Exploratory Research in Clinical and Social Pharmacy 14 (2024) 100463

Contents lists available at ScienceDirect

Exploratory Research in Clinical and Social Pharmacy

journal homepage: www.elsevier.com/locate/rcsop

"Using network analysis modularity to group health code systems and decrease dimensionality in machine learning models"

Mohsen Askar [a,*], Lars Småbrekke [a], Einar Holsbø [b], Lars Ailo Bongo [b], Kristian Svendsen [a]

[a] Department of Pharmacy, Faculty of Health Sciences, UiT-The Arctic University of Norway, PO Box 6050, Stakkevollan, N-9037 Tromsø, Norway
[b] Department of Computer Science, Faculty of Science and Technology, UiT-The Arctic University of Norway, PO, Box 6050 Stakkevollan, N-9037 Tromsø, Norway

ARTICLE INFO

Keywords:
Predictive modeling
Machine learning
Network analysis
Modularity detection
Healthcare coding systems
Categorical data encoding

ABSTRACT

Background: Machine learning (ML) prediction models in healthcare and pharmacy-related research face challenges with encoding high-dimensional Healthcare Coding Systems (HCSs) such as ICD, ATC, and DRG codes, given the trade-off between reducing model dimensionality and minimizing information loss.
Objectives: To investigate using Network Analysis modularity as a method to group HCSs to improve encoding in ML models.
Methods: The MIMIC-III dataset was utilized to create a multimorbidity network in which ICD-9 codes are the nodes and the edges are the number of patients sharing the same ICD-9 code pairs. A modularity detection algorithm was applied using different resolution thresholds to generate 6 sets of modules. The impact of four grouping strategies on the performance of predicting 90-day Intensive Care Unit readmissions was assessed. The grouping strategies compared: 1) binary encoding of codes, 2) encoding codes grouped by network modules, 3) grouping codes to the highest level of ICD-9 hierarchy, and 4) grouping using the single-level Clinical Classification Software (CCS). The same methodology was also applied to encode DRG codes but limiting the comparison to a single modularity threshold to binary encoding.
The performance was assessed using Logistic Regression, Support Vector Machine with a non-linear kernel, and Gradient Boosting Machines algorithms. Accuracy, Precision, Recall, AUC, and F1-score with 95% confidence intervals were reported.
Results: Models utilized modularity encoding outperformed ungrouped codes binary encoding models. The accuracy improved across all algorithms ranging from 0.736 to 0.78 for the modularity encoding, to 0.727 to 0.779 for binary encoding. AUC, recall, and precision also improved across almost all algorithms. In comparison with other grouping approaches, modularity encoding generally showed slightly higher performance in AUC, ranging from 0.813 to 0.837, and precision, ranging from 0.752 to 0.782.
Conclusions: Modularity encoding enhances the performance of ML models in pharmacy research by effectively reducing dimensionality and retaining necessary information. Across the three algorithms used, models utilizing modularity encoding showed superior or comparable performance to other encoding approaches. Modularity encoding introduces other advantages such as it can be used for both hierarchical and non-hierarchical HCSs, the approach is clinically relevant, and can enhance ML models' clinical interpretation. A Python package has been developed to facilitate the use of the approach for future research.

Abbreviations: ATC, Anatomical Therapeutic Chemical classification; AUC, Area Under the Curve; CCS, Clinical Classification Software; CPT, Current Procedural Terminology; DRG, Diagnosis Related Groups; ED, Emergency Department; EDA, Exploratory Data Analysis; GBM, Gradient Boosting Machine; HCPCS, Healthcare Common Procedure Coding System; HCSs, Healthcare Coding Systems; ICD, International Classification of Diseases; ICU, Intensive Care Unit; LR, Logistic Regression; ML, Machine Learning; NA, Network Analysis; RUS, Random UnderSampling; SVM, Support Vector Machine.

* Corresponding author.
E-mail addresses: mohsen.g.askar@uit.no (M. Askar), lars.smabrekke@uit.no (L. Småbrekke), einar.j.holsbo@uit.no (E. Holsbø), lars.ailo.bongo@uit.no (L.A. Bongo), kristian.svendsen@uit.no (K. Svendsen).

https://doi.org/10.1016/j.rcsop.2024.100463
Received 8 May 2024; Received in revised form 3 June 2024; Accepted 8 June 2024
Available online 11 June 2024

Introduction

The pharmacy sector is undergoing a significant transformation in many aspects from drug discovery to drug dispensing. This is due to the integration of automation, Artificial Intelligence (AI), Machine Learning (ML), and big data analysis.[1] Automation is now used in the process of drug dispensing, reducing human error and increasing efficiency.[2] AI and ML algorithms can analyze large datasets to identify drug candidates, predict clinical outcomes, and personalize treatments for individual patients.[3] Additionally, big data analysis provides insights to enhance decision-making and drive innovation in pharmaceutical research and development.[4]

In the realm of healthcare and pharmacy-related quantitative research, the datasets employed typically incorporate one or more features representing a high-dimensional Healthcare Coding System (HCS). These HCSs may have a hierarchical structure where broader categories are on the top and categories become more specific toward the bottom[5] e.g. the International Classification of Diseases (ICD),[6] the Anatomical Therapeutic Classification Codes (ATC) for medications,[7] and Diagnosis Related Group (DRG),[8] and non-hierarchical ones in which the system does not follow a hierarchical structure such as the Current Procedural Terminology (CPT)[9] and the Healthcare Common Procedure Coding System (HCPCS).[10]

The increasing adoption of ML models in healthcare has addressed the need to effectively handle high-dimensional HCSs. HCSs are crucial to achieving accurate predictions because of the amount of clinical and administrative information they carry. However, handling (encoding) these high-dimensional data poses some challenges such as increased computational complexity, risk of overfitting, and difficulty in model interpretability.[11] Efficiently handling these high-dimensional HCSs should compromise between reducing the computational burden and maintaining relevant information needed to enhance the performance of prediction models.

Encoding is the process of numerically representing the categorical values to be processible to algorithms that only process numerical values.[12] Encoding highly dimensional HCSs usually poses a challenge in ML models and is often considered one of the shortcomings in prediction modeling.[13] Handling these HCS with the classic encoding approaches such as one-hot or binary encoding approach will produce a binary vector equal to the length of the total number of health codes, hence will greatly increase model dimension, and memory requirements, demand more computational power, raise sparsity and complexity leading to decrease the model performance.[14]

Besides binary encoding, many other encoding approaches of high-dimensional categorical variables have been suggested in the literature such as hash encoding,[15] Word2Vec,[14] target encoding,[15] and similarity encoding.[16] While these encoding approaches could achieve good performance in prediction models, the balance between dimension reduction, loss of information, and enhanced clinical interpretation of the model output is still a discussion.

