International Journal of Machine Learning and Cybernetics (2025) 16:1507–1529
https://doi.org/10.1007/s13042-024-02349-3

ORIGINAL ARTICLE

Clustered Automated Machine Learning (CAML) model for clinical coding multi-label classification

Akram Mustafa[1] · **Mostafa Rahimi Azghadi**[1]

Received: 24 August 2023 / Accepted: 17 August 2024 / Published online: 3 September 2024

Abstract
Clinical coding is a time-consuming task that involves manually identifying and classifying patients' diseases. This task becomes even more challenging when classifying across multiple diagnoses and performing multi-label classification. Automated Machine Learning (AutoML) techniques can improve this classification process. However, no previous study has developed an AutoML-based approach for multi-label clinical coding. To address this gap, a novel approach, called Clustered Automated Machine Learning (CAML), is introduced in this paper. CAML utilizes the AutoML library Auto-Sklearn and cTAKES feature extraction method. CAML clusters binary diagnosis labels using Hamming distance and employs the AutoML library to select the best algorithm for each cluster. The effectiveness of CAML is evaluated by comparing its performance with that of the Auto-Sklearn model on five different datasets from the Medical Information Mart for Intensive Care (MIMIC III) database of reports. These datasets vary in size, label set, and related diseases. The results demonstrate that CAML outperforms Auto-Sklearn in terms of Micro F1-score and Weighted F1-score, with an overall improvement ratio of 35.15% and 40.56%, respectively. The CAML approach offers the potential to improve healthcare quality by facilitating more accurate diagnoses and treatment decisions, ultimately enhancing patient outcomes.

Keywords Machine learning · AutoML · Multi-label classification · NLP · Clinical coding · Computer assisted coding

1 Introduction

Artificial Intelligence (AI) plays a crucial role in addressing healthcare challenges. The integration of AI and big data analytics is considered a powerful tool for analyzing extensive datasets. AI assists doctors in diagnosing diseases with greater precision [1, 2] and accelerates drug and vaccine development through real-time diagnosis, monitoring, and treatment, as supported by recent research [3, 4] and hardware implementation of deep learning algorithms [5]. Recently, ChatGPT and Google Gemini have been leveraged to address challenges within medical education and simulate doctor-patient communication. These applications show potential for Clinical Document Improvement (CDI), demonstrating the diverse ways in which AI, exemplified by ChatGPT, contributes to enhancing various facets of the healthcare domain [6–9].

Clinical coding is a critical process in healthcare that involves assigning standardized codes to patient diagnoses and procedures. Despite advancements in technology, this process still heavily relies on human decision-making to identify and assign codes accurately. While various machine learning models have been implemented to aid clinical coders, such models still require extensive skills and expertise of data scientists to train and optimize them [10, 11]. Thus, the current question in the research community is whether clinical coding can be fully automated and whether Automated Machine Learning (AutoML) libraries can provide better results than the standard methods.

The field of AutoML has seen remarkable growth, resulting in the development of many models and tools. AutoML libraries take diverse approaches to multi-label classification, often using algorithms native to this task or creating separate models for each label. However, relying on native algorithms sometimes leads to faster processing speeds at the cost of lower accuracy. Conversely, building per-label

✉ Akram Mustafa
akram.mohdmustafa@my.jcu.edu.au

Mostafa Rahimi Azghadi
mostafa.rahimiazghadi@jcu.edu.au

[1] College of Science and Engineering, James Cook University, Townsville 4811, QLD, Australia

models often yields high accuracy but slower performance [12, 13].

This paper examines the feasibility and potential of AutoML for automated clinical coding, considering the challenges and limitations that exist in clinical coding and the capabilities of AutoML libraries. A novel clustered AutoML model is proposed, demonstrating significant performance and accuracy enhancements contrasted with a conventional AutoML approach for multi-label clinical coding classification.

Several studies have investigated the effectiveness of machine learning in comparison to rule-based methods for clinical coding, including research by Sheikhalishahi et al. [14], Obeid et al. [15], and Yogarajan et al. [16]. These studies have shown that machine learning methods generally outperform rule-based approaches. They used various Natural Language Processing (NLP) techniques, such as Bag Of Words (BOW), Continuous Bag Of Words (CBOW), skip-gram, N-gram, and MetaMap. While the majority of algorithms used were Support Vector Machine (SVM) and Naïve Bayes, the choice of algorithm had a significant impact on the results obtained, particularly for single diagnoses.

The aforementioned studies establish the potential of machine learning in clinical coding and motivated our research to advance the field through the use of AutoML. In particular, AutoML is employed in conjunction with clustering algorithms, leading to the implementation of new Clustered Automated Machine Learning (CAML) models for clinical coding.

To the best of our knowledge, no previous studies have employed AutoML for the classification of medical notes across multiple diagnoses. Also, no research employing a clustered model in an automated machine learning approach has come to our attention. This research is a significant step forward in the field of Multi-Label classification using AutoML, particularly in the context of clinical coding. The proposed approach, which develops a Clustered Automated Machine Learning (CAML) Model, involves converting unstructured medical reports into a tabular feature set using the Unified Medical Language System (UMLS) Concept Unique Identifiers (CUIs) through the clinical Text Analysis and Knowledge Extraction System (cTAKES). cTAKES [17] is an open-source solution developed collaboratively by Mayo Clinic and various universities. It is a Java-based application that harnesses Natural Language Processing (NLP), Artificial Intelligence (AI), and Role-based algorithms. cTAKES is designed to analyze clinical notes, including discharge summaries, using predefined libraries like Snomed, ICD10, and many others. It then converts many medical terminology elements such as medications, symptoms, diagnoses, lab results, and anatomy into Snomed codes, ICD10 codes, Concept Unique Identifiers (CUIs), and more.

