can cooperate is an important problem (Lissandrini et al., 2018). The factors that affect service composition usually include two types, positive and negative links. Ni et al. (2015) leveraged tags and both positive and negative links to find service patterns. In addition to positive links of services which facilitate workflows fragment construction, several negative links between services are found, which are strangling service composition. The links between two credible services that have never been cooperated and the links between two services that have been created for a long time but never cooperated together are negative links. Although the consideration of negative links can guide whether two services can be combined, the consideration of positive links is relatively simple. This method explores the influence of social attributes and historical usage on workflow fragment recommendation, but ignores the role of social connections in recommendation work, and does not explore the impact of social relations on workflow segment discovery and recommendation.

5.3 Syntax-based workflow fragment discovery

The syntax-based method focuses on the structure of workflows and the problem of service composition is regarded as a service matching problem. The matching of interface parameters is adopted as the most important metric to promote the composition. Niu et al. (2016) modeled the workflow fragment discovery problem as an uncertain web service composition planning problem. A total of two new uncertain planning algorithms using heuristic search are proposed, called UCLAO* and BHUC, which use the similarity of service interface parameters to solve the U-WSC planning problem of reduced state space, thereby improving the efficiency of finding service portfolio solutions. Empirical experiments are carried out based on running examples in E-commerce applications and large-scale simulation data sets. However, it does not take the level of expertise of different users into account. In fact, there may exist users who do not know the details of the interface, and may not be able to provide input or output parameters. Moreover, the lack of considering service semantics and social associations may not ensure the correctness of the workflow from a functional point of view.

Due to the fast increase of web services over the Internet, Lin et al. (2012) proposed a backward planning method to discover reasonable workflow fragments in a large-scale web service repository based on the lowest cost. The authors exploit the similarity of input and output parameters to construct service groups for facilitating service search. Also, a backward strategy is used to reduce the search space, in order to improve the computational efficiency during workflow construction. However, this approach also neglects the important functional semantics of services and lacks the exploration of the impact about social association among services on their combination.

Liu et al. (2014) proposed a workflow-based framework for workflow fragments discovery. It not only uses the matching degree of the interface parameters to facilitate service composition but also employs a data-centric composition principle that the parameters matching are based on the tag-based semantics. Also, the semantics of service are determined by the folksonomy. The authors first used the related tags to stand for parameters and then constructed workflows based on them. This approach considers the semantic information of the service as well and can better reflect the functionality of the services. Therefore, it can facilitate the combination of services and the recommendation of workflow fragments from a functional perspective. In fact, besides labels, there is other rich semantic information in the repository that can facilitate the construction of workflow fragments. However, these semantic information are not used. Meanwhile, social repositories contain a rich variety of social information and social correlations among items, and these social correlations are not considered in this approach.

6 Conclusion

Considering the knowledge-intensiveness, effort-consuming, and error-proneness when constructing a novel bioinformatic workflow from scratch, discovering and reusing the best practices in legacy workflows is promising when it comes to accomplishing similar tasks. Traditional methods are proposed to discover appropriate workflow fragments depending on their profiles or partial social information in service repositories. However, social relations between developers have not been explored extensively. To capture these relations, this study constructs a knowledge graph S^2KG that includes two types of entities and three types of relations. Based on S^2KG, we propose a bioinformatic workflow fragment discovery mechanism, where we identify positive and negative links for

service composition through analyzing their co-invocation possibilities and co-developer relations. A SINet is formed by *isInk* relations in S^2KG to facilitate single service discovery. Finally, the *Yen*'s method is adopted to construct bioinformatic workflow fragments with respect to user's requirements. Experimental results demonstrate that our method performs better than the state-of-the-art techniques with higher accuracy and efficiency.

Data availability statement

Publicly available data sets were analyzed in this study. This data can be found at: https://www.myexperiment.org/workflows.

Author contributions

JD: conceptualization, methodology, software, data curation, and writing—original draft preparation. ZZ: supervision, formal analysis, visualization, and writing—review and editing. XX: resources, data curation, investigation, and writing—review and editing. DZ: methodology, supervision, validation, and writing—review and editing. SC: resources, supervision, validation, and writing—review and editing.

Funding

This work was supported partially by the National Key R&D Program of China (2019YFB2101803) and partially by the National Natural Science Foundation of China (42050103).

Conflict of interest

SC was employed by Wuda Geoinformatics Co., Ltd.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest

The handling editor (initials) declared a past coauthorship with the authors (ZZ, XX).

Publisher's note

References

Almarimi, N., Ouni, A., Bouktif, S., Mkaouer, M. W., Kula, R. G., Saied, M. A., et al. (2019). Web service api recommendation for automated mashup creation using multi-objective evolutionary search. *Appl. Soft Comput.* 85, 105830. doi:10.1016/j.asoc.2019.105830

Bai, B., Fan, Y., Tan, W., and Zhang, J. (20172017). IEEE, 124–131.Sr-lda: Mining effective representations for generating service ecosystem knowledge maps*IEEE Int. Conf. Serv. Comput.*

Bastami, E., Mahabadi, A., and Taghizadeh, E. (2019). A gravitation-based link prediction approach in social networks. *Swarm Evol. Comput.* 44, 176–186. doi:10.1016/j.swevo.2018.03.001

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. *J. Mach. Learn. Res.* 3, 993–1022.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *J. Stat. Mech.*, P10008. doi:10.1088/1742-5468/2008/10/p10008

Brandt, C., Krautwurst, S., Spott, R., Lohde, M., Jundzill, M., Marquet, M., et al. (2021). porecov-an easy to use, fast, and robust workflow for sars-cov-2 genome reconstruction via nanopore sequencing. *Front. Genet.* 12, 711437. doi:10.3389/fgene.2021.711437

Coleman, T., Casanova, H., Pottier, L., Kaushik, M., Deelman, E., da Silva, R. F., et al. (2022). Wfcommons: A framework for enabling scientific workflow research and development. *Future Gener. Comput. Syst.* 128, 16–27. doi:10.1016/j.future.2021.09.043

Conforti, R., Dumas, M., García-Bañuelos, L., and La Rosa, M. (2016). Bpmn miner: Automated discovery of bpmn process models with hierarchical structure. *Inf. Syst.* 56, 284–303. doi:10.1016/j.is.2015.07.004

Fischer, M., Hofmann, A., Imgrund, F., Janiesch, C., and Winkelmann, A. (2021). On the composition of the long tail of business processes: Implications from a process mining study. *Inf. Syst.* 97, 101689. doi:10.1016/j.is.2020.101689

Gkortzis, A., Feitosa, D., and Spinellis, D. (2021). Software reuse cuts both ways: An empirical analysis of its relationship with security vulnerabilities. *J. Syst. Softw.* 172, 110653. doi:10.1016/j.jss.2020.110653

Goble, C. A., Bhagat, J., Aleksejevs, S., Cruickshank, D., Michaelides, D., Newman, D., et al. (2010). myexperiment: a repository and social network for the sharing of bioinformatics workflows. *Nucleic Acids Res.* 38, W677–W682. doi:10.1093/nar/gkq429

Gu, Q., Cao, J., and Liu, Y. (2021). Csbr: A compositional semantics-based service bundle recommendation approach for mashup development. *IEEE Trans. Serv. Comput.*, 1. doi:10.1109/TSC.2021.3085491

Hao, Y., Fan, Y., Tan, W., and Zhang, J. (2017). "Service recommendation based on targeted reconstruction of service descriptions," in *2017 IEEE international conference on web services* (IEEE), 285–292.

Hao, Y., Fan, Y., and Zhang, J. (2019). Service recommendation based on description reconstruction in cloud manufacturing. *Int. J. Comput. Integr. Manuf.* 32, 294–306. doi:10.1080/0951192x.2019.1571242

Herbold, S., Amirfallah, A., Trautsch, F., and Grabowski, J. (2021). A systematic mapping study of developer social network research. *J. Syst. Softw.* 171, 110802. doi:10.1016/j.jss.2020.110802

Huang, K., Fan, Y., and Tan, W. (2012). "An empirical study of programmable web: A network analysis on a service-mashup system," in *2012 IEEE 19th international conference on web services* (IEEE), 552–559.

Kalaï, A., Zayani, C. A., Amous, I., Abdelghani, W., and Sèdes, F. (2018). Social collaborative service recommendation approach based on user's trust and domain-specific expertise. *Future Gener. Comput. Syst.* 80, 355–367. doi:10.1016/j.future.2017.05.036

Khelloufi, A., Ning, H., Dhelim, S., Qiu, T., Ma, J., Huang, R., et al. (2021). A social-relationships-based service recommendation system for siot devices. *IEEE Internet Things J.* 8, 1859–1870. doi:10.1109/jiot.2020.3016659

Li, C., Wang, H., Zhang, Z., Sun, A., and Ma, Z. (2016). Noncoding RNAs in human cancer: One step forward in diagnosis and treatment. *Brief. Funct. Genomics* 15, 165–166. doi:10.1093/bfgp/elw004

Li, S., Huang, J., Cheng, B., Cui, L., and Shi, Y. (2019). Fass: A fairness-aware approach for concurrent service selection with constraints. In IEEE International Conference on Web Services. (IEEE), 255–259.

Liang, T., Chen, L., Wu, J., Xu, G., and Wu, Z. (2016). Sms: A framework for service discovery by incorporating social media information. *IEEE Trans. Serv. Comput.* 12, 384–397. doi:10.1109/tsc.2016.2631521

Lin, S.-Y., Lin, G.-T., Chao, K.-M., and Lo, C.-C. (20122012). A cost-effective planning graph approach for large-scale web service composition. *Math. Problems Eng.*, 21. doi:10.1155/2012/783476

Lissandrini, M., Mottin, D., Palpanas, T., and Velegrakis, Y. (2018809). "Multi-example search in rich information graphs," in *2018 IEEE 34th international conference on data engineering* (IEEE)–820.

Liu, X., Huang, G., Zhao, Q., Mei, H., and Blake, M. B. (2014). imashup: a mashup-based framework for service composition. *Sci. China Inf. Sci.* 57, 1–20. doi:10.1007/s11432-013-4782-0

Modi, K. J., and Garg, S. (2019). A qos-based approach for cloud-service matchmaking, selection and composition using the semantic web. *J. Syst. Inf. Technol.* 21, 63–89. doi:10.1108/jsit-01-2017-0006

Nepal, S., Malik, Z., and Bouguettaya, A. (20092009). Reputation propagation in composite services. *IEEE Int. Conf. Web Serv.*, 295–302.

Newman, M. E. (2006). Modularity and community structure in networks. *Proc. Natl. Acad. Sci. U. S. A.* 103, 8577–8582. doi:10.1073/pnas.0601602103

Ni, Y., Fan, Y., Tan, W., Huang, K., and Bi, J. (2015). Ncsr: Negative-connection-aware service recommendation for large sparse service network. *IEEE Trans. Autom. Sci. Eng.* 13, 579–590. doi:10.1109/tase.2015.2466691

Niu, S., Zou, G., Gan, Y., Zhou, Z., and Zhang, B. (2016). "Uclao* and bhuc: Two novel planning algorithms for uncertain web service composition," in *2016 IEEE international conference on services computing* (IEEE), 531–538.

Qiao, X., Cao, Z., and Zhang, X. (2019). Web service recommendation technology based on knowledge graph representation learning. In Journal of Physics: Conference Series, 1213. Bristol: IOP Publishing, 042015.

Ren, L., and Wang, W. (2018). An svm-based collaborative filtering approach for top-n web services recommendation. *Future Gener. Comput. Syst.* 78, 531–543. doi:10.1016/j.future.2017.07.027

Robinson, I., Webber, J., and Eifrem, E. (2015). *Graph databases: New opportunities for connected data*. Sebastopol: O'Reilly Media, Inc.

Rosa, M. J., Ralha, C. G., Holanda, M., and Araujo, A. P. (2021). Computational resource and cost prediction service for scientific workflows in federated clouds. *Future Gener. Comput. Syst.* 125, 844–858. doi:10.1016/j.future.2021.07.030

Starlinger, J., Brancotte, B., Cohen-Boulakia, S., and Leser, U. (2014). Similarity search for scientific workflows. *Proc. VLDB Endow.* 7, 1143–1154. doi:10.14778/2732977.2732988

Sun, M., Zhou, Z., Wang, J., Du, C., and Gaaloul, W. (2019). Energy-efficient iot service composition for concurrent timed applications. *Future Gener. Comput. Syst.* 100, 1017–1030. doi:10.1016/j.future.2019.05.070

Wang, H., Chi, X., Wang, Z., Xu, X., and Chen, S. (2017). "Extracting fine-grained service value features and distributions for accurate service recommendation," in *2017 IEEE international conference on web services* (IEEE), 277–284.

Wang, H., Wang, Z., Hu, S., Xu, X., Chen, S., Tu, Z., et al. (2019). Duskg: A fine-grained knowledge graph for effective personalized service recommendation. *Future Gener. Comput. Syst.* 100, 600–617. doi:10.1016/j.future.2019.05.045

Xia, B., Fan, Y., Tan, W., Huang, K., Zhang, J., Wu, C., et al. (2015). Category-aware api clustering and distributed recommendation for automatic mashup creation. *IEEE Trans. Serv. Comput.* 8, 674–687. doi:10.1109/tsc.2014.2379251

Yao, J., Tan, W., Nepal, S., Chen, S., Zhang, J., De Roure, D., et al. (2014). Reputationnet: Reputation-based service recommendation for e-science. *IEEE Trans. Serv. Comput.* 8, 439–452. doi:10.1109/tsc.2014.2364029

Yao, L., Wang, X., Sheng, Q. Z., Benatallah, B., and Huang, C. (2021). Mashup recommendation by regularizing matrix factorization with api co-invocations. *IEEE Trans. Serv. Comput.* 14, 502–515. doi:10.1109/tsc.2018.2803171

Yen, J. Y. (1971). Finding the k shortest loopless paths in a network. *Manag. Sci.* 17, 712–716. doi:10.1287/mnsc.17.11.712

Zhang, J., Pourreza, M., Lee, S., Nemani, R., and Lee, T. J. (2018). "Unit of work supporting generative scientific workflow recommendation," in *International conference on service-oriented computing* (Springer), 446–462.