Other advanced approaches have also been suggested in the literature such as Deep Feature Synthesis (DFS), which automatically generates features from raw data to capture complex relationships.[17] Entity embedding, which uses neural networks to learn dense representations of categorical variables,[18] in other approaches.[19] These methods have shown promising results but can be complex to implement and are demanding in terms of computation resources.

Notably, there is no evidence framework for how to handle HCSs in predictive models.[20] Many studies tended to simplify these features by using the total count of codes (e.g. diagnoses, medications) as a numerical indicator,[21] creating a binary variable for each code,[22] or grouping codes to a higher hierarchical level[23,24] in other methods.[25] It has been shown that models using the lowest level of ICD-10 codes performed worse than higher-level codes in a prediction model.[26] While this approach (grouping to a higher hierarchical level) preserves the hierarchical relationships among the codes, it is unsuitable if the classification system is non-hierarchical. In addition, while these approaches will reduce the number of dimensions (features), they will omit the detailed information contained in the complete codes which might be necessary for a robust and reliable model prediction.[27]

Another approach is to group (aggregate) the levels of HCS according to specific schemes.[28–31] This approach is more convenient but can be limited by the hierarchical design of the scheme itself and the need to develop or update the schemes to suit different versions of the HCS. Kansal et al. investigated the impact of the grouping method of HCS on the model performance demonstrating that the grouping methods affect the model performance and that some grouping methods yield better performance than others.[20]

In this study, we introduce "Modularity Encoding" as a method to encode HCSs in ML models and demonstrate the use of the approach on two widely used HCSs in healthcare datasets, namely the ICD and the DRG codes. We compare the performance of ML models using modularity encoding to other popular encoding approaches. We also introduce a publicly available Python package that facilitates using the approach in future research.

Material and methods

Data sources

We utilized the Medical Information Mart for Intensive Care (MIMIC-III) dataset. MIMIC-III comprises 58,976 Intensive Care Unit (ICU) encounters with 46,467 unique patients at Beth Israel Deaconess Medical Center between 2001 and 2012. The full description of MIMIC-III is available elsewhere.[32] Variables with patients' demographics, admissions, and diagnoses were used in the prediction model. The variables' description is attached to Appendix 1. The full description of the dataset variables along with univariate and bivariate Exploratory Data Analysis (EDA) is attached to Appendix 2. To enhance the quality of reporting, the IJMEDI checklist for medical AI[33] was followed and reported in Appendix 3.

Data preprocessing

We excluded patients <18 years old, elective and newborn admissions, patients who died in the hospital, and missing and error-registered ICD codes. Appendix 1, Fig. 1. Illustrates the flow of exclusions. After the exclusions, the dataset comprised 29,247 unique patients and 37,762 unique admissions. The difference in days between admissions was calculated to determine the 90-day ICU readmissions as the models' outcome. Each admission was treated as an individual patient for simplification. We generated some indicator variables such as length of stay of hospital admissions, number of Emergency Department (ED) admissions, and length of stay in each ED visit. Missing values were treated as a separate category except if were found in the ICD-9 variable then were dropped as the diagnosis variable is central for the analysis. The final model included 12 dependent variables and the outcome variable (Appendix 1, Table 3). The dataset was checked for duplicates and outliers, and no intervention was necessary. The full list of initial and final variables included in the models is attached to Appendix 1.

Network generation

We used the patient ID and ICD-9 codes to generate a network in which the *nodes* represent the ICD-9 codes while the *edges* (connections) between these nodes represent the number of patients who shared the diagnoses pairs. We discussed the methodology for generating such types of networks in a previous study.[34] The total number of nodes in the ICD-9 network was 6840 with 952,676 edges. This network represents the multimorbidity patterns that exist in the dataset population.

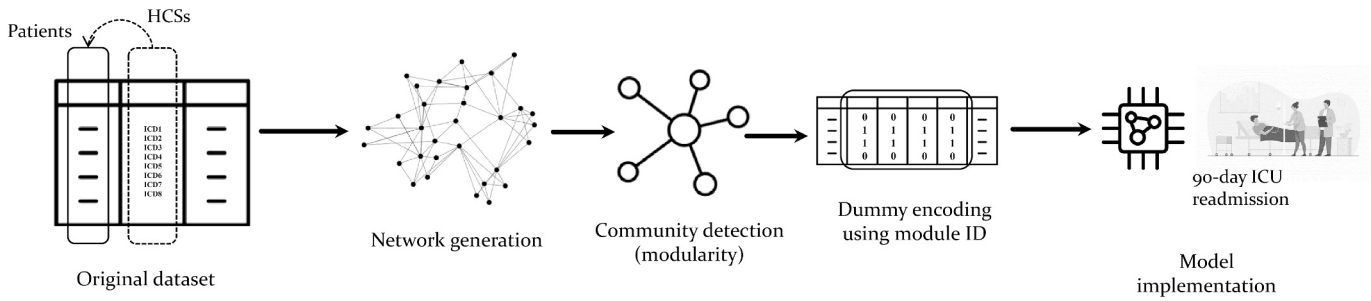

Fig. 1. The steps of conducting the Modularity Encoding approach. 1) A network is generated where the nodes are the HCS codes, and the edges are the co-occurrences of these codes in the patients' population. 2) Modules of strongly connected codes in the network were identified. 3) Each code was assigned the module id it belongs to in the network. 4) The HCS is binary encoded according to their module number, reducing the number of generated dimensions to correspond to the number of the detected modules in the network. 5) These new dimensions are used in the ML prediction models.

Modularity detection

We then used the Louvain algorithm for community detection[35] to identify the multimorbidity modules in the network using different resolution thresholds[36] to produce 6 different sets of modules ranging from 8 to 1078 modules (see Table 1). Modularity resolution is a parameter in the Louvain community detection algorithm that affects the size of the recovered modules. Applying higher resolutions results in fewer but larger communities, while lower resolutions lead to more and smaller ones. The randomizing option was specified as false to ensure the reproducibility of the communities. Tables of ICD-9 codes with their corresponding module number were extracted to be used later to encode the ICD codes in the datasets. The same methodology was applied to the DRG codes, the DRG network comprised 1661 nodes and 41,274 edges.