Following this, the model clusters diagnoses using a modified Hamming distance. These clusters are fed into an AutoML ensemble, selecting the most effective algorithms. This approach enhances clinical coding accuracy and efficiency, traditionally a manual and time-consuming process. Therefore, our research has the potential to improve healthcare quality by enabling more accurate diagnoses and treatments.

Previous research results emphasize the potential of machine learning in clinical coding, highlighting its superiority over rule-based methods. Using algorithms and feature extraction techniques like cTAKES and BOW, these studies often focused on single-diagnosis models. The complexity of clinical diagnoses necessitates advanced multi-label classification solutions

However, this study reveals significant limitations in the use of machine learning for clinical coding, particularly for multi-label classification. The challenge lies in the trade-off between performance and accuracy. Native multi-label classification algorithms offer higher performance models compared to constructing a model for each label to achieve greater accuracy. Addressing these challenges is crucial for improving the performance and accuracy of both types of models.

In summary, while prior research has laid the groundwork for automated clinical coding, challenges remain in achieving robust multi-label classification. This study advances existing work by introducing the CAML model and assessing its effectiveness in addressing these challenges, thereby contributing to the ongoing evolution of automated clinical coding.

2 Background

2.1 Clinical coding and computer assisted coding

In the healthcare industry, around 80% of the data generated by healthcare organizations, is unstructured [18], which presents a significant challenge for healthcare providers who want to leverage this data for analysis and integration with other healthcare solutions. The transformation of unstructured medical reports into structured data has, therefore, become an essential process for healthcare organizations. This transformation can be useful in data analysis, integration with other solutions and processes such as Clinical Documentation Improvement (CDI), and Revenue Cycle Management (RCM) systems [19, 20]. Machine learning models in the form of Computer Assisted Coding systems (CACs), can help this transformation by automating the clinical coding process.

The use of CACs has numerous advantages, including improved clinical coding accuracy, reduced time and cost

needed for the coding process, and integration with other healthcare systems. Currently, there are a few CAC systems available in the market, such as 3 M 360 Encompass [21] and DeepMed CodeDoctor [22]. However, despite these advancements, CAC systems have not yet reached a level of independence where they can function without human intervention. Clinical coders are still required to confirm the accuracy of diagnoses and procedures before they can be used in further processes [23–25].

In addition, building CAC models require human skills, such as data science and Natural Language Processing (NLP). Currently, there is no CAC solution that uses AutoML technologies to identify the best ML algorithm for high accuracy coding based on the healthcare organization's training dataset. This means that the development of CACs still requires human input to ensure their effectiveness. A potential intriguing automation path is the use of AutoML in CAC solutions. This could be a significant step forward, allowing healthcare providers to leverage the large amount of unstructured data generated in the industry effectively [10, 11].

2.2 Machine learning for clinical coding

Clinical coding has been extensively studied in the literature, with a focus on developing models that improve coding classification accuracy. These models have utilized various techniques at different stages of their machine learning development, including data preparation. Gehrmann et al. used cTAKES as the main tool to prepare clinical notes data. Their work demonstrates that models using cTAKES have a comparable F1-score compared to 2-gram and 3-gram models, with the exception of CNN Models [26, 27]. Feature extraction methods, such as Bag of Words (BOW) [28, 29], Term Frequency-Inverse Document Frequency (TF-IDF) [30, 31], and cTAKES [32, 33], have been employed. These models have also made use of feature selection techniques like χ^2 [34], Information Gain (IG) [35], and Leave-One-Out (LOO) [36], along with algorithm selection techniques, including random forest [30, 37], logistic regression [38], and Support Vector Machines (SVMs) [39], in addition to configuring and evaluating the selected algorithms.

The Deep Neural Network algorithms have demonstrated effectiveness as models for clinical coding classifications; however, they represent a distinct approach from AutoML, which offers an automated solution encompassing all steps of the machine learning process [40]. In the extensive examination of research papers within the realm of clinical coding, cTAKES emerged as a noteworthy tool delivering favorable outcomes in the realm of data preparation for clinical notes. cTAKES has been widely utilized in numerous studies [32, 33]. Interestingly, a notable gap in the literature is the absence of any papers employing AutoML in the context of clinical coding. This void presents various opportunities for enhancing the outcomes of clinical coding classification.

Most of the studies in the literature on diagnosis classification problems have focused on single-label classification [15, 41–43], with only a few targeting multi-diagnosis classification [16]. Clinical coding systems are usually evaluated using various metrics such as F1-score [44], recall [38], precision [45], and Area Under the Curve (AUC) [46, 47], among others [15, 31, 48].

Automated machine learning techniques have been proposed to streamline the development of clinical coding models and automate the aforementioned model development steps. In a previous article [40], An in-depth review of the methods and techniques enabling AutoML model development for clinical coding and, more broadly, clinical notes analysis was provided. Overall, the extensive research in clinical coding has produced a range of effective models that can accurately classify medical diagnoses, enabling healthcare providers to better manage patient data and improve the quality of care.

2.3 International Classification of Diseases

The International Classification of Diseases (ICD) is a comprehensive and widely-used tabulated list of diseases issued by the World Health Organisation (WHO) [49, 50]. It serves as a standardized system for identifying and classifying medical conditions and is adopted by healthcare organizations worldwide [51]. Currently, the most recent revision of the ICD is ICD-11, although it has not been adopted by as many healthcare organizations as its predecessor, ICD-10 [52].