Zhong, Y., Fan, Y., Tan, W., and Zhang, J. (2016). Web service recommendation with reconstructed profile from mashup descriptions. *IEEE Trans. Autom. Sci. Eng.* 15, 468–478. doi:10.1109/tase.2016.2624310

Zhou, Z., Cheng, Z., Zhang, L.-J., Gaaloul, W., and Ning, K. (2018). Scientific workflow clustering and recommendation leveraging layer hierarchical analysis. *IEEE Trans. Serv. Comput.* 11, 169–183. doi:10.1109/tsc.2016.2542805

Zhou, Z., Wen, J., Wang, Y., Xue, X., Hung, P. C., Nguyen, L. D., et al. (2020). Topic-based crossing-workflow fragment discovery. *Future Gener. Comput. Syst.* 112, 1141–1155. doi:10.1016/j.future.2020.05.029

Zhu, Y., Liu, M., Tu, Z., Su, T., and Wang, Z. (2021).Sraslr: A novel social relation aware service label recommendation model. In *2021 IEEE international conference on web services*. IEEE, 87–96.

frontiers | Frontiers in Genetics

TYPE Original Research
PUBLISHED 10 October 2022
DOI 10.3389/fgene.2022.964784

OPEN ACCESS

EDITED BY
Maurice H. T. Ling,
Temasek Polytechnic, Singapore

REVIEWED BY
Fan Jiang,
Xi'an University of Posts and Telecommunications, China
Guolin Sun,
University of Electronic Science and Technology of China, China

*CORRESPONDENCE
Lei Yu,
yuleiimu@sohu.com
Yucong Duan,
duanyucong@hotmail.com

SPECIALTY SECTION
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

RECEIVED 09 June 2022
ACCEPTED 21 September 2022
PUBLISHED 10 October 2022

CITATION
Yu L, Yu PS, Duan Y and Qiao H (2022), A resource scheduling method for reliable and trusted distributed composite services in cloud environment based on deep reinforcement learning. *Front. Genet.* 13:964784. doi: 10.3389/fgene.2022.964784

A resource scheduling method for reliable and trusted distributed composite services in cloud environment based on deep reinforcement learning

Lei Yu[1]*, Philip S. Yu[2], Yucong Duan[3]* and Hongyu Qiao[1]

[1]Department of Computer Science, Inner Mongolia University, Hohhot, China, [2]Department of Computer Science, University of Illinois at Chicago (UIC), Chicago, IL, United States, [3]College of Computer Science and Technology, Hainan University, Haikou, China

With the vigorous development of Internet technology, applications are increasingly migrating to the cloud. Cloud, a distributed network environment, has been widely extended to many fields such as digital finance, supply chain management, and biomedicine. In order to meet the needs of the rapid development of the modern biomedical industry, the biological cloud platform is an inevitable choice for the integration and analysis of medical information. It improves the work efficiency of the biological information system and also realizes reliable and credible intelligent processing of biological resources. Cloud services in bioinformatics are mainly for the processing of biological data, such as the analysis and processing of genes, the testing and detection of human tissues and organs, and the storage and transportation of vaccines. Biomedical companies form a data chain on the cloud, and they provide services and transfer data to each other to create composite services. Therefore, our motivation is to improve process efficiency of biological cloud services. Users' business requirements have become complicated and diversified, which puts forward higher requirements for service scheduling strategies in cloud computing platforms. In addition, deep reinforcement learning shows strong perception and continuous decision-making capabilities in automatic control problems, which provides a new idea and method for solving the service scheduling and resource allocation problems in the cloud computing field. Therefore, this paper designs a composite service scheduling model under the containers instance mode which hybrids reservation and on-demand. The containers in the cluster are divided into two instance modes: reservation and on-demand. A composite service is described as a three-level structure: a composite service consists of multiple services, and a service consists of multiple service instances, where the service instance is the minimum scheduling unit. In addition, an improved Deep Q-Network (DQN) algorithm is proposed and applied to the scheduling algorithm of composite services. The experimental results show that applying our improved DQN algorithm to the composite services scheduling problem in the container cloud environment can effectively reduce the completion time of the composite services. Meanwhile, the method improves Quality of Service (QoS) and resource utilization in the container cloud environment.

KEYWORDS

composite services, container cloud, deep reinforcement learning, service scheduling, artificial intelligence

1 Introduction

With the rapid development and popularity of the Internet, the number of network users is also increasing, but the resources of the data center decreased relatively. The development of cloud computing technology has led to a great convenience of information processing, and users can obtain reliable services through the cloud platform on a large number of data centers. However, as the composite service requested by the users are complex and diversified, the number of requests is increasing. Especially in the field of bioinformatics, biomedical research relies on a large amount of genomic and clinical data. Biomedical companies form a data chain on the cloud, and they provide services and transfer data to each other to form composite services. In such a dynamic environment, resource management and performance optimization have become a significant challenge for cloud and application providers, who not only consider user Quality of Service (QoS) but also consider the load balancing of the data center, resource utilization and problems such as energy consumption (Almansour and Allah, 2019). Therefore, an efficient and reasonable service scheduling method becomes essential for the cloud computing platform.

In addition, many cloud platforms currently use virtual machines as the underlying virtualization technology. Additional operating systems carried by virtual machines will bring performance losses to the cloud platform, and the startup speed of virtual machines is slow, so it is difficult for them to make rapid scaling responses to service load (Barik et al., 2016). As the virtualization technology at the operating system level, the container technology has minimal additional resource overhead, shorter startup and destruction time, and the performance of disk IO and CPU of the container is even close to that of the host (Joy, 2015). Therefore, it is considered to be a better solution for application distribution and deployment on the cloud platform (Bernstein, 2014). Most of the research on service scheduling is based on virtual machines, while the research on composite service scheduling based on container cloud environment is in the exploratory stage. Because the container has the characteristics of fast startup, strong migration ability, low-performance cost, and high resource utilization (Joy, 2015), it is of great value and significance to take the container as the virtualized computing resource of the cloud platform to solve the service scheduling problem. We need a model and an algorithm that can be applied to the container cloud environment to reduce the completion time of the composite service, satisfy the user service quality as much as possible, and improve the resource utilization target of the cloud platform.

Therefore, we proposed a novel composite service scheduling model and algorithm according to container instance mode which mixed reservation and on-demand. In addition, the DQN (Deep Q-Network) algorithm is improved by combining the three algorithms Dueling-DQN (Wang et al., 2016), Double-DQN (DDQN) (Van Hasselt et al., 2016), and Prioritized Experience Replay (PER) (Schaul et al., 2016). DDQN improved the training algorithm by decoupling action selection and value function evaluation. Although it is not entirely decoupled, it effectively reduced over-estimation and made the algorithm more robust. PER introduced a new learning mechanism to solve the sampling problem of experience replay and innovatively took Temporal Difference (TD) deviation as an essential consideration to ensure that important experience can be replayed first, and the priority experience replay was applied to DQN and DDQN. The learning efficiency is greatly improved. Dueling DQN is an improvement of the neural network structure, which can decouple the value and advantages of the DQN. Although the value function and the advantage function can no longer be perfectly represented as the value function and the advantage function in semantics, the accuracy of the strategy evaluation was improved, and it can be combined with other algorithms due to the strong versatility. Thus, the management of the DQN algorithm is improved. From the three levels of training algorithm, learning mechanism, and neural network structure, three improvements have been made based on DQN, but its implementation is more complex than these three algorithms. The improved DQN algorithm is used as the scheduling decision method under our model to reduce the completion time of the composite service and improve the user QoS and resource utilization of the cloud platform.

The contributions of this paper include: A new composite service scheduling model is built for container instance mode which mixed reservation and on-demand. The model considers many features, such as container storage, computing speed, network bandwidths and data streams of service output, etc. Furthermore, the model is suitable for Map-Reduce based services in distributed environments.

A new composite service scheduling algorithm is proposed, which can effectively reduce the completion time of the composite services. Meanwhile, the method improves Quality of Service (QoS) and resource utilization in the container cloud environment.

2 Related work

Cloud computing technology has greatly promoted the transformation of various industries and the development of

technological innovation. With the advancement of medical technology, the field of biomedicine has ushered in the era of big data (Yang et al., 2021). The application of cloud computing in biomedicine is becoming more and more perfect. Myers et al. (2020) developed an R package, LDlinkR, which leverages the computing resources of the cloud by harnessing the storage capacity and processing power of the LDlink web server to calculate computationally expensive LD statistics. Service scheduling, as an effective method to satisfy Quality of Service (QoS), which can rationally allocate resources and reduce energy consumption in cloud environment, has always been a research hotspot of scholars in various fields (Kyaw and Phyu, 2020). At the same time, scheduling in the cloud environment is a multi-constraint, multi-objective and multi-type optimization problem (Chen et al., 2019). Some traditional scheduling algorithms, such as Round-Robin (RR) scheduling algorithm and Least Connection (LC) algorithm, do not consider the actual load and connection status of the work node. Scheduling problem can be regarded as the problem of finding the optimal one or a group of computing resources in a limited set of computing resources under the condition of satisfying multiple constraint objectives. Heuristic algorithm is the most widely used method to solve such combinatorial optimization problems (Bernstein, 2014). The common ones are Ant Colony (AC) algorithm, Particle Swarm Optimization (PSO) algorithm, Genetic Algorithm (GA), etc. Therefore, many scholars are solving the problem of service scheduling in cloud platforms by optimizing and improving heuristic algorithms.

Panwar et al. (2019) combined Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS) algorithm and PSO algorithm to divide task scheduling into two phases, which reduces the makespan of tasks and improves resource utilization of cloud platform. Chen et al. (2019) modeled the cloud workflow scheduling problem as a multi-objective optimization problem that takes both execution time and execution cost into account, and proposed a multi-objective ant colony system based on the co-evolutionary multi-population and multi-objective framework, in which two ant colony algorithms were adopted to deal with the two objectives, respectively. Cui and Xiaoqing (2018) proposed a workflow tasks scheduling algorithm based on a genetic algorithm. It plays an optimal role in the execution time of the optimal allocation scheme. George et al. (2016) adopted the Cuckoo Search algorithm to complete the assignment of tasks with the optimization goal of minimizing the computation time of tasks. Ghasemi et al. (2019) proposed a workflow scheduling method based on the Firefly Algorithm (FA), aiming at minimizing the processing time and transmission cost of workflow.

Compared with traditional scheduling algorithms, heuristic algorithms have a stronger ability for exploration and optimization. The above improvements of heuristic not only inherited the advantages of heuristic algorithms in solving combinatorial optimization problems but also solved some problems of heuristic algorithms themselves to some extent. However, these algorithms still have some problems, such as the weight coefficients of resources according to subjective experiences, slow convergence, and easily falling into local optimal solutions.

Considering the uncertainty of user requests, the dynamic nature of computing resources, the heterogeneity of cloud platforms, and many other factors, it has higher requirements for cloud platform service scheduling strategy. In recent years, with the development of artificial intelligence-related technologies, Deep Reinforcement Learning (DRL) has shown strong perception and continuous decision-making ability when dealing with automatic control problems (Orhean et al., 2018), and many scholars have begun to apply it to resource allocation and service scheduling strategies in cloud environments. Li and Hu (2019) described job scheduling as a packing problem, used DRL algorithm to calculate the fitness of jobs and machine nodes, and selected reasonable machines for jobs according to the fitness. Finally, through experiments, it proved the superiority of deep reinforcement learning as a scheduling algorithm. Cheng et al. (2018) designed a two-level scheduler combining resource allocation and task scheduling based on Deep Q-Learning, which greatly reduced the energy consumption of the cloud platform while maintaining a low task rejection rate. Wei et al. (2018) proposed an intelligent QoS aware job scheduling framework based on Deep Q-Learning algorithm, which can effectively reduce the average response time of jobs under varying loads and improve user satisfaction. Meng et al. (2019) designed an adaptive online scheduling algorithm by combining reinforcement learning with DNN, which significantly improved the scheduling efficiency of server-side task queues. Ran et al. (2019) used the Deep Determining Policy Gradient (DDPG) algorithm to find the optimal task assignment scheme meeting the requirements of the Service Level Agreement (SLA). Zhang et al. (2019) proposed a parallel execution multi-task scheduling algorithm based on deep reinforcement learning. And compared with least connection and particle swarm optimization, this algorithm significantly reduces the completion time of the job. Dong et al. (2020) proposed a task scheduling algorithm based on DRL, which can dynamically schedule tasks that have priority relationships in the cloud server, thus minimizing the task execution time and effectively solving the task scheduling problem in the cloud manufacturing environment.

Based on the above work, both the heuristic algorithm and deep reinforcement learning algorithm show their respective advantages in solving scheduling problems in cloud environments. However, there are still some problems that

TABLE 1 Summary of reviewed papers related to the task scheduling in the cloud computing.

Algorithm	Core issues to be solved	Algorithm idea	Advantage
Dueling-DQN Wang et al. (2016)	Solved the problem that in some states, action is of low importance to the overall result, and distinguished the change of Q value caused by action and state	Improved the architecture, the idea of advantage was added to evaluate the advantage function Analyzed the advantages and disadvantages of state and action, respectively	Ensured that the relative ranking of the dominant functions of each action in this state remains unchanged Narrowed the range of Q value. Removed excess degrees of freedom. Improved the stability of the algorithm
DDQN Van Hasselt et al. (2016)	Solved the problem of overestimation in DQN algorithm	The idea of Double Q-learning is to reduce overestimations by decomposing the max operation in the target into action selection and action evaluation	More stable training results Reduced the error caused by variance
PER Schaul et al. (2016)	Changed the selection method of samples in experience replay Solved the problem of local optimization Offset the impact of sample distribution	Improved the experience buffer training strategy More robust Added weight to the original gradient update in SGD	More robust Improved the performance of DDQN Simple implementation
MOACS Chen et al. (2019)	Optimized execution time and cost	Two ant colonies are adopted to optimize execution time and execution cost, respectively A new pheromone update rule is designed. The CHS is proposed to ensure the quality of the other objective	MOACS has better global search ability, particularly when dealing with large-scale workflows MOACS can generate a solution with similar WET but lower WEC than the other approaches
TOPSIS–PSO Panwar et al. (2019)	Improved the execution time, maximum completion time, resource utilization, processing cost, and transmission time in the process of task scheduling	The task scheduling is performed in two phases TOPSIS method calculates the RC of VMs with respect to each task The PSO algorithm receives the calculated RC of each task which acts as FV of tasks (particles)	Improved average resource utilization Low processing cost Reduced makespan for tasks
Workflow tasks scheduling optimization based on genetic algorithm Cui and Xiaoqing (2018)	Applicable to cloud computing environment combining task characteristics and resource characteristics Reduced the execution cost of workflow task scheduling	Assigned priority to each task Workflow tasks were divided into different levels, and a two-dimensional coding method was designed A new genetic crossover and mutation operation were designed to produce new different offspring, so as to increase population diversity	Reduced workflow scheduling cost
FA Ghasemi et al. (2019)	Optimized the cost of executing the whole workflow and load balancing among workstations	The position of each firefly represents the feasible solution to a problem to be solved, and the brightness of the firefly represents the fitness of the firefly's position Each firefly flies towards a firefly that looks brighter than itself	Minimized the processing time Reduced transmission cost of workflow
An intelligent QoS-Aware Job Wei et al. (2018)	Met the QoS requirements of users	Learnt to make appropriate online job-to-VM decisions for continuous job requests directly from its experiences without any prior knowledge	Reduced the average response time of jobs under different loads. Improved user satisfaction

(Continued on following page)

TABLE 1 (*Continued*) Summary of reviewed papers related to the task scheduling in the cloud computing.

Algorithm	Core issues to be solved	Algorithm idea	Advantage
Scheduling and resource management algorithm for multi-user mobile-edge computing systems Meng et al. (2019)	The problem of delay-sensitive task scheduling and resource (e.g., CPU, memory) management on the server side in multi-user MEC scenario	Built a system that learns to manage resources directly from experience by using reinforcement learning with adaptive policy iteration represented *via* DNN. Designed a new reward function to reduce average slowdown and average timeout period of tasks in the queue	Reduced average slowdown and average timeout period of tasks in the queue Improved the scheduling efficiency of server-side task queue
DDPG Ran et al. (2019)	Model free strategy for learning continuous action	DDPG combines the ideas of DPG and DQN It used the experience replay and delayed update target network in DQN It can run in continuous action space based on DPG	DDPG can run in a continuous action space Solved the classical inverted pendulum control problem Met service level agreements
MDTS Zhang et al. (2019)	The problem of scheduling jobs with scalable parallel tasks in general parallel computing systems, where there is a demand to determine the task placement plan with the goal of minimizing the job completion time, the load imbalance value, and the total cost	Within each task-specific branch, there is a fully connected layer and an output layer	Reduced the job completion time and optimized the load balancing problem. Improved task scheduling performance. MDTS is superior to the raw DRL algorithm
Data-dependent tasks re-scheduling energy efficient algorithm Xiaoqing et al. (2018)	Reduced energy consumption in the data center	Set the task priority to the sum of the upper and lower values of the task Used the task priority to calculate the critical path and critical resources of the task graph Calculated the energy efficiency of each resource under the initial scheduling scheme	Reduced energy consumption in the data center
DRL-based algorithms Islam et al. (2021)	Satisfied generalization to optimize multiple objectives while capturing or learning the underlying resource or workload characteristics	Two DRL-based agents (DQN and REINFORCE) DQN: An ϵ-greedy policy was used that selects the greedy action with probability $1-\epsilon$ and a random action with probability ϵ REINFORCE: It worked by utilizing Monte Carlo roll-outs. After the collection step, the algorithm updates the underlying network using the updated policy gradient Trained them as scheduling agents in the TF-agent framework	Reduced both the total cluster VM usage cost and the average job duration
Sharer Liang et al. (2020)	Improved the efficiency of resource management in CMfg	The proposed model transformed metrics generated from the individual needs of multiple users into a multiobjective reward Proposed a blacklist mechanism and a narrow baseline to improve the learning performance of RL	Adapt to different conditions Converged quickly

have not been considered in some references when solving scheduling problems in cloud platforms. References (Almezeini and Hafez, 2017; Li and Hu, 2019; Panwar et al., 2019) only discussed a single service type without discussing the diversity of services and the correlation between services. References (Cui and Xiaoqing, 2018; Xiaoqing et al., 2018) gave the corresponding weight coefficients of each resource through subjective experience.