We compared the modularity grouping approach to two popular grouping approaches: the single-level Clinical Classification Software (CCS)[37] and grouping ICD to the highest categories in the ICD-9 classification system.[38] In total, we had 9 datasets to compare: 1) the raw dataset which includes all ICD codes, 2) six datasets of the different modularity resolutions, 3) the dataset in which ICD codes are grouped after the CCS scheme, and 4) a dataset in which ICD codes are grouped to the highest ICD-9 system hierarchy categories.

For the DRG codes, we only compared two datasets, a dataset where we binary encoded all DRG codes, and a modularity-grouped dataset at a single resolution threshold. Table 1. describes the datasets used in the study experiments.

Modeling

To minimize the potential effect of class imbalance[39] on the models' performance, we balanced the data using Random Undersampling (RUS).[40] We randomly matched each instance of the minority class (ICU readmissions) to a control instance of the majority class (non-readmissions), thereby ensuring an equal representation of both classes and a fair performance comparison across all models

Categorical variables were dummy encoded, and numerical ones were standardized. Standardizing was separately performed on the training set and applied on the testing set to avoid information leaks. We used a random split of data training and testing sets (70–30%). Three different ML algorithms were used; Logistic regression (LR), Support Vector Machine (SVM) with a non-linear kernel, and Gradient Boosting Machine (GBM) to retrospectively predict 90-day ICU readmissions. Five evaluation metrics were used to assess the performance of the models (accuracy, precision, recall, F1-score, and AUC). The definitions and formulas of these metrics along with models' confusion matrices are attached to Appendix 1. 95% Confidence Intervals (CI) were calculated by performing 1000 non-parametric bootstrap replicates on the test set. The steps of conducting the modularity encoding approach are illustrated in Fig. 1. The code used in the study is publicly available here https://anonymous.4open.science/r/modularity_encod_article-1FCC/.

Software

Stata 17 was used for data preprocessing and network generation, Gephi 0.10 for community detection and network visualization, and Python 3.6.12 and scikit-learn 1.3.1 were used for ML model implementation.

Package developing

To facilitate the use of the approach, we developed a Python package called "modularity_encoding", which is available on the Python Package Index (PyPI) and can be installed using the command "pip install modularity-encoding". The documentation of this package along with a user guide and demonstration examples of use is attached to Appendix 4.

Results

This section is organized into 4 experiments, the first three focus on ICD codes, and the last on DRG codes. The first experiment compares the model's performance using binary encoding of ungrouped ICD codes in the original dataset (Raw dataset) against grouping to the highest resolution modularity grouping on ICD codes (R1 dataset). The second one compares models' performance on ICD codes grouped on modules detected at different modularity resolutions, aiming to investigate if there is a specific modularity threshold that shows better performance. The third one compares models' performance using modularity grouping on the highest resolution to grouping to the highest ICD hierarchy and CCS scheme. The last experiment replicates the second experiment but on DRG codes instead of ICD ones.

Experiment 1. Binary encoding vs modularity encoding on ICD codes

The results show better results for models trained on the R1 dataset in terms of most metrics using the 3 classifying algorithms (see Table 2, Fig. 2).

Because of the great decrease in dimension after encoding, the training time is also significantly decreased, especially for SVM which is often a time-demanding algorithm. The training time was reduced from 83 min to train the model on the Raw ICD dataset to less than half a minute on the R1 ICD dataset) see Table 3.

Experiment 2. Different modularity resolution thresholds

The results mostly suggest that using higher resolution thresholds (fewer modules) often yields better results than applying lower thresholds. Notably, recall, F1-score, and accuracy values fall dramatically in SVM the more modules introduced to the dataset, which may throw

Table 1
The number of modules and features (dimensions) in the created datasets after binary encoding.

Dataset name	Modularity resolution threshold	Description	No. of modules/ groups	No. of all features after the binary encoding
ICD codes				
Raw	_	All ICD codes were dummy encoded into separate binary variables	_	4403
R1	1	ICD codes are grouped into 8 modules detected at a resolution threshold of 1	8 modules	56
R08	0.8	ICD codes are grouped into 16 modules detected at a resolution threshold of 0.8	16 modules	63
R06	0.6	ICD codes are grouped into 32 modules detected at a resolution threshold of 0.6	32 modules	95
R05	0.5	ICD codes are grouped into 47 modules detected at a resolution threshold of 0.5	47 modules	80
R01	0.1	ICD codes are grouped into 314 modules detected at a resolution threshold of 0.1	314 modules	362
R001	0.01	ICD codes are grouped into 1078 modules detected at a resolution threshold of 0.01	1078 modules	1123
Clinical Classification Software (CCS)	_	ICD codes are grouped into 285 categories of Single-level Clinical Classification Software.	285 groups	322
Highest ICD hierarchy (ICD_Highest)	_	The ICD codes are grouped to the highest level of the hierarchy.	18 groups	66
DRG codes				
Raw DRG	_	The raw form of the dataset. All DRG codes were dummy encoded in separate binary variables	_	1482
R1_M24	1	The DRG codes are grouped into 24 modules by resolution threshold 1	24 modules	64

some questions on the performance stability of the SVM algorithm in this experiment (see Table 4, Fig. 2). As of these results, we chose R1 resolution to use in the third experiment.

Experiment 3. Modularity encoding vs other grouping approaches

Models using grouping by the three methods yield very close performance. Specifically, models trained on R1 and the grouping to the highest level of the ICD hierarchy performed mostly better than models trained on CCS. Models trained ICD_Highest dataset generally performed slightly better than models trained on the R1 dataset, for example in terms of accuracy, recall, and F1-score (LR, SVM). While R1 performed best in terms of AUC (LR, GBM), precision (LR), and recall (GBM). Both ICD_Highest and R1 outperformed CCS in most metrics across the three algorithms. Notably, the performance of models trained by SVM falls in terms of accuracy, F1-score, and recall when trained on the CCS dataset (see Table 5, Fig. 2). Fig. 2 summarizes the results from these 3 experiments.

Experiment 4. Binary encoding vs modularity encoding on the DRG codes

Similar to the comparison between Raw and R1 datasets in the ICD codes, the models trained on modularity-grouped DRG codes yielded better results in almost all metrics for all three algorithms, Table 6.

Discussion

In this study, we suggest Modularity Encoding as a method of grouping HCSs using Network Analysis modularity. The usability of the approach was demonstrated on two healthcare coding systems: the ICD and DRG code systems. The approach can, nevertheless, be used to group other HCSs such as ATC codes.