ICD-10 is currently used by numerous healthcare organizations worldwide [53, 54]. Some countries, however, have modified ICD codes and issued their own revisions of ICD-10. For instance, Australia issued the ICD-10 AM, which stands for Australian Modification [55], the United States issued the ICD-10 CM (Clinical Modification) [56], and Canada issued ICD-10 CA through the Canadian Institute for Health Information (CIHI) [57]. Additionally, there is a procedure code version called ICD-10 PCS, which is the American version [49, 51].

ICD-9, which preceded ICD-10, has approximately 13,000 different codes, which belong to 18 main categories referred to as Level 1 [58]. For example, codes from 001 to 139 denote infectious and parasitic diseases, codes from 140 to 239 denote neoplasms, and additional E and V codes are used for external causes of injury and supplemental classification. Under each category, there are more detailed subcategories referred to as Level 2, such as codes from 050 to 059 for viral diseases accompanied by exanthem. The disease name constitutes Level 3, such as code 053 for Herpes zoster, and goes into further detail for the fourth and fifth

levels of the disease, such as 053.1 for Herpes zoster with other nervous system complications and 053.11 for Geniculate herpes zoster (Fig. 1) [30, 37, 59–62].

2.4 Model's evaluation

Model evaluation can be approached through various methods, with the selection of the appropriate method contingent upon factors such as the target labels' characteristics, whether numeric or categorical, the nature of the business model, and the specific objectives.

Numerous studies [16, 26, 30, 33, 38] have explored the task of classifying clinical diagnoses from clinical notes, and this has been well-documented in previous research [40]. Among the methods frequently employed in this domain, the F1-score stands out as a popular choice.

The F1-score leverages both precision and recall, as defined by the following equation:

$$F1 = 2(Precision.Recall)/(Precision + Recall), \quad (1)$$

where precision measures the proportion of true positives among predicted positives, and recall represents the proportion of true positives captured among all actual positives.

In the context of multi-label classification, there exist three variations of the F1-score. First, the Micro F1-score computes the F1-score across all records and all classes. Conversely, the Macro F1-score determines the average F1-score for each class individually and then computes the overall average. Lastly, the Weighted F1-score calculates the average F1-score for each class, taking into account the weight assigned to each class. The choice of which F1-score variant to employ depends on the specific evaluation requirements and objectives [63].

2.5 Clinical coding with multi-label classification (MLC)

Clinical coding is an essential task in the healthcare industry. This task is challenging due to the large number of labels involved in the ICD system [64, 65]. Many studies have utilized various neural network techniques and algorithms to improve the accuracy of clinical coding in multi-label classification, i.e. when more than one ICD code should be assigned to one record, namely multi-label coding [16, 38, 66].

CNN models have been used in several studies. For instance, Karmakar [59] used a CNN to classify medical reports. Because the ICD fifth level has a large number of labels, Karmakar employed two approaches. The first approach involved using the 20 most common labels of level 5, these labels represent the ICD codes that exhibited the highest frequency within the dataset. While the second approach utilized all level 1 ICD codes, limiting the target set to 17 labels only. The results showed that both approaches achieved high accuracy.

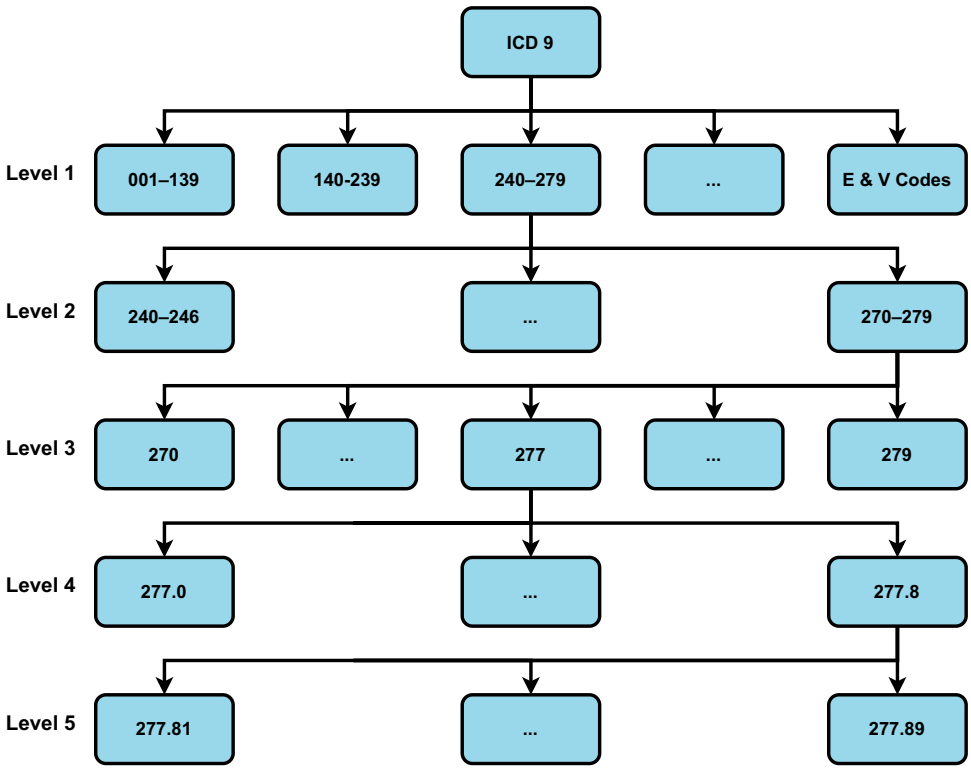

Fig. 1 ICD-9 hierarchy

Similarly, Gehrmann et al. [26] employed a CNN model in a phenotyping experiment with 10 phenotype labels, utilizing a binary classification approach for each label. The CNN model outperformed models like Linear Regression and Random Forest, yielding higher F1-scores. Additionally, Xu [67] demonstrated the effectiveness of CNNs in classifying 32 ICD labels, with individual models for each label, and found that the CNN model consistently achieved superior accuracy compared to other models in their study.