References (Zhang et al., 2019; Dong et al., 2020) did not take into account the transmission cost between resource nodes of the execution results of services in the actual scheduling process of composite services. In the actual environment, the data transmission time between sub-services affects the completion time and operation cost of composite services to some extent. With the increasing complexity of user requests and the increasing granularity of services, each service can be scheduled for parallel execution in multiple servers to reduce the response time of services and improve the quality of services for users. References (Orhean et al., 2018; Xiaoqing et al., 2018; Chen et al., 2019; Dong et al., 2021) did not consider the parallelism of services when discussing the problem of service scheduling. We compared some algorithms in Table 1.

In addition, most of the above studies took virtual machines as virtualized computing resources to study the problem of service scheduling, while containers have the advantages of simple deployment and fast startup speed, so it is of certain research significance and value to discuss the problem of service scheduling based on the container cloud environment.

3 System model

3.1 Problem description

Based on the container cloud environment, this section focuses on the scheduling method of composite services. In the initialization stage, a certain number of host nodes are set, and each host node initializes: 1) a certain number of reserved container instances with different configurations; 2) a certain number of on-demand containers. In the reserved mode, the container instance is in the startup state and uses the allocated resources for the scheduled services at any time. The container in on-demand mode is dormant initially and takes a period of time to be started. The composite service is defined as the three-level structure of "composite service, sub-service, instance." As the basic scheduling unit, the sub-service instance is scheduled to be executed in the container, which in essence represents the number of parallel execution of sub-services. In addition, the scheduling of sub-service instances and the starting of containers in on-demand mode are determined by the service scheduling algorithm.

3.2 Problem constraints

A composite service consists of multiple sub-services (hereinafter referred to as "Services") that have an association relationship, including the order of prior execution and data dependencies among the services. In addition, each service includes one or more service instances, and each service instance of the same service has the same physical performance requirements. A composite service can be represented by a directed acyclic graph, i.e., $CS = (SVC, E)$, where the finite set $SVC = \{svc_1, \ldots, svc_m\}$ indicates that a composite service contains $m(m \geq 1, m \in N^+)$ services. Each service has $n(n \geq 1, n \in N^+)$ service instances, denoted as $svc_i = \{st_i^1, \ldots, st_i^n\}(i \in m)$. The set of directed edges $E = \{(svc_i, svc_j)|1 \leq i, j \leq m, i, j \in N^+\}$ describes the relationship between services, (svc_i, svc_j) means that svc_i is the predecessor service of svc_j, and svc_j is called the successor service of svc_i. Only after all service instances of all precursor services of svc_j have been executed, svc_j is allowed to be scheduled and executed. The service without the precursor service is called the start service svc_{start}, and each composite services has at least one start service. Service without successor services is called end service svc_{end}. Similarly, each composite service has at least one end service. Each Roman character (e.g., I, II) represents the number of service instances contained in the corresponding service. This scenario is prevalent for Map-Reduce algorithms in distributed environments.

Each service instance will be scheduled to a container, and each service contains multiple service instances, which means that each service can be executed by multiple containers together. The characteristic definition of service svc_i can be denoted by Eq. 1, where cpu_i, mem_i, $disk_i$ represent the physical performance requirements of service svc_i, such as CPU, memory, and disk storage, respectively. $length_i$ denotes the length of the result data after the completion of the service execution; $inst_num_i$ denotes the number of service instances of service svc_i; $duration_i$ represents the expected execution time of the subservice svc_i.

$$svc_i = \{cpu_i, mem_i, disk_i, length_i, inst\_num_i, duration_i\}(i \in n) \tag{1}$$

As the smallest scheduling unit in a composite service, the service instances have the same physical resource requirements as the service it belongs to. All instances of the same service can be executed in parallel, and instances of each service are able to execute different binary files for Map-Reduce scenarios. Eq. 2 defines the *kth* service instance of svc_i.

$$st_i^k = \{k, cpu_i, mem_i, disk_i, length_i, duration_i\}(i \in n, k \in m) \tag{2}$$

3.3 Resource model

In the cloud platform, physical hosts are the infrastructure that truly provides physical resources such as CPU and memory for containers and services. All hosts in a host cluster are denoted

as $H = h_1, \ldots, h_P$, where p is the number of hosts in the cluster. $h_x(x \in p)$ represents the xth host in the host cluster, and the definition of h_x is shown in Eq. 3.

$$h_x = \{hid, cpu\_cap_x, mem\_cap_x, disk\_cap_x, bw\_cap_x, container\_num_x, cpu_x, mem_x, disk_x, bw_x\}(x \in p) \quad (3)$$

where hid represents the unique ID of the host. And cpu_cap_x, mem_cap_x, $disk\_cap_x$, bw_cap_x, respectively represent the CPU capacity, memory capacity, disk storage capacity, and bandwidth capacity of the host. $container\_num_x$ represents the maximum number of containers that can be allocated by the host h_x. cpu_x, mem_x, $disk_x$, bw_x respectively represent the remaining amount of the host's CPU, memory, disk storage, and bandwidth.

In addition, all containers in the cluster can be represented by the set $C = \{c_1, \ldots, c_q\}$, where q is the number of containers. $c_y(y \in q)$ represents the physical performance state of the yth container, and the definition of c_y is shown in Eq. 4.

$$c_y = \{cid_y, hid_y, cpu\_cap_y, mem\_cap_y, disk\_cap_y, bw_y, cpu_y, mem_y, disk_y, act_y, act\_time_y\}(y \in q) \quad (4)$$

where cid_y represents the container ID, which is the unique identifier of the container. hid_y represents the host ID to which the container c_y belongs. cpu_cap_y, mem_cap_y, $disk\_cap_y$, bw_y respectively represent the CPU capacity, memory capacity, disk capacity, and bandwidth capacity of the container c_y. cpu_y, mem_y, $disk_y$, respectively represent the remaining amount of the container's CPU, memory, and disk during operation. act_y is the judgment flag, which indicates whether the container c_y is already in the state of the host. If $act_y = 1$, means that the container c_y is in the running state, and $act_y = 0$ means that the container c_y is in the dormant state. act_time_y represents the startup time of the container.

In order to compare and analyze resource utilization from three dimensions of CPU, memory, and disk, UST_i^k is defined as the resource utilization after each service instance is scheduled. The definition of average resource utilization AVUST is shown in Eq. 5.

$$AVUST = \frac{\sum_{i=1}^{m}\sum_{k=1}^{n} UST_i^k}{number\ of\ service\ instances} \quad (5)$$

3.4 Scheduling model

Before all composite services are scheduled, the hosts and containers in the data center need to be initialized. In the initialization phase, a series of physical hosts with different configurations are first created, and each host is allocated with $container\_num_x$ containers, including different configurations of reserved and on-demand containers. The containers in the reservation mode can run the scheduled service instances at any time based on the allocated resources. The containers in the on-demand mode are in the dormant state by default, which occupies a certain amount of physical resources, but there are no remaining amount of resources. The resource state of the containers in the on-demand mode is shown in Eq. 6.

$$\begin{cases} cpu\_cap_y > 0 \\ mem\_cap_y > 0 \\ disk\_cap_y > 0 \\ bw_y > 0 \\ cpu_y = 0 \\ mem_y = 0 \\ disk_y = 0 \end{cases} \quad (6)$$

Constraints must be satisfied to schedule the service to the container for execution. When the service instance st_i^k is scheduled to the container c_y, the physical resource requirements of the service instance st_i^k must not be greater than the corresponding physical resource capacity of the container c_y, otherwise it will wait for the right resources to execute. Therefore, the constraint condition that needs to be met to dispatch the service instance st_i^k to the container c_y is shown in Eq. 7.

$$\begin{cases} cpu_k \le cpu\_cap_y \\ mem_k \le mem\_cap_y \\ disk_k \le disk\_cap_y \end{cases} \quad (7)$$

When a service svc_i is ready, all service instances of the service can be scheduled to the containers for execution one by one within the same scheduling time window. However, the resource status of the container changes from time to time as the service scheduling progresses. When the service instance is scheduled to the appropriate container, it will not be executed immediately. Because the following three steps are required:

(1) First, the status of the selected container needs to be determined. If the container has already been started, that is, $act_y = 1$, then ignore this step. Otherwise, $act_y = 0$, start the container, which will consume the time of act_time_y.
(2) After the completion of step one, it is necessary to wait for the execution result of the precursor service to be transmitted to the container. The data transmission time is related to the result data length after the execution of the precursor subservice, the bandwidth of the container, and the host. Since the precursor service has multiple service instances, each service instance will be scheduled to run in a container. It can be understood that each service can be scheduled to run in multiple containers, so it is necessary to calculate the minimum transmission time of the result data from the container scheduled by the precursor service to the container where the current service instance is located. The data transmission time between containers in the same host is negligible. The data transmission time between different hosts is directly related to factors such as container bandwidth and data length. The data transmission time is shown in Eq. 8.

$$transTi^k(c_u, c_v) = \begin{cases} 0, & u = v \ or \ hid_u = hid_v \\ ratio, & other \end{cases}$$
$$ratio = \frac{length_i}{min\ (bandwidth_u,\ bandwidth_v)} \tag{8}$$

(3) In addition to the data transmission time, it is necessary to wait for the remaining amount of the physical resources of the container to meet the physical resource requirements of the service instance itself. Record the waiting resource time of the service instance st_i^k in the container c_y as wr_i^k.

Based on the above three steps, it can be concluded that after the service instance st_i^k is scheduled, the period before execution is the total waiting time of the service instance TW_i^k:

$$TW_i^k = \begin{cases} transTi^k + wr_i^k, & act_y = 1 \\ act\_time_y + transTi^k + wr_i^k, & act_y = 0 \end{cases} \tag{9}$$

As mentioned above, the execution of the service is finished when all the instances of the service svc_i are executed. Therefore, the response time T_i of the service svc_i should be denoted as:

$$T_i = \max_k \left(T_i^k\right) \tag{10}$$

Taking the submission time of the composite services as the earliest start execution time T_{start} and the completion time T_{end} of the last service instance in the sub-service as the completion time of the composite service, thus the actual completion time TC of the entire composite service is denoted by Eq. 11.

$$TC = T_{end} - T_{start} \tag{11}$$

In order to denote the expected completion time of the composite services more conveniently, the composite service is divided into layers according to the execution order of the service. The start sub-service is placed in the first layer, and the end sub-service is placed in the last layer.

The service completion time of each level is the response time of the service with the longest response time in the level, as shown in Eq. 12, where l represents the level and u represents the number of services contained in the level.

$$TL_l = \max_u (T_i) \tag{12}$$

Define the maximum expected completion time for an entire composite service as:

$$TE = 2 \sum_v TL_l \tag{13}$$

The interaction between the user and the cloud platform takes the whole composite service as the unit, and the user can set the desired QoS demand when sending the request. The completion time of the composite service is an important QoS indicator for users, so this paper takes the maximum expected completion time of the composite services TE as the user's QoS demand. Eq. 14 indicates whether the user's demand QoS can be met:

$$success(CS) = \begin{cases} 1, & TC \le TE \\ 0, & else \end{cases} \tag{14}$$

For cloud and service providers, the goal of service scheduling is to meet users' QoS requirements as far as possible while completing service execution under the constraints of limited IaaS or PaaS resources, which needs to be implemented through an efficient online service scheduling algorithm.

4 Algorithm design and implementaion

4.1 Prioritized 3-deep Q-network

In the process of using DQN (Deep Q-Network), there will be a problem of overestimate (Liang et al., 2020). Therefore, in recent years, many scholars have proposed improved algorithms for DQN, including DDQN, Dueling DQN, distributed DQN, PER, etc. This section combines DDQN, Dueling DQN, and Prioritized Experience Replay three algorithms to improve DQN at the same time to construct Prioritized Dueling-DDQN (hereinafter referred to as Prioritized 3-DQN) algorithm. This algorithm avoids overestimation of DQN to a certain extent. At the same time, when updating the parameters of neural network, PER algorithm is used to replace the random sampling method in DQN and select the most effective learning samples from the sample memory to achieve the purpose of efficient learning.

The Prioritized 3-DQN algorithm also uses two neural networks with the same structure: the Eval network and the Target network. The Eval network is used to calculate the estimated Q value and can be updated in real time. The Target network is used to calculate the target Q value, and it is a temporarily frozen network. This article has made three improvements to DQN: two decoupling actions and one sampling method improvement. The specific descriptions are as follows:

(1) The output layer of the neural network is decoupled into two output streams, which output the current state value V and the action advantage function A, respectively, and then combine the state value V and the advantage function A to form the Q value. The advantage function refers to the degree of merit of the value that can be obtained by taking an action relative to the average value of the state for a particular state. In order to calculate the advantage function value corresponding to each action more conveniently, the average value of the advantage function value of all actions is set to 0. If the advantage function value corresponding to a certain action is greater than the average value in the state, then the advantage function

value corresponding to the action is positive, and vice versa. At this time, the calculation method of the Q value is shown in Eq. 15, where θ represents the neural network parameter, α and β represent the output flow neural network parameters corresponding to the state value and the action advantage function, and n is the action dimension.

$$Q(s,a;\theta) = V(s;\alpha) + A(s,\ a;\beta) - \frac{\sum_{a'}^{n} A(s,\ a';\beta)}{n} \tag{15}$$

(2) Based on DQN, the overestimation problem is solved by decoupling the selection of target action and calculating the target Q value. When calculating the actual value of Q, the Eval network provides the action in the next environment state, and the Target network provides the Q value of this action.