Regardless of the approach used to encode HCSs, there will be always a trade-off between information loss and dimension reduction. However, deciding the encoding method could be crucial to obtain reliable performance results. An inefficient encoding technique could limit patients' characteristics and hence limit prediction performance.[13] Therefore, encoding such HCSs should follow a meaningful grouping scheme of aggregating the HCS codes in order to provide the model with the meaningful necessary information for better performance. Our purpose in this study was not to develop the best predictive model among state-of-the-art ML models but rather to compare the performance of some commonly used ML algorithms on the same dataset using different grouping schemes on two commonly used HCSs in ML prediction models. To enrich the comparison, we used three algorithms with three distinct algorithmic strategies, LR (linear algorithm), SVM with non-linear kernel, and GBM (tree-based boosting algorithm). The evaluation metrics were selected to cover both model performance (accuracy, AUC, F1-score) and clinical performance (precision, recall) as recommended here.[41]

After we aggregated each pair of health codes on the patient level, a network that represents the multimorbidity patterns in the dataset population was created. Network analysis modularity was proposed by Newman[26] and is defined as the measure of the structure of networks which is used to reveal the clusters (communities or modules) of the network. We used the Louvain modularity detection algorithm to group these health codes in the networks into modules. Louvain method is a popular community detection algorithm due to its simplicity, speed, and effectiveness in detecting network modules.[42] Each module represents a group of health codes that have denser connections between each other than the rest of the network. In the ICD codes network, each module represents a cluster of diagnoses that co-occur in the dataset population (i.e., multimorbidity).[43,44] If the nodes were the ATC codes instead, then the modules would represent the comedication pattern in the population.[34]

As expected, modularity encoding outperformed binary encoding of the ungrouped codes of HCSs for both ICD and DRG code systems. This may be because of the considerable number of dimensions in the latter which makes it more complicated for the algorithm to find the pattern in the dataset, i.e. curse of dimensionality.[45] Additionally, the more

Table 2
A comparison of different performance metrics comparing raw dataset binary encoding to modularity encoding. Bold font indicates the best performance in the comparison metric.

Algorithm	Dataset	Accuracy (95%CI)	Precision (95%CI)	Recall (95%CI)	F1-score (95%CI)	AUC (95%CI)
LR	**Raw**	0.727 (0.723–0.731)	0.764 (0.759–0.77)	**0.656** (0.649–0.662)	**0.71** (0.701–0.711)	0.804 (0.800–0.808)
	R1	**0.736** (0.727–0.746)	**0.782** (0.769–0.795)	0.654 (0.640–0.668)	**0.71** (0.701–0.723)	**0.813** (0.805–0.823)
SVM	**Raw**	0.73 (0.726–0.735)	**0.787** (0.781–0.793)	0.632 (0.626–0.639)	0.7 (0.696–0.707)	0.822 (0.816–0.824)
	R1	**0.778** (0.769–0.787)	0.765 (0.753–0.777)	**0.802** (0.790–0.815)	**0.78** (0.774–0.793)	**0.837** (0.828–0.845)
GBM	**Raw**	0.779 (0.775–0.783)	0.751 (0.746–0.757)	0.835 (0.831–0.840)	**0.79** (0.787–0.795)	**0.837** (0.834–0.842)
	R1	**0.78** (0.772–0.789)	**0.752** (0.740–0.764)	**0.836** (0.826–0.847)	**0.79** (0.826–0.847)	0.836 (0.828–0.845)

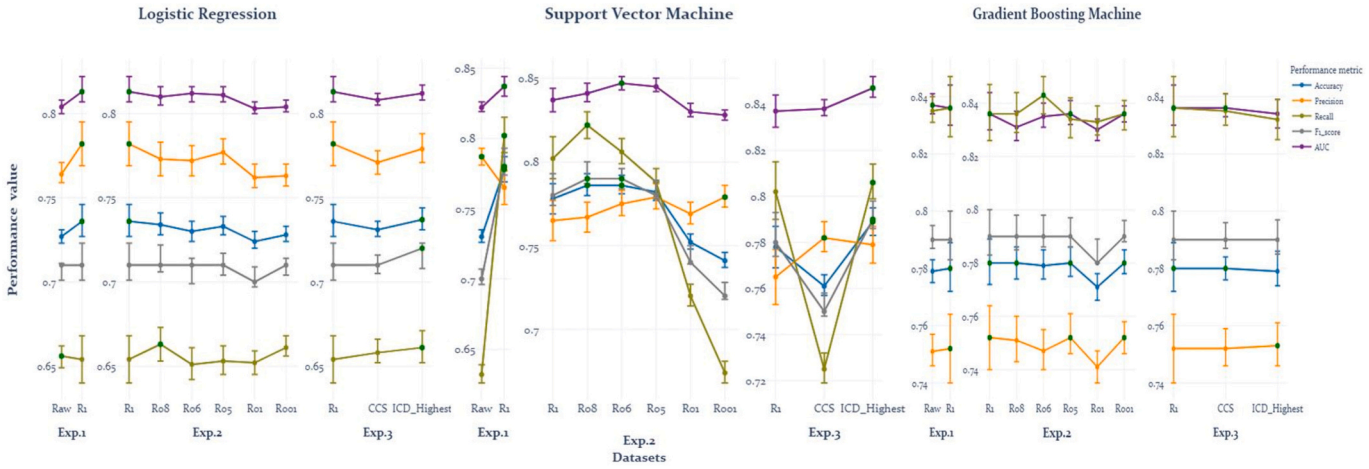

Fig. 2. Represents the results from the first three experiments. LR, SVM, and GBM were used in all experiments. For all models, accuracy, precision, recall, F1-score, and AUC metrics with 95% confidence intervals were used to evaluate the models' performances. Green markers indicate the highest value of the evaluation metric in the respective comparison. **Experiment 1**. (binary vs modularity encoding), to the left of each subfigure, shows generally better results of modularity grouping over dummy encoding of the raw ICD codes. In **experiment 2.** (different resolutions threshold encoding), the performance results of different resolutions are close. LR and GBM models suggest that R1 is the best resolution threshold, while SVM suggests that R08 is the best. In **experiment 3**. (comparison of modularity, highest hierarchy, and CCS encoding), grouping ICD codes to the highest level of system hierarchy yielded generally best results, followed by modularity grouping and CCS. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
The training time of the models on the different ICD datasets.