In another study, Huang et al. [30] compared different deep neural network algorithms, such as CNN, LSTM, Gated Recurrent Unit (GRU), and Feed Forward Neural Network, along with other statistical algorithms like Logistic Regression and Random Forest. The study utilized ICD9 Level 4 (Codes) and ICD9 Level 3 (Categories) and built four different models for the top 10 level 4 codes, top 10 level 3 codes, top 50 level 4 codes, and top 50 level 3 codes. The results showed that Recurrent Neural Network (RNN) models (GRU and LSTM) achieved the highest accuracy, while RNN models along with Linear Regression had the highest F1-scores.

Binary relevance, as illustrated in Fig. 2, Section A, is an MLC technique that converts labels into a set of binary codes, either 0 or 1, and builds a separate model for each label in the targeted label set. In this technique, each label is independent and does not depend on the relations between labels [16, 38, 60]. However, in some cases, clinical diseases (labels) can be dependent, and one disease can be an indication of another disease, as is the case with diabetes and hypertension [68].

Classifier Chain is a technique for handling label dependence in MLC problems. It employs a chain of binary classifiers, each trained to predict a label and all preceding labels in the chain. Each classifier's output becomes a feature for the next one, incorporating label dependencies into the prediction process [16, 62, 69, 70]. Figure 2, Section B illustrates this process. Classifier Chains can work through positive chaining, where diseases are linked, or negative chaining, where diseases are mutually exclusive [71]. An example of negative chaining is hemophilia type A which is related to chromosome X and ovarian cancer which is a female-only disease [72]. This technique is valuable for complex MLC tasks where label dependencies play a crucial role in accurate predictions.

Finding relations between labels is an essential task in MLC techniques. However, a major challenge in achieving this task is the time complexity involved in identifying these relations. As the number of labels increases, the time taken to identify these relationships also increases significantly, resulting in a big-O complexity of $2\binom{n}{2} = n(n-1)$.

Label Powerset is a widely used approach for transforming a multi-label label set into a single label [38].

In comparison, Random K-Labelset (RAKEL) [73] extends this approach by randomly selecting label subsets for model training and classification. Another approach to multi-label classification is the Hierarchy of Multi-label Classifiers (HOMER) [74], which aims to convert large multi-label target sets into smaller hierarchical label sets. The effectiveness of these techniques has been demonstrated in various studies [70, 75–77]. Figure 2, Section C illustrates the Label Powerset technique, and Fig. 2, Section D illustrates the Hierarchy of Multi-label Classifiers technique.

Correlation and Hamming Loss are common techniques utilized for MLC [71, 74, 78–80]. Hamming Loss measures the percentage of unmatched labels to the total number of labels in the target set. Many studies have employed Hamming Loss as a measure to compare the actual and predicted label values [69, 81]. Huang [30], for instance, compared different models using Hamming Loss, AUC, and Precision. In another study, Su et al. [82] evaluated ten TPOT models using various evaluation methods, including Hamming Loss, Kappa Score, and Accuracy, among others [83].

2.6 AutoML with multi-label classificatioin

The field of automated machine learning has experienced significant growth, with a multitude of models and tools

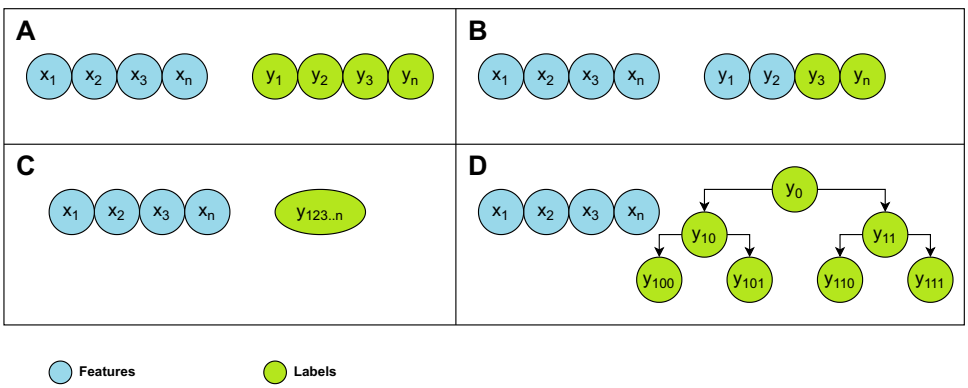

Fig. 2 Multi-label classification techniques. **A** Binary relevance (BR). **B** Classifier chain (CC). **C** Label power (LP). **D** Hierarchal labels

being developed [84]. Some of these tools are limited to predetermined models and workflows, exemplified by Rapid Mine [85]. Alternatively, there exist tools that possess the ability to identify the optimal algorithm based on the characteristics of the training dataset and the desired outcome. Auto-WEKA [86, 87] is one of these tools. Auto-WEKA is a Java-based application that builds upon the functionality of WEKA. Specifically, it is designed to address the Combined Algorithm Selection and Hyperparameter Optimization (CASH) problem using single label classifiers. WEKA has undergone significant improvements since its initial release, culminating in the development of an extended version called MEKA [88]. This updated platform includes multi-label classifiers, multi-label targeted label transformation techniques (e.g., binary relevance and classifier chain), and multi-label evaluation methods. MEKA has also paved the way for the development of Auto-MEKA, an automated machine learning application that builds on the functionality of MEKA. Recently, De Sa et al. [89] developed Auto-MEKA$_{GGP}$, which is an automated machine learning model based on Grammar Genetic Programming that uses MEKA's MLC algorithms and configurations.