The Q value. At this time, the update process of the neural network is shown in Eq. 16, where θ and θ^{-} represent the Eval network and the Target network, respectively.

$$\begin{aligned} Q(s_t,a_t;\theta) &\leftarrow Q(s_t,a_t;\theta) \\ &+ \propto \left[r_t + \gamma Q\left(s', a^{max}\left(s';\theta\right);\theta^{-}\right) - Q(s_t,a_t;\theta)\right] \end{aligned} \tag{16}$$

(3) In the offline training phase of traditional DQN, the training samples are randomly selected from the experience replay pool without considering the priority relationship of the samples. However, different samples have different values, and the samples directly affect the training effect of the neural network. In order to improve the training effect of the neural network, it is necessary to determine a priority for each sample and conduct sampling according to the priority of the sample. As mentioned above, the Target network does not have the function of real-time updates. Therefore, as the Eval network is continuously updated, there will be a certain gap between the two networks while calculating the Q value. This gap is named the timing difference TD_Error. TD_Error can be represented by Eq. 17. The larger the TD_Error, the larger the gap between the current Q function and the target Q function, the more the neural network needs to be updated at this time, so TD_Error can be used to measure the value of the sample. In order to prevent the network from overfitting, samples can be drawn by probability. At this time, the probability of samples being drawn is shown in Eq. 18, where ϵ is a small value close to 0, which guarantees Samples with TD_Error of 0 may also have a chance to be drawn.

$$TD\_Error = r_t + \gamma Q\left(s',\ a^{max}\left(s';\theta\right);\ \theta^{-}\right) - Q(s_t,\ a_t;\ \theta) \tag{17}$$

$$P(i) = \frac{p_i}{\sum p_i} \tag{18}$$

where, $p_i = |TD_Error + \epsilon|$. The process of our Prioritized 3-DQN algorithm is as follows:

Initialize θ - network parameters; θ^{-} - copy of θ;Initiative N - the capacity of replay memory D;N_b - training batch size; C – target network replacement freq
Set $Q(s,a;\pi) = V(s;\pi,\beta) + (A(s,a;\pi,\alpha) - mean\sum_{a'} A(s,a';\pi,\alpha))$, while $\pi\epsilon(\theta,\theta^{-})$, and α, β are the parameters of the two streams of fully-connected layers
for $episode = 1$ to M **do**
  Initialize frame sequence $x \leftarrow ()$
  for $t = 1, T$ **do**
    Set state $s_t \leftarrow x$
    With probability ε select a random action a_t
    Otherwise select $a_t = argmax_a Q(s,\ a;\theta)$
    Execute action a_t in emulator and observe reward r_t and $s_{t+1} \leftarrow x$
    Store transition $(s_t,\ a_t,\ r_t,\ s_{t+1})$ in D
    Simpling N_b samples $(s_j,\ a_j,\ r_j,\ s_{j+1}),\ j\epsilon(1,\ N_b)$ from D with probability $P(j) = p_j/\sum_i (p_i)$
    Construct target values, one for each of $N_b$tuples :
    Define $a^{max}(s';\theta) = argmax_{a'} Q(s',a';\theta)$
    Set $y_j = \begin{cases} r_j & ,if\ s'isterminal \\ r_j + \gamma Q(s',\ a^{max}(s';\theta);\theta^{-}) & ,otherwise \end{cases}$
    Perform a gradient descent step on $(y_j - Q(s_j,a_j;\theta))^2$ with repect to the network parameters θ
    Update the sampling weight of each sample according to the formula $(y_j - Q(s_j,\ a_j;\theta))$
    Replace target parameters $\theta^{-} \leftarrow \theta$ every C steps
  end for
end for

Algorithm 1. Prioritized 3-DQN.

4.2 State space

When the service $svc_i$is ready, the method selects an instance of svc_i each time st_i^k and schedules it to a certain container. The environment status at this time is mainly determined by the physical relevant factors of the service instance st_i^k, such as resource requirements, running status of the container cluster are determined. Therefore, the state space can be denoted by Eq. 19:

$$S_i^k = \left[st_i^k, c_1, obs_{c_1}, pre_{c_1}^{svc_i}, \ldots, c_q, obs_{c_q}, pre_{c_q}^{svc_i}\right] \tag{19}$$

where

$obs_{c_y} = [que\_len_y, cpu\_len_y, mem\_len_y, disk\_len_y],\ (y \in q)$

Each value in the state space affects the scheduling decision of DRL, where st_i^k represents the current service instance to be scheduled, which is represented by the aforementioned Eq. 2, and cy represents the resource state of the yth container in the cluster, as shown in Eq. 4. It should be noted that there is a one-to-many relationship between service instances and containers. Each service instance can only be completed by one container, but each container can be assigned multiple service instances. When the remaining physical resources of the container are insufficient and the resource requirements of the service instance are required, the newly scheduled service instance needs to be added to the services queue to be executed in the container. obs_{c_y} is the running status of container c_y, where que_len_y represents the length of the service instance queue to be executed in container c_y, and cpu_len_y, mem_len_y, and $disk\_len_y$ respectively represent the sum of the CPU, memory, and disk storage space requirements of the waiting queue. The characteristic value $pre_{c_1}^{svc_i}$ represents the proportion of the result data length of the predecessor service of the current service instance in the container c_y after execution. For example svc_3 has two predecessor services svc_1 and svc_2. Assume that the length of the result data after the execution of these two precursor services is 4 and 6, so only the service instance of svc_1 is scheduled to the container c_1. The service instance of svc_3 is st_3^1. When being scheduled, $pre_{c_1} = 4/(4+6) = 0.4$.

TABLE 2 Table of data relation comparison.

Fields of batch_task table	Attributes of class service	Description
task_name	service_name	Service name
inst_num	inst_num	The number of instances
job_name	cs_name	The name of composite service
Duration	Duration	Expected execution time
plan_cpu	cpu	CPU cores requirements
plan_mem	mem	Memory requirements
Disk	Disk	Disk storage requirements
Length	Length	The length of result

TABLE 3 Table of dataset settings.

Dataset name	The number of composite services	The number of services	The number of service instances
Training data set	1,036	5,832	38,586
Test data set1	345	1,500	12,320
Test data set2	426	2,200	18,020
Test data set3	512	2,780	25,200

4.3 Action space

During scheduling decision-making, a suitable container is selected for the service instance as the action in DRL, and the action space is all the containers that can be selected. Suppose that the data center contains p hosts $\{h_1, \ldots, h_p\}$ at a certain time, host h_x can assign at most $container\_num_x$ containers with different configurations. When service instance st_i^k is ready to be scheduled, the agent in DRL can schedule it to any container in the cluster for execution, including all containers in reserved and on-demand modes. The action space at this time is shown in Eq. 20.

$$a_{num} = h_x \times container\_num_x \quad (x \in p) \tag{20}$$

4.4 Reward function

In order to enable the agent in DRL to learn effectively and obtain an effective scheduling strategy that optimizes the goal, a reasonable reward function needs to be designed to guide the learning process of the agent. In our model, in order to minimize the completion time and improve the user QoS and resource utilization of the cloud platform, this paper uses the difference between the expected execution time of the service instance and the waiting time. It then uses the ratio of the expected execution time as the reward for each scheduling. The value is as follows:

TABLE 4 Resource node settings.

Hosts	Containers	Detailed description (CPU cores; Memory capacity; Disk capacity; Bandwidth; Status)
Host 0	Container 0	4; 1.56; 10; 5; Running
	Container 1	4; 1.56; 10; 5; Stopped
	Container 2	8; 3.13; 18; 8; Running
Host 1	Container 3	4; 1.56; 10; 5; Stopped
	Container 4	8; 3.13; 18; 3; Running
	Container 5	8; 3.13; 18; 3; Stopped
Host 2	Container 6	4; 2.34; 12; 5; Stopped
	Container 7	8; 3.13; 18; 3; Running
Host 3	Container 8	4; 2.34; 12; 3; Running
	Container 9	8; 3.13; 18; 5; Stopped

$$r_i^k = \frac{\left(durtation_i - TW_i^k\right)}{duration_i} = 1 - \frac{TW_i^k}{duration_i} \tag{21}$$

Based on Eq. 21, the interval of reward value can be deduced as$[-\infty, 1]$. When the overall waiting time of the service instance

TABLE 5 Algorithm parameter setting.

Parameter name	Value
The number of hidden layers	3
Activation function	ReLU
Greed index ε	0.9
Experience replay pool size N	3,000
Number of sample sets N_b	200
Learning rate α	0.001
Discount factor γ	0.9
ϵ	0.001
Target network update frequency C	30

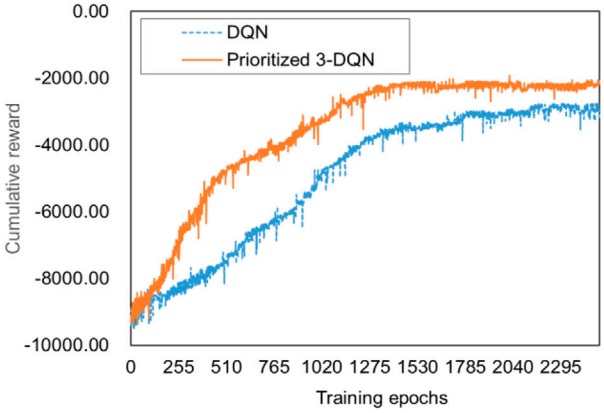


FIGURE 1
Training effect comparison chart.

TW_i^k is 0, the scheduling reward reaches the highest value of 1; when the overall waiting time TW_i^k is equal to the expected execution time, the reward value is 0; when the overall waiting time TW_i^k is greater than the expected execution time, the reward value begins to show a negative value. The longer the waiting time for execution, the smaller the reward value, and the greater the punishment. Through the reasonable design of the reward function, DRL can learn an effective service scheduling policy.

5 Experimental results

5.1 Simulation experiment setup

This paper uses Alibaba Cluster Data V2018 (Alibaba, 2018) as the data set for the simulation experiment. The data set contains six files in CSV format, describing the status information of the physical machine cluster, container cluster, and batch processing tasks. The original data set has a huge amount of data. There is inevitably a problem of missing data, and the data set is scattered and difficult to operate. Therefore, it is necessary to preprocess the original data set to obtain more targeted and valuable data. During the experiment, the preprocessed batch job data needs to be parsed and mapped into a composite service entity. The comparison between the fields of the preprocessed batch_task table and the attributes of the service class is shown in Table 2.

This paper divides the experimental data set into two parts: the training data set and the test data set, as shown in Table 3. In this experiment, 5,832 pieces of data are selected as services from the batch_task table, forming a total of 1,036 composite services, including 38,586 service instances. At the same time, to fully verify the effectiveness of Prioritized 3-DQN as a scheduling algorithm, this paper sets up three test sets with different data volumes.

In the initial stage of the simulation experiment, four hosts with different configurations are set, and each host contains container instances with different configurations and states. The relevant configuration of each container is shown in Table 4.

In implementing the Prioritized 3-DQN algorithm, the parameter settings are shown in Table 5. Both the Eval network and the Target network contain three fully connected neural network hidden layers, the last layer of which is divided into two output channels: state value and action advantage function. The greedy coefficient ε is 0.9. Each time the neural network parameters are updated, it will increase by 0.0001. That is, when selecting the container for the service instance, the container with the largest Q value will be selected with a probability of 0.9, and the container will be randomly explored with a probability of 0.1. After 1,000 updates, the value of ε becomes 1, and random exploration is no longer performed when selecting a container, but only the container corresponding to the largest Q value is selected. ϵ is set to 0.001, which ensures that samples whose timing difference TD_Error is 0 will also have a chance to be sampled. The target network update frequency C is set to 30, which means that every 30 times the Eval network is updated, its network parameters are copied to the Target network.

5.2 Prioritized 3-deep Q-network training effect

The essence of deep reinforcement learning algorithm learning is to maximize the cumulative reward of the round as the optimization goal, so the training effect can be reflected by the trend of the cumulative reward as the value changes with the number of training rounds. In addition, the Prioritized 3-DQN scheduling algorithm proposed in this

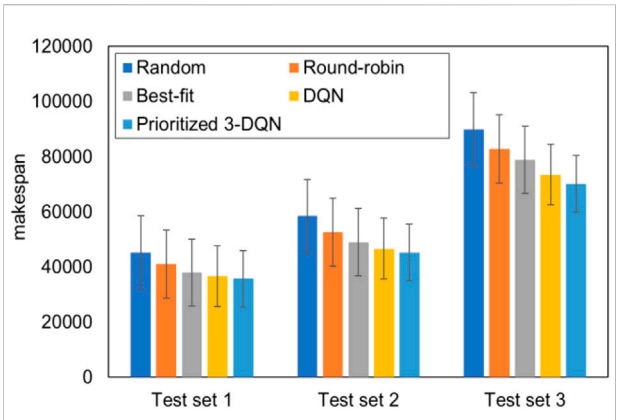


FIGURE 2
Comparison chart of average completion time of each test set with standard error.

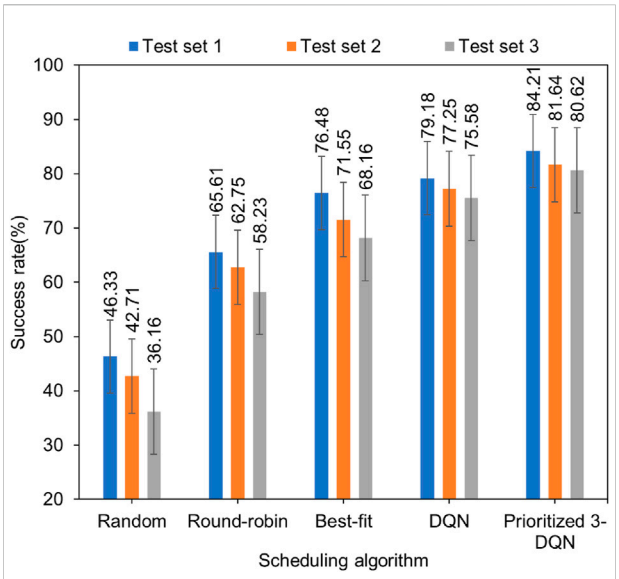


FIGURE 3
Comparison chart of composite services success rate with standard error.

paper is improved based on the DQN algorithm. In order to evaluate the convergence and stability of the improved Prioritized 3-DQN scheduling algorithm, it is compared with the original DQN algorithm. After 2,500 rounds of training using the training data set, they finally reached their optimal training effects. Figure 1 is a comparison chart of training effects.

It can be seen from Figure 1 that as the number of training rounds increases, the cumulative reward values calculated by the two algorithms show a gradual upward trend. After a certain number of rounds, they have reached a stable trend, indicating Prioritized 3-DQN and DQN are reasonable as the scheduling algorithm of the composite service model proposed in this paper. However, from the perspective of convergence, our algorithm can obtain a higher cumulative reward value under the same number of training rounds. In addition, When the number of training epochs reaches around 1,600, the Prioritized 3-DQN scheduling algorithm starts to converge. The DQN starts to converge when the number of training rounds reaches about 2,200. Thus, the convergence speed of our algorithm is faster, and a higher cumulative reward value is obtained after the iteration is completed. This is because each time the weight parameters of the neural network are updated in our algorithm, the experience samples with larger time-series differences are selected first, so as to ensure the learning effect of the neural network. From the perspective of stability, Prioritized 3-DQN decouples the selection of the target Q value action and the target Q value calculation, thereby avoiding the problem of overestimation. Therefore, compared with the DQN rising trend, the upward trend of our results is slightly smoother and more stable. In general, our Prioritized 3-DQN is very suitable for composite service scheduling strategies. Compared with DQN, it has higher learning efficiency and can converge earlier to achieve better results.

5.3 Makespan comparison

To verify the generalization ability of Prioritized 3-DQN as a composite service scheduling algorithm, DQN and the four common scheduling algorithms mentioned above are respectively applied to the composite service model. In the process of the comparative experiment, three test sets were used for 20 experiments, the completion time of the composite service was calculated, and the average results were obtained. Figure 2 summarizes the average completion time obtained after 20 experiments on each of the three test data sets.