	Raw	R1	R08	R06	R05	R01	R001	CCS	ICD_Highest
LR	14 s	**0.1 s**	0.4 s	0.8 s	0.7 s	1.5 s	2.7 s	1.6 s	0.6 s
SVM	83 m	**0.5 m**	1 m 24 s	2 m 29 s	3 m 5 s	8 m	25 m	7 m 17 s	2 m 9 s
GBM	6 m 2 s	**2 s**	4 s	7 s	9 s	30 s	1 m 20 s	26 s	6 s

sparsity of the dataset will worsen the algorithm discrimination performance.[46] The substantial dimension reduction obtained by grouping codes on modules significantly reduces the model training time maintaining, or even improving, model performance in many cases.

We also tested if the different thresholds of modularity resolutions would notably affect the results. Despite some worsening in model recall in some models (i.e., SVM), we saw no considerable differences in the overall model performance on datasets encoded after different resolution thresholds. However, in general, with the thresholds we experimented with, we saw that the fewer modules will perform generally better suggesting that grouping codes into fewer, larger modules may capture more clinically relevant multimorbidity patterns. This also likely enhances the models' ability to generalize from training data to unseen data.

We further compared modularity grouping against two other commonly used approaches to group ICD codes, namely single-level CCS and grouping to the highest hierarchy of the ICD classification system. A previous study highlighted the impact of grouping methods of diagnosis codes on model performance. They compared raw codes, truncated codes, grouping using AHRQ-Elixhauser,[47] and single-level CCS. They showed the difference in the performance of models using different schemes of groups and suggested using CCS grouping as a baseline for future models.[20] They also showed that grouping using AHRQ-Elixhauser[47] performed worse on MIMIC-III dataset compared to the highest hierarchy grouping and single-level CCS, hence we did not include it in our analysis. In experiment 3, the performance was generally similar between the three grouping methods (Table 5, Fig. 2). However, the results show slightly higher performance of the highest hierarchy grouping, followed by modularity grouping and then CCS respectively. The significant reduction in the number of dimensions and training time

Table 4
A comparison of different performance metrics of models where ICD codes are encoded using different modularity resolutions. Bold font indicates the best performance in the comparison metric.

Logistic Regression					
	Accuracy	**Precision**	**Recall**	**F1-score**	**AUC**
R1	**0.736** (0.727–0.746)	**0.782** (0.769–0.795)	0.654 (0.640–0.668)	0.71 (0.701–0.723)	**0.813** (0.805–0.823)
R08	0.734 (0.728–0.741)	0.773 (0.763–0.783)	0.663 (0.653–0.673)	0.71 (0.706–0.722)	0.81 (0.804–0.817)
R06	0.73 (0.724–0.736)	0.772 (0.763–0.781)	0.651 (0.642–0.661)	0.71 (0.699–0.714)	0.812 (0.807–0.817)
R05	0.733 (0.728–0.739)	0.777 (0.770–0.785)	0.653 (0.645–0.662)	0.71 (0.704–0.717)	0.811 (0.806–0.817)
R01	0.724 (0.720–0.730)	0.762 (0.756–0.770)	0.652 (0.645–0.659)	0.7 (0.697–0.709)	0.803 (0.799–0.808)
R001	0.728 (0.724–0.733)	0.763 (0.757–0.770)	**0.661** (0.656–0.668)	0.71 (0.704–0.714)	0.804 (0.801–0.809)
Non-Linear Support Vector Machine					
R1	0.778 (0.769–0.787)	0.765 (0.753–0.777)	0.802 (0.790–0.815)	0.78 (0.774–0.793)	0.837 (0.828–0.845)
R08	**0.786** (0.780–0.793)	0.767 (0.758–0.776)	**0.822** (0.814–0.830)	**0.79** (0.787–0.800)	0.841 (0.835–0.847)
R06	**0.786** (0.781–0.792)	0.775 (0.768–0.783)	0.806 (0.799–0.814)	**0.79** (0.784–0.796)	**0.847** (0.840–0.851)
R05	0.782 (0.777–0.788)	**0.779** (0.772–0.787)	0.788 (0.781–0.796)	0.78 (0.778–0.789)	0.845 (0.841–0.850)
R01	0.752 (0.748–0.757)	0.769 (0.763–0.776)	0.72 (0.714–0.727)	0.74 (0.739–0.750)	0.83 (0.825–0.834)
R001	0.741 (0.737–0.746)	**0.779** (0.773–0.786)	0.674 (0.668–0.681)	0.72 (0.718–0.728)	0.828 (0.820–0.828)
Gradient Boosting Machine					
R1	**0.78** (0.772–0.789)	**0.752** (0.740–0.764)	0.836 (0.826–0.847)	**0.79** (0.826–0.847)	**0.836** (0.828–0.845)
R08	**0.78** (0.774–0.786)	0.751 (0.743–0.760)	0.836 (0.829–0.844)	**0.79** (0.785–0.798)	0.831 (0.825–0.837)
R06	0.779 (0.774–0.785)	0.747 (0.740–0.755)	**0.843** (0.836–0.850)	**0.79** (0.786–0.798)	0.835 (0.830–0.841)
R05	**0.78** (0.775–0.786)	**0.752** (0.746–0.761)	0.834 (0.827–0.842)	**0.79** (0.786–0.797)	0.836 (0.831–0.841)
R01	0.771 (0.766–0.776)	0.741 (0.735–0.747)	0.833 (0.828–0.839)	0.78 (0.780–0.789)	0.83 (0.826–0.834)
R001	**0.78** (0.776–0.785)	0.752 (0.746–0.758)	0.836 (0.830–0.841)	**0.79** (0.788–0.796)	**0.836** (0.832–0.840)

Table 5
A comparison of different performance metrics of different grouping strategies models. Bold font indicates the best performance in the comparison metric.