Auto-Sklearn [90–92] has emerged as a prominent AutoML framework, winning numerous AutoML competitions. However, it lacks native support for multi-class multi-label classification, which necessitates the utilization of alternative methods to convert MLC labels into another form, such as one-vs-all. Nonetheless, not all Auto-Sklearn algorithms can be employed with a one-vs-all structure. Consequently, Auto-Sklearn must either classify each label individually as a single-label model, which significantly impacts performance or exclusively employ algorithms that support multi-label classification. This issue has been discussed in prior research [12, 13].

3 The research problem

Clinical coding refers to the process of manually identifying and assigning codes to diseases from medical reports such as discharge summaries [93]. This process, typically carried out by healthcare professionals including doctors and clinical coders, and involves a significant amount of time and effort in analyzing medical reports to extract patients' disease information. While various models have been developed to aid clinical coders in this process, the selection of an appropriate algorithm and model often requires the expertise of data scientists. For instance, Shi et al. [46] utilized a Long Short-Term Memory (LSTM) algorithm based on their prior research and personal experience. It should be noted that many of these models tend to focus on a specific disease rather than the entire range of diseases a patient might have. For example, Gehrmann et al. [26] focused solely on the primary disease among the list of diseases that patients had been diagnosed with.

While several studies have explored multi-label classification using deep neural network algorithms, there is no fully automated machine learning framework specializing in clinical coding that is currently known to us. Therefore, the proposition is to use AutoML libraries, such as Auto-Sklearn, in combination with feature extraction methods, such as cTAKES, as a viable approach to automate clinical coding processes. By utilizing these techniques, it would be possible to identify algorithms that offer the highest level of disease identification accuracy, without relying on human expertise.

As previously stated in Sect. 2.6, Auto-Sklearn can manage multi-label target sets either by employing multiple single-label classifications for each label y_i; or by exclusively employing algorithms that natively support multi-label targets. The first approach involves significant processing time because the time required to classify the entire set of labels is equivalent to the sum of the individual processing times for each label classification, which can be expressed as: $AutoSklearn(Y) = \Sigma_{i=1}^{n} AutoSklearn(y_i)$.

This processing time may be acceptable for smaller label sets, but it can significantly decrease performance for larger sets of labels. To illustrate, suppose it takes 2 min to process a single label; in that case, it would take over 3 h to classify a set of 100 labels.

The second approach of restricting Auto-Sklearn to models that inherently support multi-label classification has also been explored [13, 94]. The proposed approach benefits from the effectiveness of algorithms tailored for multi-label classification, enabling their execution in a single iteration encompassing all labels simultaneously. However, it is important to note that this approach may encounter limitations due to the availability of algorithms exclusively designed to handle multi-label classification. Consequently, this constraint imposes a restriction on the range of algorithms that can be effectively employed.

In summary, clinical coders dedicate considerable time to manually identify ICD codes from discharge summaries. Presently, various machine learning models, including those utilizing deep neural networks, are used to assist clinical coders in identifying ICD codes through multi-label classification. However, no research has been found that uses AutoML for the classification of multiple ICD codes. The adoption of AutoML in the clinical coding process offers the potential to decrease human involvement and allows for experimentation with multiple algorithms simultaneously, thereby enhancing the accuracy of results.

4 The proposed approach

The present study introduces a Clustered Automated Machine Learning (CAML) model for clinical coding multi-label classification, which outperforms the Auto-Sklearn model's F1-score for multi-label multi-class classification while maintaining a comparable timeframe for a large number of features. CAML utilizes an ensemble of algorithms that collaboratively work on a single dataset, leading to a superior model that yields a higher F1-score. Below, the details of the proposed approach will be explained.

4.1 NLP process

4.1.1 Building feature set

The initial stage of our methodology entails working with clinical records in the form of unstructured data and utilizing natural language processing techniques to convert them into structured data. This process begins with the use of the cTAKES to identify medical terminologies and convert medical text into a comprehensive set of clinical Concept Unique Identifiers (CUIs), which are incorporated into the Unified Medical Language System (UMLS). An instance of such a transformation is the conversion of the phrase "Chief Complaint: headache and neck stiffness Major Surgical or Invasive Procedure: central line placed, arterial line placed History of Present Illness" to "C0004482 C0004482 C0224473 C0004482 C0224473 C0719349 C0230431 C0420607 C4074814 C0719349", where each code starting with C denotes a CUI. Subsequently, the BOW technique, which is one of the feature extraction methods in natural language processing, is applied to the CUI list, quantifying these CUIs into a structured feature set.

4.1.2 Building labels

Medical reports include diagnoses that are originally presented in a list format, with each report having multiple diagnoses. The Binary Relevance (BR) technique is employed to transform this list format into a large multi-label binary label set. The ultimate dataset comprises a feature set consisting of Bag-of-Words (BOWs) of cTAKES Concept Unique Identifiers (CUIs) and binary codes of diagnoses. An illustration of the structure of the final dataset is presented in Table 1. In this table, each row shows a medical report that has been transformed into a feature set using cTAKES and BOWs representation. The features in this table are presented in CUI format. Additionally, the table displays the ICD codes associated with each medical report. A value of 1 is assigned to an ICD code if the medical report has been diagnosed with that particular code. Conversely, a value of 0 is assigned if the medical report does not have a diagnosis for that specific ICD code. In this table, one can observe that report 1012 details a patient's diagnosis with both ICD 300 and ICD 303. Additionally, report 1013's patient has received diagnoses of ICD 295, ICD 300, and ICD 560, while report 1014's patient has been diagnosed with ICD 295 and ICD 303.