It can be seen from Figure 2 that the completion time of Prioritized 3-DQN on different test data sets is shorter than the results of the other four scheduling algorithms. Among them, the difference in completion time between DQN and Prioritized 3-DQN is smaller than the other three scheduling algorithms. The completion time of Prioritized 3-DQN on three data sets is about 3.32% less than that of DQN on average. The number of service instances in the three test sets increases sequentially. With the increase in the number of service instances, the increase in the completion time of the composite service under different scheduling algorithms is different, and the gap in completion time between Prioritized 3-DQN and the other four scheduling algorithms is more prominent. This means that the algorithm and DQN algorithm proposed in this paper are more adaptable than other algorithms in terms of completion time.

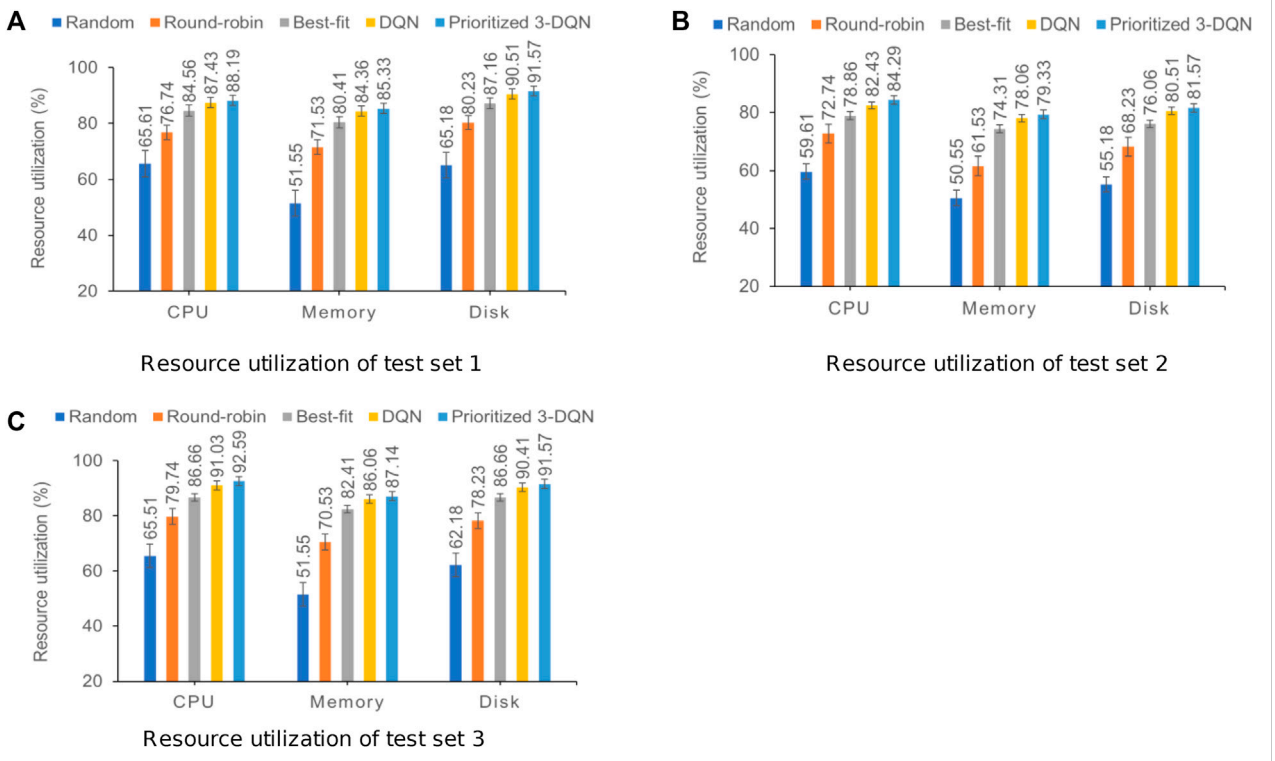


FIGURE 4
Resource utilization results of the three composite service test sets with standard error. **(A)** Resource utilization of test set 1. **(B)** Resource utilization of test set 2. **(C)** Resource utilization of test set 3.

5.4 Quality of service comparison

The degree of user satisfaction is also the main optimization goal of this article. The degree of user satisfaction is closely related to many factors, such as the number of requests for composite services reached per unit time, the number of service instances contained in each composite service, and the processing capacity of the container cluster set in the experiment. In this experiment, five scheduling algorithms are used in the same experimental environment to simulate simulation experiments on three composite service test sets, and then the success rate of each composite service test set is recorded, as shown in Figure 3.

By observing the above graph from a horizontal perspective, our algorithm can achieve the highest success rate compared to other scheduling algorithms. Vertically, with the increase in the number of composite services and service instances, the success rate of each scheduling algorithm after the completion of the composite service allocation is continuously reduced, but the reduction is different. Our algorithm is compared with the other four algorithms. It can be maintained in a relatively stable state, which ensures that the success rate of composite services is about 80% under different composite service test sets. The composite service success rate of Prioritized 3-DQN on the three data sets is about 4.82% higher than that of DQN. From the perspective of diversified loads, the Prioritized 3-DQN is more capable of making reasonable service scheduling decisions than other scheduling algorithms, thereby it increases the success rate of composite services and improves user QoS.

5.5 Resource utilization comparison

In addition to completion time and user QoS, the resource utilization of a container cluster can also be used as one of the criteria for evaluating the performance of scheduling algorithms. This section compares and analyzes resource utilization from the three dimensions: CPU, memory, and disk. During the simulation experiment, the resource utilization rate of the container cluster was recorded after each service instance was scheduled, and the average result of each resource utilization rate was calculated after one round of scheduling was completed. Figure 4 shows the resource utilization results of the three composite service test sets.

The above three graphs show that our prioritized 3-DQN, DQN, and Best-fit algorithms are significantly higher than the

other two algorithms in terms of resource utilization in the three dimensions, indicating that they can make full use of limited resources when scheduling service instances to complete the execution of composite services. When the Best-fit algorithm schedules service instances, it does not consider the data transmission relationship between services and the scheduling of subsequent service instances. It only schedules the current service instance to the container with the best performance and the shortest execution time. Therefore, the resource utilization in the three dimensions is lower than Prioritized 3-DQN and DQN. On the three data sets, the resource utilization of Prioritized 3-DQN on CPU, memory, and disk is about 1.39%, 1.11%, and 1.09% higher than that of DQN, respectively. The Prioritized 3-DQN is also higher than DQN in terms of resource utilization, indicating that Prioritized 3-DQN can make more reasonable scheduling decisions compared to DQN and has a more stable optimization capability under the same environment.

6 Conclusion

Cloud computing has brought great flexibility and cost-effectiveness to end-users and cloud application providers, and it has become a very attractive computing mode for various fields. With the continuous development of biological technology, massive biological data are continuously generated, and the requirements for data processing operation speed, computing power, and stability in practical applications also increase rapidly. Cloud computing has the characteristics of high-speed computing power, high storage capacity, and convenient use, which can meet the needs of biological research. At the same time, cloud providers provide security services to ensure the privacy and integrity of data. When biological samples are processed, each step needs to be supported and completed by cloud services. Between stages, biopharmaceutical companies realize data isolation by transferring data between services. Data quality plays a crucial role in the application effect of data, and the problem of data timeliness is one of the main factors affecting data quality. The timeliness of data can be improved synergistically by combining timeliness rules with statistical technical conditions or functional dependencies. How to use service scheduling strategy to improve service quality and resource utilization has become a key issue in cloud computing. This paper focuses on the core problem of service scheduling management in the container cloud platform. We proposed the composite service model under the modes of container instance (mixed reservation and on-demand), and we proposed the improved DQN algorithm as the scheduling algorithm of the composite service model in this paper. The simulation results show that, under the model presented in this paper, our 3-DQN algorithm is superior to the original DQN algorithm in terms of reliability and convergence. In addition, the algorithm can effectively reduce the completion time of the composite service and improve the user QoS and resource utilization in the container cloud environment.

The method proposed in this paper still has many defects for the actual cloud environment. From the results represented in the paper, the differences in completion time, composite service success rate, and resource utilization between DQN and Prioritized 3-DQN are small. The reason for the smaller difference may be that the scale of our experiments is relatively small. If the scale of the experiments is large, the advantages of Prioritized-3DQN may be more prominent. We also consider comparing Prioritized 3-DQN with the three algorithms used in this paper in the future. In addition, in the process of designing the composite service model in the container cloud environment, the energy consumption and resource cost of the cloud platform are not considered. We can do further research in future work.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: https://github.com/alibaba/clusterdata/blob/v2018/cluster-trace-v2018.

Author contributions

LY, PY, and YD contributed to conception and design of the study. LY wrote the manuscript and performed the statistical analysis. PY and YD helped supervise the project. HQ completed the formatting and editing of the manuscript. All authors contributed to manuscript revision, read, and approved the submitted version.

Funding

NSFC (Nos 61962040, 72062015, and 61662021), Hainan Education Department Project No. Hnky 2019-13, and Hainan University Educational Reform Research Project (Nos HDJY2102 and HDJWJG03).

Acknowledgments

This work was supported by grants from NSFC, Hainan Education Department Project and Hainan University Educational Reform Research Project.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Almansour, N., and Allah, N. M. (2019). "A survey of scheduling algorithms in cloud computing," in 2019 International Conference on Computer and Information Sciences (ICCIS), Sakaka, Saudi Arabia, 03-04 April 2019, 1–6.

Almezeini, N., and Hafez, A. (2017). Task scheduling in cloud computing using lion optimization algorithm. *Int. J. Adv. Comput. Sci. Appl.* 8 , 78–83. doi:10.14569/ijacsa.2017.081110

Barik, R. K., Lenka, R. K., Rao, K. R., and Ghose, D. (2016). "Performance analysis of virtual machines and containers in cloud computing," in 2016 international conference on computing, communication and automation (iccca), Greater Noida, India, 29-30 April 2016 (IEEE), 1204–1210.

Bernstein, D. (2014). Containers and cloud: From lxc to docker to kubernetes. *IEEE Cloud Comput.* 1, 81–84. doi:10.1109/mcc.2014.51

Chen, Z.-G., Zhan, Z.-H., Lin, Y., Gong, Y.-J., Gu, T.-L., Zhao, F., et al. (2019). Multiobjective cloud workflow scheduling: A multiple populations ant colony system approach. *IEEE Trans. Cybern.* 49, 2912–2926. doi:10.1109/TCYB.2018.2832640

Cheng, M., Li, J., and Nazarian, S. (2018). "Drl-cloud: Deep reinforcement learning-based resource provisioning and task scheduling for cloud service providers," in 2018 23rd Asia and South pacific design automation conference, Jeju, Korea (South), 22-25 January 2018, 129–134. ASP-DAC. IEEE.

Cui, Y., and Xiaoqing, Z. (2018). "Workflow tasks scheduling optimization based on genetic algorithm in clouds," in 2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA) (IEEE), Chengdu, China, 20-22 April 2018, 6–10.

[Dataset] Alibaba (2018). Cluster-trace-v2018. [EB/OL]. Available at: https://github.com/alibaba/clusterdata/blob/v2018/cluster-trace-v2018 (Accessed December 13,2018).

Dong, T., Xue, F., Xiao, C., and Li, J. (2020). Task scheduling based on deep reinforcement learning in a cloud manufacturing environment. *Concurr. Comput. Pract. Exper.* 32, e5654. doi:10.1002/cpe.5654

Dong, T., Xue, F., Xiao, C., and Zhang, J. (2021). Workflow scheduling based on deep reinforcement learning in the cloud environment. *J. Ambient. Intell. Humaniz. Comput.* 12, 10823–10835. doi:10.1007/s12652-020-02884-1

George, N., Chandrasekaran, K., and Binu, A. (2016). "Optimization-aware scheduling in cloud computing," in Proceedings of the International Conference on Informatics and Analytics, August 25 - 26, 2016, Pondicherry India, 1–5.

Ghasemi, S., Kheyrolahi, A., and Shaltooki, A. A. (2019). Workflow scheduling in cloud environment using firefly optimization algorithm. *JOIV Int. J. Inf. Vis.* 3, 237–242. doi:10.30630/joiv.3.3.266

Islam, M. T., Karunasekera, S., and Buyya, R. (2021). Performance and cost-efficient spark job scheduling based on deep reinforcement learning in cloud computing environments. *IEEE Trans. Parallel Distrib. Syst.* 33, 1695–1710. doi:10.1109/tpds.2021.3124670

Joy, A. M. (2015). "Performance comparison between linux containers and virtual machines," in 2015 International Conference on Advances in Computer Engineering and Applications (IEEE), Ghaziabad, India, 19-20 March 2015, 342–346.

Kyaw, L. Y., and Phyu, S. (2020). "Scheduling methods in hpc system," in 2020 IEEE Conference on Computer Applications (ICCA) (IEEE), Yangon, Myanmar, 27-28 February 2020, 1–6.

Li, F., and Hu, B. (2019). "Deepjs: Job scheduling based on deep reinforcement learning in cloud data center," in Proceedings of the 2019 4th international conference on big data and computing, Guangzhou China, May 10 - 12, 2019, 48–53.

Liang, S., Yang, Z., Jin, F., and Chen, Y. (2020). "Data centers job scheduling with deep reinforcement learning," in *Advances in knowledge discovery and data mining* (New York: Springer International Publishing), 906–917.

Meng, H., Chao, D., Huo, R., Guo, Q., Li, X., and Huang, T. (2019). "Deep reinforcement learning based delay-sensitive task scheduling and resource management algorithm for multi-user mobile-edge computing systems," in Proceedings of the 2019 4th International Conference on Mathematics and Artificial Intelligence, Chegndu China, April 12 - 15, 2019, 66–70.

Myers, T. A., Chanock, S. J., and Machiela, M. J. (2020). Ldlinkr: An r package for rapidly calculating linkage disequilibrium statistics in diverse populations. *Front. Genet.* 11, 157. doi:10.3389/fgene.2020.00157

Orhean, A. I., Pop, F., and Raicu, I. (2018). New scheduling approach using reinforcement learning for heterogeneous distributed systems. *J. Parallel Distributed Comput.* 117, 292–302. doi:10.1016/j.jpdc.2017.05.001

Panwar, N., Negi, S., Rauthan, M. M. S., and Vaisla, K. S. (2019). Topsis–pso inspired non-preemptive tasks scheduling algorithm in cloud environment. *Clust. Comput.* 22, 1379–1396. doi:10.1007/s10586-019-02915-3

Ran, L., Shi, X., and Shang, M. (2019). "Slas-aware online task scheduling based on deep reinforcement learning method in cloud environment," in 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS) (IEEE), Zhangjiajie, China, 10-12 August 2019, 1518–1525.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. (2016). "Prioritized experience replay," in International Conference on Learning Representations, San Juan, Puerto Rico, May 2-4, 2016.

Van Hasselt, H., Guez, A., and Silver, D. (2016). "Deep reinforcement learning with double q-learning,", Phoenix, Arizona USA, February 12–17, 2016, 2094–2100. doi:10.1609/aaai.v30i1.10295*Proc. AAAI Conf. Artif. Intell.*30

Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., and Freitas, N. (2016). "Dueling network architectures for deep reinforcement learning," in International conference on machine learning (PMLR), New York NY USA, June 19 - 24, 2016, 1995–2003.

Wei, Y., Pan, L., Liu, S., Wu, L., and Meng, X. (2018). Drl-scheduling: An intelligent qos-aware job scheduling framework for applications in clouds. *IEEE Access* 6, 55112–55125. doi:10.1109/access.2018.2872674

Xiaoqing, Z., Yajie, H., and Chunlin, A. (2018). "Data-dependent tasks re-scheduling energy efficient algorithm," in 2018 IEEE 4th International Conference on Computer and Communications (ICCC), Chengdu, China, 07-10 December 2018, 2542–2546. IEEE.