Logistic Regression					
	Accuracy	**Precision**	**Recall**	**F1-score**	**AUC**
R1	0.736 (0.727–0.746)	**0.782** (0.769–0.795)	0.654 (0.640–0.668)	0.71 (0.701–0.723)	**0.813** (0.805–0.823)
CCS	0.731 (0.727–0.736)	0.771 (0.764–0.778)	0.658 (0.652–0.666)	0.71 (0.705–0.716)	0.808 (0.804–0.813)
ICD_Highest	**0.737** (0.731–0.744)	0.779 (0.771–0.788)	**0.661** (0.652–0.671)	**0.72** (0.708–0.723)	0.812 (0.807–0.819)
Non-Linear Support Vector Machine					
R1	0.778 (0.769–0.787)	0.765 (0.753–0.777)	0.802 (0.790–0.815)	0.78 (0.774–0.793)	0.837 (0.828–0.845)
CCS	0.761 (0.757–0.766)	**0.782** (0.776–0.789)	0.725 (0.719–0.732)	0.75 (0.748–0.758)	0.838 (0.833–0.841)
ICD_Highest	**0.789** (0.783–0.795)	0.779 (0.771–0.787)	**0.806** (0.799–0.814)	**0.79** (0.786–0.798)	**0.847** (0.841–0.853)
Gradient Boosting Machine					
R1	**0.78** (0.772–0.789)	0.752 (0.740–0.764)	**0.836** (0.826–0.847)	**0.79** (0.826–0.847)	**0.836** (0.828–0.845)
CCS	**0.78** (0.776–0.784)	0.752 (0.746–0.759)	0.835 (0.830–0.841)	**0.79** (0.787–0.796)	**0.836** (0.833–0.841)
ICD_Highest	0.779 (0.774–0.786)	**0.753** (0.746–0.761)	0.832 (0.825–0.839)	**0.79** (0.785–0.797)	0.834 (0.828–0.839)

Table 6
The performance of prediction of models trained on raw binary encoded DRG codes dataset compared to modularity encoded DRG codes dataset.

Logistic Regression					
	Accuracy	**Precision**	**Recall**	**F1-score**	**AUC**
Raw_DRG	0.68 (0.675–0.688)	0.694 (0.684–0.705)	**0.644** (0.635–0.654)	0.67 (0.661–0.677)	0.749 (0.742–0.756)
R1_M24	**0.701** (0.691–0.712)	**0.734** (0.719–0.750)	0.629 (0.615–0.645)	**0.68** (0.665–0.691)	**0.771** (0.761–0.782)
Non-Linear Support Vector Machine					
Raw_DRG	0.682 (0.676–0.689)	0.688 (0.679–0.698)	0.667 (0.658–0.676)	0.68 (0.670–0.685)	0.756 (0.751–0.762)
R1_M24	**0.721** (0.711–0.731)	**0.72** (0.707–0.734)	**0.723** (0.709–0.738)	**0.72** (0.710–0.733)	**0.794** (0.785–0.805)
Gradient Boosting Machine					
Raw_DRG	0.715 (0.709–0.722)	0.688 (0.680–0.698)	0.786 (0.778–0.794)	0.73 (0.727–0.741)	0.787 (0.781–0.794)
R1_M24	**0.73** (0.720–0.741)	**0.701** (0.689–0.716)	**0.8** (0.788–0.814)	**0.75** (0.737–0.759)	**0.798** (0.788–0.809)

without worsening the model performance suggests that modularity encoding was efficient in summarizing but keeping sufficient clinical information for accurate predictions, which makes the approach a practical solution to encode large healthcare datasets.

Besides model performance, modularity grouping has additional advantages over the other approaches. Unlike other systems that are designed for specific HCSs and may require updates to align with the code system changes such as CCS, modularity encoding relies on the

patterns found in the dataset and dynamically reflects these patterns in the HCSs codes grouping.

Modularity grouping is a data-driven method that does not rely on the outcome variable. This means that it will draw the pattern of grouping from the studied dataset itself and does not need to be updated for different versions of the health code system.

Unlike approaches that used grouping to a higher level of hierarchy as a grouping method, modularity encoding does not assume the system hierarchy which makes it usable for both hierarchical and non-hierarchical systems as it derives its grouping from the occurrences in the dataset. Grouping using hierarchy inherently assumes that hierarchical similarity is a factor that could enhance model predictions. While this could be true sometimes, it is not always the case as it does not always reflect the actual clinical realities. For example, in grouping to the highest level of ICD, grouping codes will correspond to their hierarchical similarities (ex. infectious diseases, neoplasms (tumors), and mental disorders will be grouped together) which could be useful but neglects that multimorbidity patterns do not necessarily occur between similar conditions and diseases but rather between different ones, which gives the modularity grouping an advantage from the clinical point of view.

By exploring the HCS codes in each detected module in the network, modularity encoding can additionally enhance the model's clinical interpretation. Many studies have investigated the clinical patterns of diseases using the Network Analysis approach[43,44,48–50] revealing meaningful clinical patterns in each module. Table 7 represents a comparison between some popular approaches of grouping and the modularity one.

Table 7
A comparison summary of grouping approaches used in this study.

Aspect	Raw data dummy encoding	Clinical Grouping (e.g., CCS)	Grouping to the highest hierarchy of the system	Modularity grouping
Training time	−	+	+	+
Clinical relevancy	−	+	−	+
Enhance interpretation	−	+	−	+
Prediction performance	−	+	+	+
Dimension reduction	−	+	+	+
Universality for code systems	+	−	+	+
Easy implementation	+	−	+	−
Introducing additional clinical info.	−	+	−	+
No need for updating/change with updating the code system	+	−	+	+
Suitability for non-hierarchical code systems	−	−	−	+

Advantage (+), disadvantage (−).

The study findings have many practical implications for healthcare and pharmacy research. The approach can be used in a wide variety of settings including studies of drug use and co-medication. It could be widely adopted to enhance predictive modeling in clinical settings. It could also be utilized to enhance clinical interpretations of models' predictions. Furthermore, reduction of computational resources and training time makes the approach a practical solution for encoding large healthcare datasets, which can be a challenge to handle with traditional encoding approaches.

Our study has some limitations. We used MIMIC-III dataset which contains data from a single institution limiting the generalizability of our results to other populations. Additionally, some methodological choices were made during the study such as handling each instance in the dataset as a separate patient which could have ignored the correlation between related patient's information and affected the prediction accuracy. We also balanced the outcome classes to minimize the impact of class imbalance on the comparison results, this will affect the models' performance in real-world scenarios where the datasets are naturally imbalanced. While these choices will potentially affect the performance results of the study models, their impact is expected to be equal across all models, hence unlikely to bias the comparisons.

Modularity encoding as an approach has also some limitations. The approach demands the presence of a type of relationship between the HCSs codes in the dataset (e.g. a multimorbidity or comedication pattern) and the researcher must decide how to represent these relations in the network (i.e. defining the network edges). The approach is also a multistep one and could be quite complex to implement. It adds extra steps to the data preparation steps which represent more data preprocessing burden. Additionally, building big networks and performing modularity detection could demand high computational power. To ease implementing modularity encoding, we developed a Python package that performs the encoding using a simple syntax.