4.2 Label clustering

In the proposed approach, a modified Hamming distance method is presented for determining the distance between each pair of diagnoses. Our approach involves constructing a binary array for each label result, which is subsequently used to compare each diagnosis in the label set with other diagnoses. The distance between diagnoses is expressed as a percentage of unmatching results compared to the total number of results. Furthermore, it is noted that the inverted diagnoses relation shares the same distance as the matching relation.

To illustrate the application of the approach, an example is provided in which the calculation of distances between diagnoses is performed using a binary array (see Table 2). Specifically, diagnoses 295 and 303 have 6 matching reports out of a total of 9, resulting in a 66.7% match. Diagnoses 295 and 560 have a 22.2% match, while diagnoses 330 and 560 have a 55.6% match. In cases where the percentage of unmatching results is less than 50%, the distance is calculated as the unmatching percentage. However, if the unmatching percentage exceeds 50%, the distance is expressed as the matching percentage. This is due to the negative correlation relation between labels. For example, in our example, the distance between diagnoses 295 and 303 is 33.3%, the distance between diagnoses 295 and 560 is 22.2%, and the distance between diagnoses 303 and 560 is 44.4%.

Table 1 Dataset after NLP process

Report ID	Features					ICDs				
	C0004482	C0224473	C0719349	C0230431	C0420607	295	300	303	540	560
1012	6	0	0	4	2	0	1	1	0	0
1013	0	2	2	8	0	1	1	0	0	1
1014	0	0	4	4	9	1	0	1	0	0

Table 2 Diagnoses binary array

Diagnosis	Rep 1	Rep 2	Rep 3	Rep 4	Rep 5	Rep 6	Rep 7	Rep 8	Rep 9
295	1	1	0	0	0	0	1	1	0
303	0	1	0	0	0	1	0	1	0
560	0	0	1	1	0	1	0	1	1

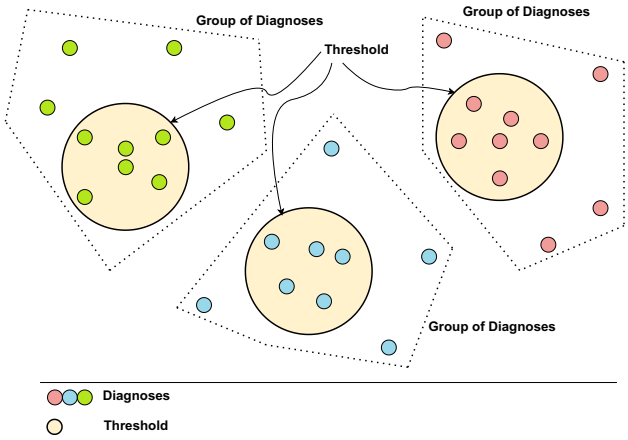

Fig. 3 Illustrate the grouping of diagnoses within the threshold

After the label results set is processed, diagnoses are grouped into multiple clusters based on their distances, with those diagnoses that are closer to each other being assigned to the same cluster. To ensure that only diagnoses that are very close to each other are included in a given cluster, a threshold or distance limit may be set. Figure 3 illustrates the grouping of diagnoses within the specified threshold.

4.3 Algorithm processing

In lieu of incorporating all labels in the classification procedure, the proposed approach involves selecting a single label from each group for model processing. Additionally, any labels beyond the prescribed threshold distance are handled individually, in conjunction with the selected label from each group. A visual representation of this methodology is depicted in Fig. 4.

In this figure, the multi-label dataset undergoes clustering and grouping based on the distance between labels. The result is a partitioning of the multi-label dataset into sets of groups, where each group aggregates multiple labels under a single representation, while any remaining individual labels are left ungrouped. Each group is then subjected to an algorithm designed to optimize label accuracy within that specific cluster. For the ungrouped individual labels, they are treated as separate single-label datasets.

The collection of involves the ensemble of distinct algorithms chosen for each group forms an ensemble that collectively maximizes label accuracy across the entire dataset.

4.3.1 Auto-Sklearn

The present study employs the Audo-Sklearn classification models framework to determine the algorithm that yields the highest F1-score for the designated labels within each group. This approach allows for all available algorithms to be considered in the selection process. Subsequently, the identified algorithm is applied to process all labels within the respective group. In cases where the chosen algorithm supports multi-label classification, the labels within that group are amalgamated and processed using a single multi-label classification model. However, if the selected algorithm does not support multi-label classification, each label within the group is processed individually.

5 The experiment

5.1 Dataset

The Medical Information Mart for Intensive Care (MIMIC III) is currently the only available dataset that contains medical reports. This dataset is composed of over 2 million medical reports of various types including radiology reports, nursing reports, nutrition reports, and others. For the purpose of this research, the focus was on a subset of MIMIC III, which comprises 59,652 discharge summary reports.

To facilitate testing, the discharge summary dataset was randomly split into multiple subsets of varying sizes, including 1000, 2000, 4000, 8000, and 16,000 reports. This approach allowed us to obtain multiple datasets for testing purposes.