Yang, S., Zhu, F., Ling, X., Liu, Q., and Zhao, P. (2021). Intelligent health care: Applications of deep learning in computational medicine. *Front. Genet.* 12, 607471. doi:10.3389/fgene.2021.607471

Zhang, L., Qi, Q., Wang, J., Sun, H., and Liao, J. (2019). "Multi-task deep reinforcement learning for scalable parallel task scheduling," in 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 09-12 December 2019, 2992–3001. IEEE.

frontiers | Frontiers in Genetics

TYPE Original Research
PUBLISHED 22 November 2022
DOI 10.3389/fgene.2022.966483

OPEN ACCESS

EDITED BY
Alfredo Pulvirenti,
University of Catania, Italy

REVIEWED BY
Lianyong Qi,
Qufu Normal University, China
Yanwei Xu,
Tianjin University, China

*CORRESPONDENCE
Lei Yu,
yuleiimu@sohu.com
Yucong Duan,
duanyucong@hotmail.com

SPECIALTY SECTION
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

RECEIVED 11 June 2022
ACCEPTED 02 November 2022
PUBLISHED 22 November 2022

CITATION
Yu L and Duan Y (2022), Responsive and intelligent service recommendation method based on deep learning in cloud service.
Front. Genet. 13:966483.
doi: 10.3389/fgene.2022.966483

Responsive and intelligent service recommendation method based on deep learning in cloud service

Lei Yu[1]* and Yucong Duan[2]*

[1]Department of Computer Science, Inner Mongolia University, Hohhot, China, [2]Department of Data Science and Big Data Technology, Hainan University, Haikou, China

The rapid expansion of the cloud service market is inseparable from its widely acclaimed service model. The rapid increase in the number of cloud services has resulted in the phenomenon of service overload. Service recommendations based on services' function attributes are important because they can help users filter services with specific functions, such as the function of guessing hobbies on shopping websites and daily recommendation functions in the listening app. Nowadays, cloud service market has a large number of services, which have similar functions, but the quality of service (QoS) is very different. Although the recommendation based on services' function attributes satisfies users' basic demands, it ignores the impact of the QoS on the user experience. To further improve users' satisfaction with service recommendations, researchers try to recommend services based on services' non-functional attributes. There is sparsity of the QoS matrix in the real world, which brings obstacles to service recommendation; hence, the prediction of the QoS becomes a solution to overcome this obstacle. Scholars have tried to use collaborative filtering (CF) methods and matrix factorization (MF) methods to predict the QoS, but these methods face two challenges. The first challenge is the sparsity of data; the sparsity makes it difficult for CF to accurately determine whether users are similar, and the gap between the hidden matrices obtained by MF decomposition is large; the second challenge is the cold start of recommendation when new users (or services) participate in the recommendation; its historical record is vacant, making accurately predicting the QoS value be more difficult. To solve the aforementioned problems, this study mainly does the following work: 1) we organized the QoS matrix into a service call record, which contains user characteristic information and current QoS. 2) We proposed a QoS prediction method based on GRU–GAN. 3) We used the time series data for quality predictions and compared some QoS prediction methods, such as CF and MF. The results showed that the prediction results based on GRU–GAN are far superior to other prediction methods under the same data density. We aim to help the engineering community promote their findings, shape the technological revolution, improve multidisciplinary collaborations, and collectively create a better future.

KEYWORDS

deep learning, QoS prediction, service recommendation, services, intelligent

1 Introduction

While massive web services bring convenience and innovation, the coverage of web service resources is getting wider and faster (Zhang Y. et al., 2020). The scale of the cloud service market in the future should not be underestimated. The explosive growth of the number of web services in the cloud service community has become a new challenge because it is difficult for users to select high-quality services from many services. In recent years, more and more people try to use deep learning (DL) models (Nath and Wu, 2020) to learn the functional properties of web services, such as service operations, inputs, outputs, and prerequisites, to realize user demand for functionality. Its advantage is that it can select services with specific functions in a targeted manner according to the specific functions of the services required. This technology is now relatively mature. However, existing research has found that when there are a large number of services with the same or similar functions, filtering only based on the functional attributes of the services is less effective. People began to try to introduce QoS to measure the non-functional performance of a service. The study found that although QoS-based service recommendations can recommend higher quality services for people, the sparseness of real-world data makes the recommendation results unreliable for users. Therefore, people try to use the quality prediction of web services (hereinafter referred to as quality prediction) to solve the problem caused by data sparse. Quality prediction is an essential link in service recommendation; however, most quality predictions now use collaborative filtering (CF) and matrix factorization (MF) methods. However, these two types of methods only use the interaction information between the user and the service as the main basis, ignoring the user's personalized feature information (Hamilton, 2022). Therefore, when faced with the cold start problem, the reliability of the predicted results is low due to the lack of the aforementioned interactive information. Figure 1 shows a service recommendation scenario where service users used a list of services and intend to use more suitable services by automatic recommendations.

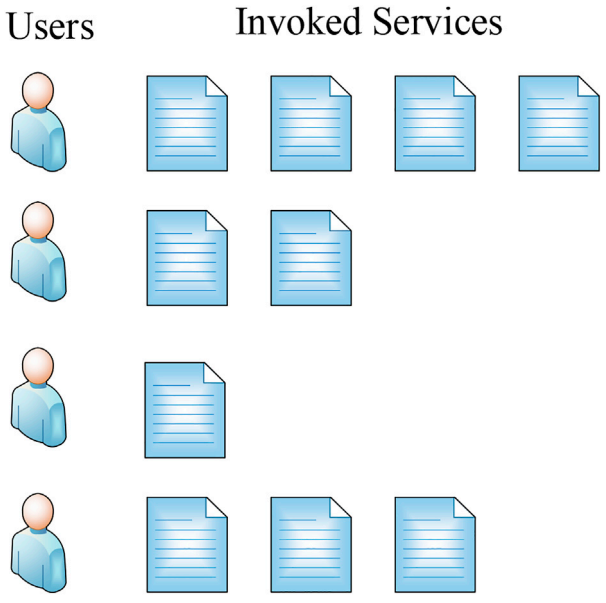


FIGURE 1
Recommendation scenario.

Based on the aforementioned problems, this study rearranged each call record to include the feature information of the user under the service and uses a deep neural network to analyze the nonlinear relationship between the feature information and response time in each record so that the QoS prediction value is reliable. The scale of the cloud service market is growing rapidly (Sandhu, 2021). Although it brings more choices to users, the dazzling array of services also makes users dazzled. Cloud service recommendation can effectively solve this problem, so the main work of this study is to use deep learning to research cloud service recommendations and explore more accurate cloud service recommendation methods to recommend services with higher QoS values to users. The purpose of this subject is to make its contribution in the field of service recommendation, which has certain academic value and research significance. With the promotion of personalized service concepts and the wide application of service-oriented computing (SOC), more and more enterprises provide users with personalized products. Service-oriented architecture (SOA) is the most recognized implementation, which makes service invocation more convenient. Therefore, there is explosive growth in the number of web services (WSs), and it is difficult for conventional service recommendation methods to efficiently process and utilize services (Zhang W. et al., 2020). This phenomenon is also called service overload. People try to use the service recommendation method to solve the problem of service overload. Service recommendation is mainly divided into two recommendation methods based on function and non-function. Among them, the technology of mining users' functional requirements is relatively mature. How to filter service recommendation methods with higher service quality became a hotspot in the field of service recommendation. To solve our problems, this study mainly did the following work:

1) We organized the QoS matrix into a service call record, which contains user characteristic information and current QoS. This study used the public dataset WS-Dream as

experimental data and found that there are a large number of irrelevant and redundant information in the dataset, so we used a variety of tools to extract feature information, specifically including filtering fields, filling missing values, morphological restoration, and converting lowercase, and then performing feature extraction to generate time series data.

2) We proposed a QoS prediction method based on GRU–GAN. Compared with other service recommendation methods, this method can overcome the drawbacks of linear operations brought by collaborative filtering-based service recommendation methods and matrix factorization-based service recommendation methods by learning the nonlinear relationship between eigenvalues in time series data. Through the generative adversarial network, the real-time series data and the predicted time series data are used for adversarial training to improve the prediction performance of the model.
3) We used the time series data for quality prediction and compared some QoS prediction methods, such as CF and MF. The results show that the prediction results based on GRU–GAN are far superior to other prediction methods under the same data density.

At present, there are roughly three solutions for QoS-based service recommendation: CF-based service recommendation (Lo et al., 2012a), MF-based service recommendation (Lo et al., 2012b), and DL-based service recommendation (Hudson et al., 2021). The CF-based method is an ancient algorithm in the field of service recommendation. The main idea of the algorithm is to use interaction records to find similar users (or services) and to recommend services for users based on the view that similar users have similar evaluations of services. CF-based methods are further subdivided into two service recommendation methods: memory-based and model-based methods. Among them, memory-based methods can be subdivided into three types, which are user-based (Shao et al., 2007), item-based (Deshpande and Karypis, 2004), and a combination of the two (Zheng et al., 2009). The memory-based CF method calculates the similarity between users (or services) according to the matrix of users calling services, then predicts the QoS of similar users (or services) based on the similarity, and then sorts the QoS in a certain order according to the prediction results, and finally selects the top K services with the best service quality and recommend them to users. Wu et al. (2012) proposed a neighborhood-based CF method, which eliminated different levels of the QoS by adjusting the similarity calculation method, and then used a similarity fusion method to reduce the impact of data sparsity. To improve prediction accuracy, scholars have begun to pay attention to the impact of contextual information such as time and space on the QoS. For example, Yu and Huang (2014) proposed a time-aware CF algorithm to predict the missing QoS; Liu et al. (2015) used a spatially aware CF algorithm to improve the performance of service recommendations. However, when the aforementioned methods were not able to provide real-time recommendations when faced with a large amount of data, Zhang et al. (2011) proposed the WSPred model to improve prediction accuracy by embedding temporal information. Although CF has achieved more intentional results in the field of service recommendation in the early stage (Wu et al., 2012; Liu et al., 2015), there are still the following drawbacks: 1) data sparsity: CF methods mainly rely on the call records between users and services to calculate similarity; these call records usually only provide some low-dimensional and linear features, so when the data density is small, insufficient learning of features limits the improvement of prediction performance. 2) Cold start: when a new user (or service) is the target user, the reliability of the similarity calculation result is low.

To solve the interference caused by the aforementioned problems, the MF method has been applied to service recommendations by many scholars. For example, there is a QoS matrix R_{m*n} generated by a user calling service, and the "user implicit matrix U_{m*k}" and "service implicit matrix S_{k*n}" are obtained through MF. These two matrices are used to describe the characteristics of users and services, respectively. By optimizing the objective function to make the product of the "hidden matrix" closer to R, the missing data in R are also filled.

Tang N. et al. (2016) proposed a QoS prediction algorithm ClustTD based on location clustering and tensor decomposition. This method uses location information to cluster users and services, and then performs tensor decomposition on the user and service vectors. The results are weighted and combined to finally obtain the predicted value of the QoS; Xu et al. (2013) used the upper and lower information of the service and user location to perform matrix decomposition, and proposed LE-MF for the prediction of missing values, and user clustering and service clustering, reduce the volume of the QoS matrix, and finally complete the prediction task of vacancy values through matrix decomposition; Yin et al. (2016) considered the impact of the network environment on the QoS, and combined the autonomous system into the judging network location neighbor index. A QoS prediction method based on network location-aware neighbor selection web service recommendation was proposed, which improved the prediction performance by reducing the solution space; Qi et al. (2020) considered from the perspective of service security and concluded that although the spatiotemporal information of users and services improved, it improves the reliability of the recommendation but also reduces the security, so they add the location-sensitive hashing technology to the space-time information to enhance the privacy protection of users and services. Matrix factorization improves the reliability of quality prediction results by alleviating the problem of data sparsity, but the number of features involved in the calculation is limited, which makes it difficult to overcome the challenges brought by a cold start.

2 Related work

In SOA recommended systems (Nitu et al., 2021), the user's personalized needs are presented in the form of QoS (Tran and Tsuji, 2009). QoS is an important indicator for evaluating service performance, so it can be used as the most important factor to distinguish the quality of web services. QoS includes performance, reliability, security, and some other metrics. For users, the level of the QoS value determines the QoS experience; for services, the level of the QoS value indicates the quality of service performance and also affects the popularity of the service.

Deep learning (DL) is an important subcategory of machine learning, which trains and captures data features through neural networks (Li et al., 2020). DL learns the abstract expression of the data and the inherent laws between the data in the massive data, and extracts the feature representation of the complex level from it. Through the aforementioned process, the computer has the ability to analyze and learn like a human. The early neural network is similar to the combination of simple neural units to form an artificial neural network. Since the DL model is widely recognized, various sub-models are also derived, such as the convolutional neural network (CNN) model, deep belief network (DBN) model, belief network(BN) model, and stacked autoencoder (SAE) model (Zhang et al., 2021).

The core of the RNN analysis problem is to find the invisible connection between the input time series data. The RNN is used to process time series data, and the effective information contained in time series data at different times is different, so the RNN can be regarded as a kind of a neural network with short-term memory ability.

In the RNN, the current neuron can accept the information of not only its previous and backward neurons but also its own information, and finally form a network structure with loops. Because of the characteristics of receiving neurons, the RNN has a stronger memory ability. At present, RNNs have been widely used in tasks such as computer vision, meteorology, and text sentiment classification. However, due to the long training time of the original RNN, the training will cause the gradient to be in two extreme states, that is, explosion and disappearance.

Gated recurrent units (GRUs) are a variant of LSTM. The structure of LSTM is relatively complex, and the number of parameters it contains far exceeds that of GRU, which makes the training difficulty of parameters sharply increased. In response to the aforementioned situation, the GRU was proposed to reduce the number of parameters in LSTM and ensure the effect of training. The specific method is that the GRU combines the forgetting gate and the input gate to reduce the complexity of the neural network, which not only ensures the memory ability of the RNN but also reduces the complexity of the neural network. It improves the training efficiency of the network. The GRU contains two gates. The update gate determines the degree to which the information in the previous time series data is brought into the future, and the larger the value, the greater the degree of introduction; the reset gate determines the importance of the information in the current time series, and the larger the value is, the current time series data are less important (Qi et al., 2020a).

The GAN is a neural network that uses game thinking (Creswell et al., 2018). The GAN is mainly composed of generator G (Generator) and discriminator D (Discriminator). Generator G learns the distribution of the given data, and when the noise is input to the generator, it will generate "fake data" similar to the real sample; discriminator D mainly identifies "fake data" from a sample set that is a mixture of real and fake. G and D continuously update the loss function through adversarial training to achieve the overall optimization goal. Through multiple game processes, the generator can achieve the goal of "mixing the fake with the real." The optimal state is achieved when discriminator D cannot distinguish the authenticity of the data, that is, when the output probability of discriminator D is 1/2.

3 Model design

The user-service call records are generated through the real data set WS-Dream, and then a combination of the gated recurrent neural network and the adversarial neural network-based adversarial gated recurrent neural network (GRU–GAN) is proposed in this study to predict the motivation and value of the QoS-specific implementation process. This method can effectively predict the QoS value when the data sparsity is low and can also alleviate the impact of user cold start to a certain extent. Figure 2 shows our methodological framework.

3.1 Generator model

In the generative adversarial network, the design of the generator needs to be specifically designed according to specific experiments. In the training phase of the GRU–GAN model, the main purpose of G is to "cheat" D with the predicted QoS. The generator uses the time series within the 0-t time series to train the weight of each hidden neuron, and then whenever the noise data converted from the user feature information at the next moment are entered, it will predict the next moment when the user invokes the service. This section will introduce the specific design scheme and training process of the generator in the quality prediction model of this study. Figure 3 shows the generator model in the QoS prediction model proposed in this study.