Conclusion

This study demonstrated modularity encoding as a method to encode HCSs in ML models. The approach enhances the performance and interpretability of prediction models while capturing clinically relevant patterns, reducing model dimensions, training time, and computational resources. Future research should focus on applying modularity encoding to various HCSs, diverse populations, advanced machine learning techniques, and other clinical outcomes prediction.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

CRediT authorship contribution statement

Mohsen Askar: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Lars Småbrekke:** Writing – review & editing, Supervision. **Einar Holsbø:** Writing – review & editing, Supervision. **Lars Ailo Bongo:** Writing – review & editing, Supervision, Conceptualization. **Kristian Svendsen:** Writing – review & editing, Supervision, Project administration, Conceptualization.

Declaration of Competing Interest

The authors declare no conflict of interest in conducting this research.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.rcsop.2024.100463.

References

1. Khan O, Parvez M, Kumari P, Parvez S, Ahmad S. The future of pharmacy: how AI is revolutionizing the industry. *Intell Pharm*. 2023;1:32–40. https://doi.org/10.1016/j.ipha.2023.04.008.
2. Chalasani SH, Syed J, Ramesh M, Patil V, Pramod Kumar TM. Artificial intelligence in the field of pharmacy practice: a literature review. *Explor Res Clin Soc Pharm*. 2023;12, 100346. https://doi.org/10.1016/j.rcsop.2023.100346.
3. Maleki Varnosfaderani S, Forouzanfar M. The role of AI in hospitals and clinics: transforming healthcare in the 21st century. *Bioengineering*. 2024;11:337. https://doi.org/10.3390/bioengineering11040337.