5.1.1 Feature set

The current study utilized cTAKES to extract CUIs and the BOW method to quantify the extracted CUIs in order to construct the feature set. The discharge summary dataset, comprising 59,652 clinical reports, produced 28,615

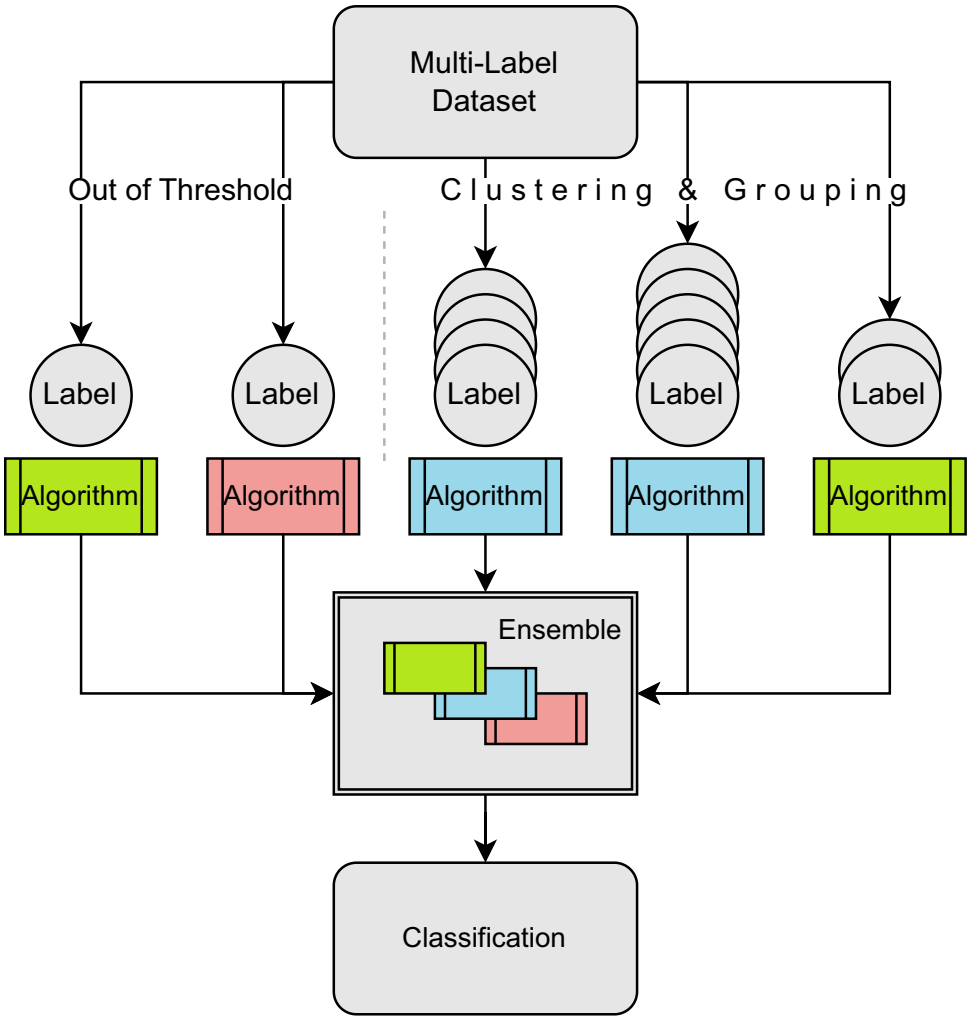

Fig. 4 Illustration of building algorithm ensemble

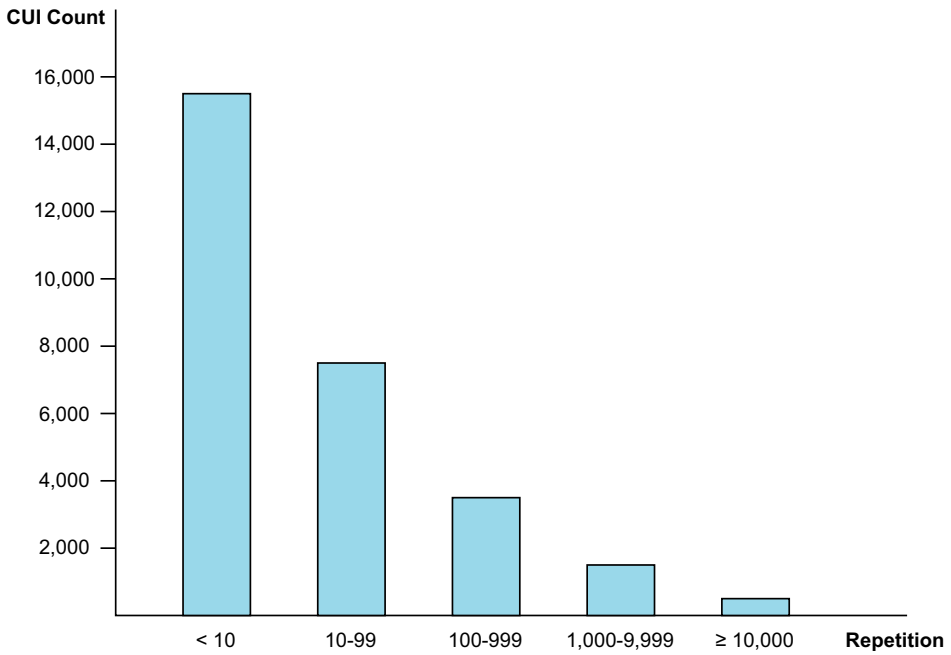

Fig. 5 CUIs repetition in discharge summary dataset

distinct CUIs. A graphical representation of the CUI count distribution is provided in Fig. 5. The distribution indicates that 15,415 CUIs appeared in less than 10 clinical reports, 7690 CUIs appeared in between 10 and 99 clinical reports, 3721 CUIs appeared in between 100 and 999 clinical reports, 1517 CUIs appeared in between 1000 and 9999 clinical reports, and 272 CUIs appeared in more than 10,000 clinical reports. Due to a large number of features, only the top 10,000 CUIs, ordered by the sum of their appearance counts, were selected for inclusion in the limited feature set.