The input of the generator is a time series and the output is the QoS prediction value at the future time. The QoS of the generator from input noise to output can be summarized as Eq. 1.

$$\widehat{y_{u,t+1}} = \mathrm{G}\left(t_{u,t}, x_{u,t+1}\right), \tag{1}$$

where $\widehat{y_{u,t+1}}$ represents the QoS of user u at time t+1, $t_{u,t}$ represents t real-time series generated by u users calling s

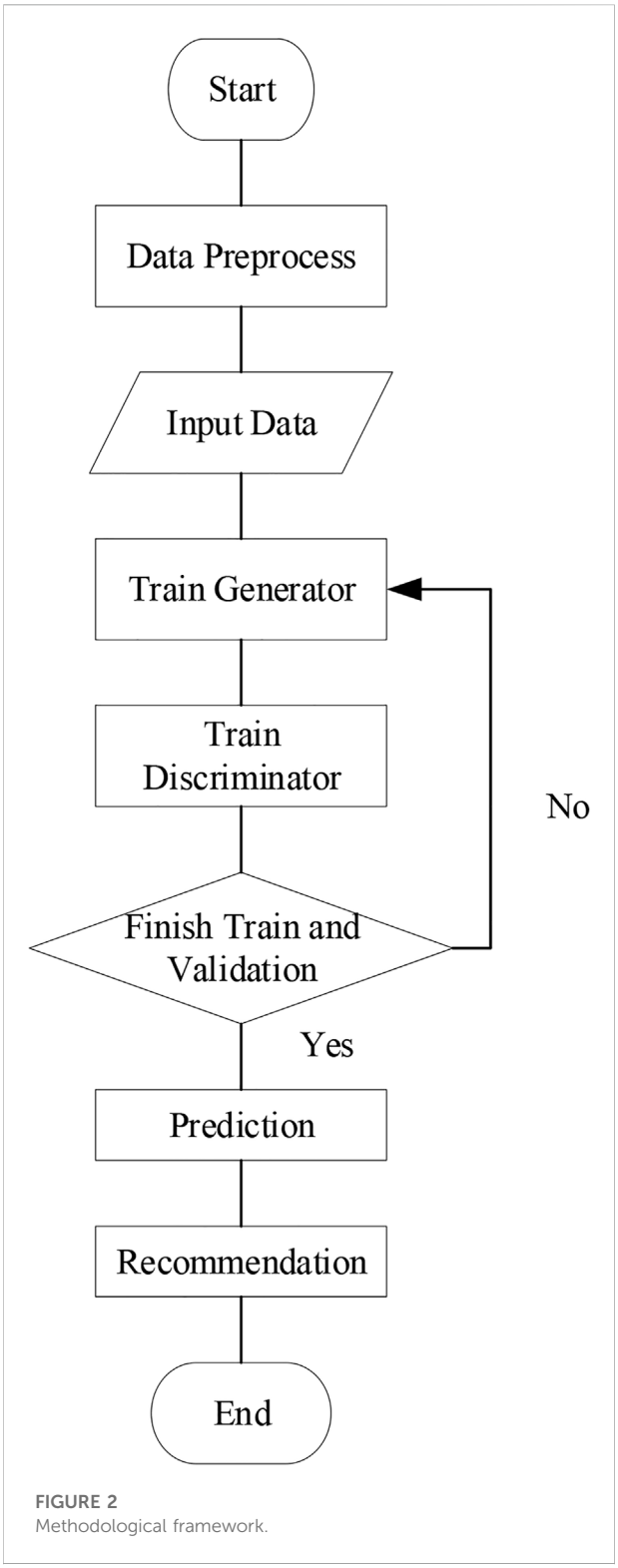


FIGURE 2
Methodological framework.

services, and $x_{u,t+1}$ represents the feature information of user u in the t+1st time series. G () is to train the generator function using the time series at time 0-t, which contains the weights for each hidden neuron.

Since the data type used in this study are time series data, RNN itself is a kind of neural network suitable for studying time series data. In related research, it was found that GRU can better deal with the gradient decay problem of RNN and can better capture the relationship between time series. Considering the applicability of GRU in this study, this article puts GRU into the generator model. At the same time, to ensure the consistency of the data dimensions of each connected part, the fully connected neural network is also put into the generator model, and the Leaky ReLU activation function is used in the fully connected layer, where a = 0.02. In GRU–GAN, the loss function of the generator is defined as the error between the predicted value of the QoS and the real value of the QoS. This study uses the L1 loss function to measure the error. The loss function calculation formula of generator G is as shown in Eq. 2.

$$Loss_G = \sum_{t=1}^{n} \left| y_t - \hat{y}_t \right|. \tag{2}$$

Among them, y_t represents the real QoS value at time t and $\hat{y}_t$ represents the predicted QoS value at time t. By minimizing the loss function of G, the error between the real data and the predicted data can be reduced, thereby improving the prediction performance of the generator.

In the generator network, the real data set is first input in chronological order; then the fully connected layer is used to map the dimension of the real data to the same dimension as the input layer of the GRU network, and the distribution characteristics of each feature in the real data and the QoS characteristics are learned. The fitting process, which can be expressed as a regression equation, constructs a function from a historical variable to the current value of a variable in a certain dimension. This fitting process can be expressed as Eq. 3.

$$\hat{y}_t = \theta_n x_n^t + e_t. \tag{3}$$

Among them, $\hat{y}_t$ represents the predicted value of the QoS at time t, θ_n is the 1*n-dimensional weight vector at time t, x_n^t is the feature vector of the n*1-dimensional user-service call record at the current time, and e_t is the current time error.

Therefore, whenever the basic information of the current state of the user is input, Eq. 3 will calculate the QoS value of the service invoked by the user at the current moment. The more the training samples are input, the better the fitting effect of the function will be. The QoS will get closer and closer to the real QoS as the number of iterations increases. The forward training process and backpropagation process of GRU in this study will be introduced separately in the following section.

The reset gate determines how much of the previously input information is written on the candidate set. First, the product of the weight matrix A_r and h_{t-1} and x_t spliced into a matrix is calculated, and then the gate to convert the calculation result of $A_r \cdot [h_{t-1}, x_t]$ is reset between 0 and1 through the activation function. The larger the value of r_t, the more information is

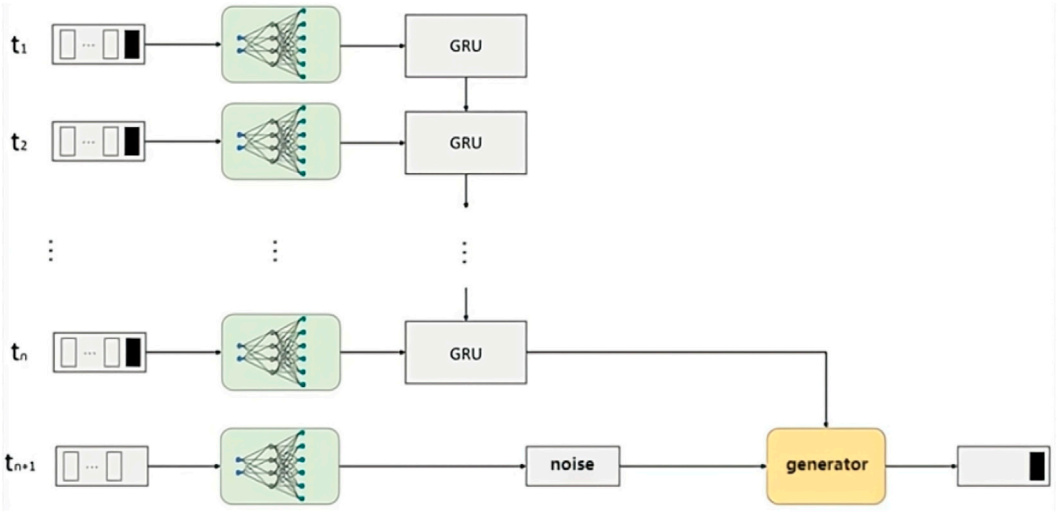


FIGURE 3
Generator model.

written in the previous state. The calculation method of the reset gate is shown in Eq. 4.

$$r_t = \sigma(A_r \cdot [h_{t-1}, x_t] + e_r), \tag{4}$$

where A_r represents the weight matrix of the reset gate, h_{t-1} represents the hidden state at time t-1, x_t represents the sequence input at time t (through the fully connected layer), e_r represents the bias of the reset gate, and σ is the sigmoid function.

The r_t value calculated by Eq. 4 will be used for the calculation of the candidate hidden state h_{t-1}; tanh converts the calculation result of $W_{\tilde{h}} \cdot [r_t * h_{t-1}, x_t]$ into a value between −1 and 1. From the calculation formula of $\tilde{h}_t$ Eq. 5, it can be seen that when r_t is smaller, the smaller is $\tilde{h}_t$, that is, more past information is needed.

$$\tilde{h}_t = \tanh(A_{\tilde{h}} \cdot [r_t * h_{t-1}, x_t]), \tag{5}$$

where $A_{\tilde{h}}$ represents the weight of the candidate hidden state and tanh represents the activation function.

The update gate z_t determines the extent to which the state information at time t-1 is brought into the current state. The calculation of the update gate is similar to that of the reset gate, and its calculation method is shown in Eq. 6.

$$z_t = \sigma(A_z \cdot [h_{t-1}, x_t] + c_z), \tag{6}$$

where A_z represents the weight of the update gate and c_z represents the bias of the update gate.

Based on the aforementioned calculation process, hidden state h_t at the next moment can be obtained. In Eq. 7, it can be seen that when the value of z_t is larger, memory data $z_t * \tilde{h}_t$ are more, and forgotten data $(1 - z_t) * h_{t-1}$ are less.

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t. \tag{7}$$

After completing the forward propagation process, we obtained relatively good neural network parameters. To optimize the parameters of the neural network, we need to optimize the weight parameters and bias parameters through backpropagation until the upper limit of the number of iterations is reached. At this time, $Loss_G$ is lower; that is, the performance of the generator network is better. The process of backpropagation will be described as follows.

h_t obtained by Eq. 1 is $\hat{y}_t$ in Eq. 1, and minimizing the loss function of the entire QoS is the goal of the entire training period. The loss function definition at this time can be expressed as Eq. 8.

$$\begin{cases} l(t) = \left| h_t - \hat{y}_t \right| \\ L = \sum_{1}^{T} (t) \end{cases}, \tag{8}$$

where $l(t)$ represents the loss function value calculated at time t and L represents the cumulative loss of the entire time series during training.

3.2 Discriminator model

This section mainly introduces the discriminator model, including the main tasks, model structure, and processing process of the discriminator model. The main task of discriminator D is to distinguish true from false from the real data set and the predicted data set generated by the generator, and give a probability value between 0 and 1 to the records in input D. The loss function of D is shown in Eq. 9.

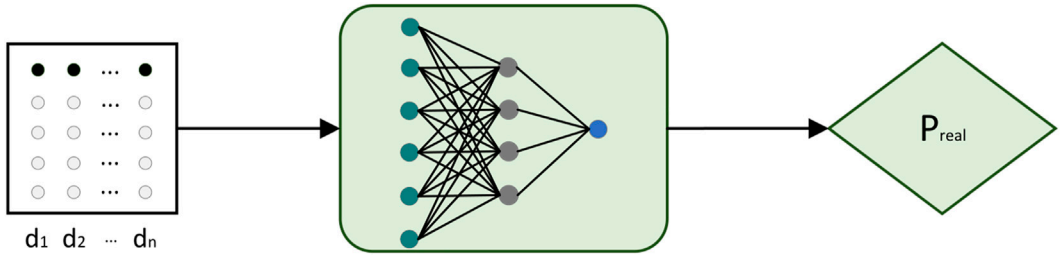


FIGURE 4
Discriminator model.

$$Loss_D = \sum_{t=1}^{n} D\left(\hat{y}_t\right) - D\left(y_t\right). \tag{9}$$

The discriminator mainly consists of three fully connected layers. The input time series is mapped to a probability value between 0 and 1 through a fully connected neural network. The larger the probability value, the more likely the current input time series is to be true. Otherwise, the input is considered to be more likely to be a predicted value. The structure of the discriminator network is shown in Figure 4.

3.3 GRU–GAN

In the first two sections, the generator model and the discriminator model are introduced, respectively. In this subsection, the whole of the proposed GRU–GAN-based QoS prediction model will be elaborated. In the initial GAN algorithm network, the generator will introduce random noise, and the random noise and the real data satisfy the same distribution and have similar probability densities. The discriminator integrates the input real samples and noise samples into a new sample collection and obtains the distribution probability between 0 and1 through the fully connected layer, that is, the probability that the sample includes the real data. The generator is responsible for feeding the data, and the discriminator is responsible for separating the data and using constant comparison to complete the balance to achieve learning. The input to the generator in a GAN is early data in the sequence, and the output is the predicted sequence data. Therefore, the input of the discriminator can be represented by two parts, which are the real-time series and the future time series obtained by the generator; the output of the discriminator is the probability distribution of these two kinds of data.

For the data set in this study, the data output by the generator need to meet the same distribution law as the real data, and the generated data are also the time series. The difference from the real data is the response time in the sequence.

In a general generative adversarial network, the generator converts a set of input noises into a fake sample set, and then through adversarial training, the predicted data generated by the generator have the same distribution as the real data. In this study, the output of the generator is the response time of a specific time series, so the output of the generator has a one-to-one correspondence with the input. Therefore, in the GRU–GAN model, the input random noise z is a specific set of time series data.

4 Experiment

4.1 Experiment data

This study conducts experiments on the WS-Dream dataset, which is widely used by academia to study QoS prediction problems. The dataset was originally collected by Zhang et al. (QoS values for 5,828 services from 339 distributed computers in PlanetLab's 30 countries).

4.2 Evaluation criteria

To evaluate the performance of the QoS prediction model based on GRU–GAN, we choose the two most widely used metrics for continuous variables: mean absolute error (MAE) and root mean squared error (RMSE).

4.2.1 MAE

MAE represents the mean of the absolute error between the predicted value and the observed value, and it represents the mean margin of error of the predicted value. It is also a commonly used regression loss function, and its calculation method is shown in Eq. 10.

$$MAE = \frac{\sum_{u,s}\left|r_{u,s} - \widehat{r_{u,s}}\right|}{N}. \tag{10}$$

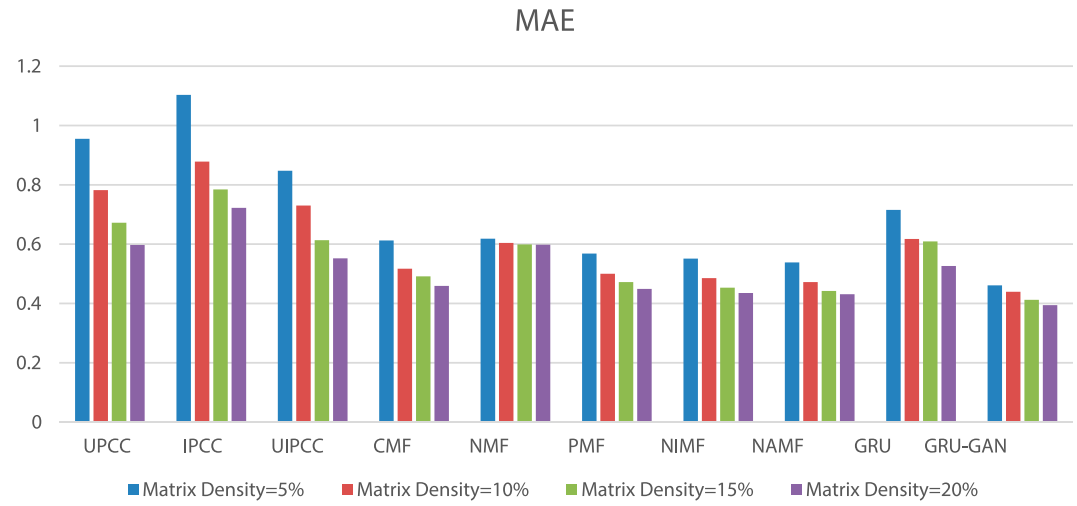


FIGURE 5
MAE comparison.