4. Raghupathi W, Raghupathi V. Big data analytics in healthcare: promise and potential. *Heal Inf Sci Syst*. 2014;2:3. https://doi.org/10.1186/2047-2501-2-3.
5. Cimino JJ. Desiderata for controlled medical vocabularies in the twenty-first century. *Methods Inf Med*. 1998;37:394–403. http://www.ncbi.nlm.nih.gov/pubmed/9865037.
6.. W.H. Organization. International Classification of Diseases (ICD). https://www.who.int/standards/classifications/classification-of-diseases; 2023. accessed September 9, 2023.
7.. W.H. Organization. WHOCC - ATC/DDD Index. https://www.whocc.no/atc_ddd_index/; 2023. accessed September 9, 2023.
8. Fetter RB, Shin Y, Freeman JL, Averill RF, Thompson JD. Case mix definition by diagnosis-related groups. *Med Care*. 1980;18:i–53. http://www.jstor.org/stable/3764138.
9. Hirsch JA, Leslie-Mazwi TM, Nicola GN, et al. Current procedural terminology; a primer. *J Neurointerv Surg*. 2015;7:309–312. https://doi.org/10.1136/neurintsurg-2014-011156.
10. U.S.N.L. of Medicine, UMLS Metathesaurus - HCPCS (HCPCS - Healthcare Common Procedure Coding System) - Source Representation. https://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/HCPCS/sourcerepresentation.html; 2024.
11. Miotto R, Wang F, Wang S, Jiang X, Dudley JT. Deep learning for healthcare: review, opportunities and challenges. *Brief Bioinform*. 2018;19:1236–1246. https://doi.org/10.1093/bib/bbx044.
12. Dahouda MK, Joe I. A deep-learned embedding technique for categorical features encoding. *IEEE Access*. 2021;9:114381–114391. https://doi.org/10.1109/ACCESS.2021.3104357.
13. Wang H, Cui Z, Chen Y, Avidan M, Ben Abdallah A, Kronzer A. Predicting hospital readmission via cost-sensitive deep learning. *IEEE/ACM Trans Comput Biol Bioinforma*. 2018;15:1968–1978. https://doi.org/10.1109/TCBB.2018.2827029.
14. Johnson JM, Khoshgoftaar TM. Encoding high-dimensional procedure codes for healthcare fraud detection. *SN Comput Sci*. 2022;3:362. https://doi.org/10.1007/s42979-022-01252-4.
15. Pargent F, Pfisterer F, Thomas J, Bischl B. Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features. *Comput Stat*. 2022;37:2671–2692. https://doi.org/10.1007/s00180-022-01207-6.
16. Cerda P, Varoquaux G. Encoding high-cardinality string categorical variables. *IEEE Trans Knowl Data Eng*. 2020;34:1164–1176. https://doi.org/10.1109/TKDE.2020.2992529.
17.. Kanter JM, Veeramachaneni K. Deep feature synthesis: Towards automating data science endeavors. In: *2015 IEEE Int. Conf. Data Sci. Adv. Anal*. IEEE; 2015:1–10. https://doi.org/10.1109/DSAA.2015.7344858.
18.. Guo C, Berkhahn F. Entity Embeddings of Categorical Variables. http://arxiv.org/abs/1604.06737; 2016.
19. Hancock JT, Khoshgoftaar TM. Survey on categorical data for neural networks. *J Big Data*. 2020;7:1–41. https://doi.org/10.1186/S40537-020-00305-W/FIGURES/4.
20. Kansal A, Gao M, Balu S, et al. Impact of diagnosis code grouping method on clinical prediction model performance: a multi-site retrospective observational study. *Int J Med Inform*. 2021;151. https://doi.org/10.1016/j.ijmedinf.2021.104466. N.PAG-N.PAG.
21. Kulkarni P, Smith L, Woeltje K, Smith LD, Woeltje KF. Assessing risk of hospital readmissions for improving medical practice., health care. *Manag Sci*. 2016;19:291–299. https://doi.org/10.1007/s10729-015-9323-5.
22. Futoma J, Morris J, Lucas J. A comparison of models for predicting early hospital readmissions. *J Biomed Inform*. 2015;56:229–238. https://doi.org/10.1016/j.jbi.2015.05.016.
23. Yu K, Xie X. Predicting hospital readmission: a joint ensemble-learning model. *IEEE J Biomed Heal Informat*. 2020;24:447–456. https://doi.org/10.1109/JBHI.2019.2938995.
24. Pakbin A, Rafi P, Hurley N, Schulz W, Harlan Krumholz M, Bobak Mortazavi J. Prediction of ICU readmissions using data at patient discharge. In: *Inst. Electr. Electron. Eng. Inc. Conf. Proc*; 2018:4932–4935. https://www.proquest.com/conference-papers-proceedings/prediction-icu-readmissions-using-data-at-patient/docview/2126684977/se-2?accountid=17260.
25.. Singh A, Nadkarni G, Guttag J, Bottinger E. *Leveraging Hierarchy in Medical Codes for Predictive Modeling*. 2023. https://doi.org/10.1145/2649387.2649407.
26. Deschepper M, Eeckloo K, Vogelaers D, Waegeman W. A hospital wide predictive model for unplanned readmission using hierarchical ICD data. *Comput Methods Prog Biomed*. 2019;173:177–183. https://doi.org/10.1016/j.cmpb.2019.02.007.
27. Shameer K, Johnson KW, Yahi A, et al. Predictive modeling of hospital readmission rates using electronic medical record-wide machine learning: a CASE-study using Mount SINAI HEART failure cohort. *Pac Symp Biocomput*. 2017;22:276–287. https://doi.org/10.1142/9789813207813_0027.
28. Fenn A, Davis C, Buckland DM, et al. Development and validation of machine learning models to predict admission from emergency department to inpatient and intensive care units. *Ann Emerg Med*. 2021;78:290–302. https://doi.org/10.1016/j.annemergmed.2021.02.029.
29.. Zhao P, Yoo I, Naqvi SH. Early prediction of unplanned 30-day hospital readmission: model development and retrospective data analysis. *JMIR Med Inform*. 2021;9(3):E16306. Https//Medinform.Jmir.Org/2021/3/E16306 9 (2021) e16306 https://doi.org/10.2196/16306.
30. Panicacci S, Donati M, Fanucci L, Bellini I, Profili F, Francesconi P. Population health management exploiting machine learning algorithms to identify high-risk patients, 2018 31ST. *IEEE Int Symp Comput Med Syst (CBMS)*. 2018;2018:298–303. https://doi.org/10.1109/CBMS.2018.00059.
31.. Hong WS, Haimovich AD, Taylor RA. Predicting hospital admission at emergency department triage using machine learning. *PLoS One*. 2018;13, e0201016. https://doi.org/10.1371/journal.pone.0201016.
32. Johnson AEW, Pollard TJ, Shen L, et al. MIMIC-III, a freely accessible critical care database. *Sci Data*. 2016;3, 160035. https://doi.org/10.1038/sdata.2016.35.
33. Cabitza F, Campagner A. The need to separate the wheat from the chaff in medical informatics. *Int J Med Inform*. 2021;153, 104510. https://doi.org/10.1016/j.ijmedinf.2021.104510.
34. Askar M, Cañadas RN, Svendsen K. An introduction to network analysis for studies of medication use. *Res Soc Adm Pharm*. 2021;17:2054–2061. https://doi.org/10.1016/j.sapharm.2021.06.021.
35. Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. *J Stat Mech Theory Exp*. 2008;2008:P10008. https://doi.org/10.1088/1742-5468/2008/10/P10008.
36. Lambiotte R, Delvenne J-C, Barahona M. Laplacian dynamics and multiscale modular structure in networks. *IEEE Trans Netw Sci Eng*. 2009;1:76–90. https://doi.org/10.1109/TNSE.2015.2391998.
37.. W.H. Organization. Clinical Classifications Software (CCS) for ICD-9-CM. https://hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp; 2023. accessed August 9, 2023.
38.. W.H. Organization. ICD - ICD-9-CM - International Classification of Diseases, Ninth Revision, Clinical Modification. https://www.cdc.gov/nchs/icd/icd9cm.htm; 2023.
39. He Haibo, Garcia EA. Learning from imbalanced data. *IEEE Trans Knowl Data Eng*. 2009;21:1263–1284. https://doi.org/10.1109/TKDE.2008.239.
40. Tanha J, Abdi Y, Samadi N, Razzaghi N, Asadpour M. Boosting methods for multi-class imbalanced data classification: an experimental review. *J Big Data*. 2020;7:70. https://doi.org/10.1186/s40537-020-00349-y.
41. Norgeot B, Quer G, Beaulieu-Jones BK, et al. Minimum information about clinical artificial intelligence modeling: the MI-CLAIM checklist. *Nat Med*. 2020;26:1320–1324. https://doi.org/10.1038/s41591-020-1041-y.
42. Hu B, Li W, Huo X, Liang Y, Gao M, Pei P. Improving Louvain algorithm for community detection. In: *Proc. 2016 Int. Conf. Artif. Intell. Eng*. Paris, France: Appl., Atlantis Press; 2016. https://doi.org/10.2991/aiea-16.2016.20.
43. Guo M, Yu Y, Wen T, et al. Analysis of disease comorbidity patterns in a large-scale China population. *BMC Med Genet*. 2019;12:177. https://doi.org/10.1186/s12920-019-0629-x.
44.. Chen Y, Xu R. *Network Analysis of Human Disease Comorbidity Patterns Based on Large-Scale Data Mining*. 2014:243–254. https://doi.org/10.1007/978-3-319-08171-7_22.
45. Keogh E, Mueen A. Curse of dimensionality. *Encycl Mach Learn Data Min*. 2017:314–315. https://doi.org/10.1007/978-1-4899-7687-1_192.
46. Jiang B, Deng C, Yi H, et al. XDL: An industrial deep learning framework for high-dimensional sparse data. In: *Proc. 1st Int. Work. Deep Learn. Pract. High-Dimensional Sparse Data, ACM*. 2019:1–9. https://doi.org/10.1145/3326937.3341255. New York, NY, USA.
47. Elixhauser A, Steiner C, Harris DR, Coffey RN. Comorbidity measures for use with administrative data. *Med Care*. 1998;36:8–27. https://doi.org/10.1097/00005650-199801000-00004.
48. Zhou D, Wang L, Ding S, Shen M, Qiu H. Phenotypic disease network analysis to identify comorbidity patterns in hospitalized patients with ischemic Heart disease using large-scale administrative data. *Healthcare*. 2022;10:80. https://doi.org/10.3390/healthcare10010080.
49. Mu X-M, Wang W, Jiang Y-Y, Feng J. Patterns of comorbidity in hepatocellular carcinoma: a network perspective. *Int J Environ Res Public Health*. 2020;17:3108. https://doi.org/10.3390/ijerph17093108.
50. Jones I, Cocker F, Jose M, Charleston M, Neil AL. Methods of analysing patterns of multimorbidity using network analysis: a scoping review. *J Public Health (Bangkok)*. 2023;31:1217–1223. https://doi.org/10.1007/s10389-021-01685-w.