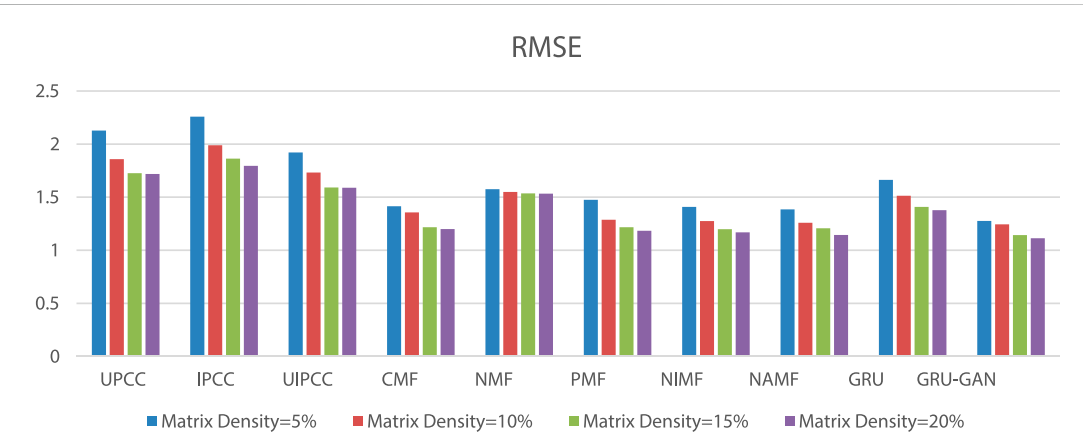


FIGURE 6
RMSE comparison.

4.2.2 RMSE

The formula for calculating the root mean square error is as follows:

$$RMSE = \sqrt{\frac{\sum_{u,s}\left(r_{u,s} - \widehat{r_{u,s}}\right)^2}{N}}. \quad (11)$$

4.3 Contrast methods

For the prediction effect of this deep learning-based service recommendation model, this study selects some representative QoS prediction methods for comparison. These comparison methods are described in detail next.

1. UPCC (Shao et al., 2007): this method is a memory-based collaborative filtering algorithm that uses the Pearson coefficient to find similar users and uses the QoS values of similar users to predict the QoS value of the target user.
2. IPCC (Deshpande and Karypis, 2004): similar to method 1, this method looks for similar services and uses the QoS values of similar services to predict the QoS value of the target service.
3. UIPCC (Zheng et al., 2009): this method combines the advantages of UPCC and IPCC to predict QoS values, and add parameters to balance the roles of the two.

4. CMF (Koren, 2010): this method uses the classical matrix factorization method to build a global model for quality prediction.
5. NMF (Lee and Seung, 1999): although this method is also based on matrix decomposition to solve the QoS value, this method adds a non-negative factor to matrix decomposition to improve the reliability of matrix decomposition.
6. PMF (Mnih and Salakhutdinov, 2008): this method introduces a probability model for probability matrix decomposition and optimizes the original matrix decomposition model.
7. NIMF (Zheng et al., 2013): this method calculates N(u) through the Pearson coefficient and adds the user's domain information to the matrix decomposition.
8. NAMF (Tang et al., 2016a): this method adds basic user information to matrix decomposition, filters domain users according to geographic location, and adds neighborhood information to matrix decomposition.
9. QoS prediction method based on GRU: this method is the reference experiment of this experiment. The difference between the two is that only the GRU network is included in method 9, while the method proposed in this study combines two neural networks: GRU and GAN; with the same point, because the input data of these two methods are the same, they can be used to predict the QoS value of the vacancy.

Based on Figure 5 and Figure 6, we found that the prediction accuracy based on MF is better than that based on the CF method, and the GRU–GAN-based method proposed in this study has the prediction accuracy of MF. Comparing the mean values of MAE under the four densities, the values of models in this study are decreased with a range from 0.325 to 0.044 and lower than UPCC, IPCC, UIPCC, CMF, NMF, PMF, NIMF, NAMF, and GRU. The decrement of the average value of RMSE ranges from 0.78 to 0.3, and it is lower than the aforementioned 9 methods. Our method takes longer periods to achieve better prediction results because our model has more parameters to be trained to better fit the training data, which are the characteristics of GRU–GAN.

5 Conclusion

Although the recommendation based on the services' function attributes satisfies users' demands for service function, it ignores the impact of the QoS on the user experience. To further improve users' satisfaction with service recommendations, people try to recommend services based on services' non-functional attributes. There is sparsity of the QoS matrix in the real world, which brings obstacles to service recommendation; hence, the prediction of QoS becomes a solution to overcome this obstacle. Scholars have tried to use collaborative filtering (CF) methods and matrix factorization (MF) methods to predict the QoS, but these methods face two challenges. The first challenge is the sparsity of data; the sparsity makes it difficult for CF to accurately determine whether users are similar, and the gap between the hidden matrices obtained by MF decomposition is large; the second challenge is the cold start of recommendation when new users (or services) participate in the recommendation; its historical record is vacant, making accurately predicting the QoS value be more difficult. To solve the aforementioned problems, this study mainly did the following work: 1) we organized the QoS matrix into a service call record, which contains user characteristic information and current QoS. 2) We proposed a QoS prediction method based on GRU–GAN. 3) We used the time series data for quality prediction and compared some QoS prediction methods, such as CF and MF. The results showed that the prediction results based on GRU–GAN are far superior to other prediction methods under the same data density.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Materials; further inquiries can be directed to the corresponding authors.

Author contributions

LY and YD contributed to the conception and design of the study. LY wrote the manuscript and performed the statistical analysis. LY helped supervise the project. LY completed the formatting and editing of the manuscript. All authors contributed to the manuscript revision, and read and approved the submitted version.

Funding

NSFC (Nos. 61962040 and 72062015), the Natural Science Foundation of Inner Mongolia Autonomous Region (No. 2022MS06024), and the Hainan University Educational Reform Research Project (Nos. HDJY2102 and HDJWJG03).

Acknowledgments

The authors would like to thank their students for their contributions.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., and Bharath, A. A. (2018). Generative adversarial networks: An overview. *IEEE Signal Process. Mag.* 35, 53–65. doi:10.1109/msp.2017.2765202

Deshpande, M., and Karypis, G. (2004). Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.* 22, 143–177. doi:10.1145/963770.963776

Hamilton, M. J. (2022). Collective computation, information flow, and the emergence of hunter-gatherer small-worlds. *J. Soc. Comput.* 3, 18–37. doi:10.23919/jsc.2021.0019

Hudson, N., Khamfroush, H., and Lucani, D. E. (2021). "QoS-aware placement of deep learning services on the edge with multiple service implementations," in 2021 International Conference on Computer Communications and Networks (ICCCN), Athens, Greece, 19-22 July 2021 (IEEE), 1–8.

Koren, Y. (2010). Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Trans. Knowl. Discov. Data* 4, 1–24. doi:10.1145/1644873.1644874

Lee, D. D., and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 788–791. doi:10.1038/44565

Li, T., Li, C., Luo, J., and Song, L. (2020). Wireless recommendations for internet of vehicles: Recent advances, challenges, and opportunities. *Intell. Converged Netw.* 1, 1–17. doi:10.23919/icn.2020.0005

Liu, J., Tang, M., Zheng, Z., Liu, X., and Lyu, S. (2015). Location-aware and personalized collaborative filtering for web service recommendation. *IEEE Trans. Serv. Comput.* 9, 686–699. doi:10.1109/tsc.2015.2433251

Lo, W., Yin, J., Deng, S., Li, Y., and Wu, Z. (2012b). "An extended matrix factorization approach for QoS prediction in service selection," in 2012 IEEE ninth international conference on services computing, Honolulu, HI, USA, 24-29 June 2012 (IEEE), 162–169.

Lo, W., Yin, J., Deng, S., Li, Y., and Wu, Z. (2012a). "Collaborative web service QoS prediction with location-based regularization," in 2012 IEEE 19th international conference on web services, Honolulu, HI, USA, 24-29 June 2012 (IEEE), 464–471.

Mnih, A., and Salakhutdinov, R. R. (2008). Probabilistic matrix factorization. *Adv. Neural Inf. Process. Syst.* 20, 1257–1264.

Nath, S., and Wu, J. (2020). Deep reinforcement learning for dynamic computation offloading and resource allocation in cache-assisted mobile edge computing systems. *Intell. Converged Netw.* 1, 181–198. doi:10.23919/icn.2020.0014

Nitu, P., Coelho, J., and Madiraju, P. (2021). Improvising personalized travel recommendation system with recency effects. *Big Data Min. Anal.* 4, 139–154. doi:10.26599/bdma.2020.9020026

Qi, L., Hu, C., Zhang, X., Khosravi, M. R., Sharma, S., Pang, S., et al. (2020a). Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment. *IEEE Trans. Ind. Inf.* 17, 4159–4167. doi:10.1109/tii.2020.3012157

Qi, L., Zhang, X., Li, S., Wan, S., Wen, Y., and Gong, W. (2020). Spatial-temporal data-driven service recommendation with privacy-preservation. *Inf. Sci.* 515, 91–102. doi:10.1016/j.ins.2019.11.021

Sandhu, A. K. (2021). Big data with cloud computing: Discussions and challenges. *Big Data Min. Anal.* 5, 32–40. doi:10.26599/bdma.2021.9020016

Shao, L., Zhang, J., Wei, Y., Zhao, J., Xie, B., and Mei, H. (2007). "Personalized QoS prediction for web services via collaborative filtering," in IEEE international conference on web services (ICWS 2007), Salt Lake City, UT, USA, 09-13 July 2007 (IEEE), 439–446.

Tang, M., Zheng, Z., Kang, G., Liu, J., Yang, Y., and Zhang, T. (2016a). Collaborative web service quality prediction via exploiting matrix factorization and network map. *IEEE Trans. Netw. Serv. Manage.* 13, 126–137. doi:10.1109/tnsm.2016.2517097

Tang, N., Xiong, Q., Wang, X., Gao, M., Wen, J., and Zeng, J. (2016b). Web service recommendation based on location clustering and tensor decomposition. *Comput. Eng. Appl.* 52, 65–72.

Tran, V. X., and Tsuji, H. (2009). A survey and analysis on semantics in QoS for web services. In 2009 International Conference on Advanced Information Networking and Applications, Bradford, UK, 26-29 May 2009 (IEEE), 379–385.

Wu, J., Chen, L., Feng, Y., Zheng, Z., Zhou, M. C., and Wu, Z. (2012). Predicting quality of service for selection by neighborhood-based collaborative filtering. *IEEE Trans. Syst. Man. Cybern. Syst.* 43, 428–439. doi:10.1109/tsmca.2012.2210409

Xu, Y., Yin, J., Lo, W., and Wu, Z. (2013). "Personalized location-aware QoS prediction for web services using probabilistic matrix factorization," in *International conference on web information systems engineering* ((Berlin, Heidelberg: Springer), 229–242.

Yin, Y., Aihua, S., Min, G., Yueshen, X., and Shuoping, W. (2016). Qos prediction for web service recommendation with network location-aware neighbor selection. *Int. J. Soft. Eng. Knowl. Eng.* 26, 611–632. doi:10.1142/s0218194016400040

Yu, C., and Huang, L. (2014). "Time-aware collaborative filtering for QoS-based service recommendation," in 2014 IEEE International Conference on Web Services, Anchorage, AK, USA, 27 June 2014 - 02 July 2014 (IEEE), 265–272.

Zhang, S., Liu, H., He, J., Han, S., Du, X., Lu, Y., et al. (2021). Myoblast differentiation of C2C12 cell may related with oxidative stress. *Intractable Rare Dis. Res.* 4, 173–178. doi:10.5582/irdr.2021.01058

Zhang, W., Chen, X., and Jiang, J. (2020a). A multi-objective optimization method of initial virtual machine fault-tolerant placement for star topological data centers of cloud systems. *Tsinghua Sci. Technol.* 26, 95–111. doi:10.26599/tst.2019.9010044

Zhang, Y., Zhang, H., Cosmas, J., Jawad, N., Ali, K., Meunier, B., et al. (2020b). Internet of radio and light: 5g building network radio and edge architecture. *Intell. Converged Netw.* 1, 37–57. doi:10.23919/icn.2020.0002

Zhang, Y., Zheng, Z., and Lyu, M. R. (2011). "Wspred: A time-aware personalized QoS prediction framework for web services," in 2011 IEEE 22nd international symposium on software reliability engineering, Hiroshima, Japan, 29 November 2011 - 02 December 2011 (IEEE), 210–219.

Zheng, Z., Ma, H., Lyu, M. R., and King, I. (2013). Collaborative web service QoS prediction via neighborhood integrated matrix factorization. *IEEE Trans. Serv. Comput.* 6, 289–299. doi:10.1109/tsc.2011.59

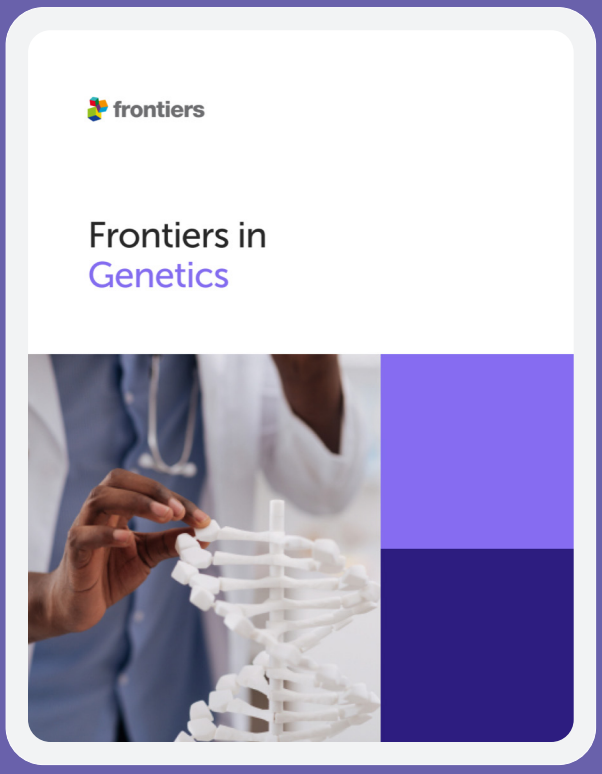
Zheng, Z., Ma, H., Lyu, M. R., and King, I. (2009). "Wsrec: A collaborative filtering based web service recommender system," in In 2009 IEEE International Conference on Web Services, Los Angeles, CA, USA, 06-10 July 2009 (IEEE), 437–444.

Frontiers in Genetics

Highlights genetic and genomic inquiry relating to all domains of life The most cited genetics and heredity journal, which advances our understanding of genes from humans to plants and other model organisms. It highlights developments in the function and variability of the genome, and the use of genomic tools.

Discover the latest Research Topics

See more →



Frontiers

Avenue du Tribunal-Fédéral 34
1005 Lausanne, Switzerland
frontiersin.org

Contact us

+41 (0)21 510 17 00
frontiersin.org/about/contact