5.1.2 Label set

Each report in the MIMIC III dataset contains a list of diagnoses in ICD9 format. In the discharge summary reports subset that has been selected, there are 6918 distinct ICD9 codes at the most granular level (i.e., 5th, 4th, or 3rd level). The diagnoses in these reports are classified into three categories: 1069 Level 3 ICD9 codes, 165 Level 2 ICD9 codes, and 18 Level 1 ICD9 codes. Due to the large number of codes, the dataset was further divided into Level 2, as well as Level 3 ICD9 codes related to the following categories, which are also shown in Table 3: Neoplasms (ICD9: 140-239), Injury And Poisoning (ICD9: 800-999), Supplementary Classification of External Causes of Injury and Poisoning, and Supplementary Classification of Factors Influencing Health Status and Contact with Health Services (ICD9: E800-E999 and V01-V82).

In this table, there is a total of 20 datasets grouped into four distinct categories. The first group is dedicated to Neoplasms (ICD9: 140-239) and features five datasets with varying sizes: 1000 records, 2000 records, 4000 records, 8000 records, and 16,000 records. These datasets have targets of ICD9 Level 3 labels ranging from 72 to 89 and consist of 10,000 features, except for the 1000 records dataset, which includes 9570 features.

The Injury & Poisoning (ICD9: 800-999) and Supplementary (ICD9: E & V) datasets are similar to the Neoplasms datasets in structure. Each category also includes five datasets with sizes ranging from 1000 to 16,000 records. They have targets of ICD9 Level 3 labels numbering between 113 and 192. These datasets consist of 10,000 features, with the exception of the 1000 records dataset, which contains 9364 features for Injury & Poisoning and 9224 features for Supplementary.

The fourth group of datasets comprises all ICD9 codes and maintains similarities to the Neoplasms category. This group also encompasses five datasets of sizes ranging from 1000 to 16,000 records. The ICD9 labels used are of Level 2 and range from 141 to 161. Like the previous groups, these datasets feature 10,000 features, with the exception of the 1000 records dataset, which includes 8999 features.

5.2 The environment

In this study, AutoML models were used for the analysis of large datasets with up to 10,000 features and more than 100

Table 3 Final datasets metadata

Dataset name	Diagnoses	ICD level	Reports #	Features #	Used features #	Labels #
Neoplasms (L3-1k)	140-239	Level 3	1000	9570	9570	72
Neoplasms (L3-2k)	140-239	Level 3	2000	11,982	10,000	82
Neoplasms (L3-4k)	140-239	Level 3	4000	14,759	10,000	85
Neoplasms (L3-8k)	140-239	Level 3	8000	17,892	10,000	88
Neoplasms (L3-16k)	140-239	Level 3	16,000	18,740	10,000	89
Injury & Poisoning (L3-1k)	800-999	Level 3	1000	9364	9364	121
Injury & Poisoning (L3-2k)	800-999	Level 3	2000	11,801	10,000	136
Injury & Poisoning (L3-4k)	800-999	Level 3	4000	14,581	10,000	156
Injury & Poisoning (L3-8k)	800-999	Level 3	8000	17,653	10,000	166
Injury & Poisoning (L3-16k)	800-999	Level 3	16,000	21,181	10,000	174
Supplementary (L3-1k)	E & V	Level 3	1000	9224	9,224	113
Supplementary (L3-2k)	E & V	Level 3	2000	11,585	10,000	130
Supplementary (L3-4k)	E & V	Level 3	4000	14,529	10,000	156
Supplementary (L3-8k)	E & V	Level 3	8000	17,790	10,000	167
Supplementary (L3-16k)	E & V	Level 3	16,000	21,321	10,000	192
All (L2-1k)	All	Level 2	1000	8999	8999	141
All (L2-2k)	All	Level 2	2000	11,453	10,000	146
All (L2-4k)	All	Level 2	4000	14,295	10,000	152
All (L2-8k)	All	Level 2	8000	17,484	10,000	158
All (L2-16k)	All	Level 2	16,000	21,030	10,000	161

labels. Due to the increasing size of datasets, it was essential to employ high-performance computing resources. To ensure the efficient execution of the AutoML models, Tinaroo [95], a high-performance computer was used, provided by the University of Queensland.

Tinaroo consists of 244 compute nodes, each of which includes 2 Intel Xeon 12-core CPUs, 128 GB RAM, and 1 TB disk space, resulting in approximately 6000 CPU cores, 30 TB RAM, and 0.5 PB disk space. The vast computing resources available on Tinaroo allowed us to efficiently process and analyze our large datasets. Tinaroo operates on CentOS Linux 7 (Core), a widely used open-source operating system in the scientific computing community. The system's stability and reliability ensured that the execution of the AutoML models was consistent across all experiments.

To ensure consistency across all machine learning techniques and models used in the experiments, The models were executed on a single node equipped with a maximum of 24 core CPU and 120 GB RAM. This approach ensured that each model received the same amount of computing resources and eliminated any variations that could arise from executing the models on different nodes.

5.3 Model preprocessing

The proposed model in this paper is designed to address the challenges of medical data classification by using a combination of preprocessing, clustering, grouping, and classification techniques. The model starts with the preprocessing of medical data using cTAKES to extract meaningful features from the data.

Following preprocessing, as demonstrated in Fig. 4, the model applies clustering and grouping techniques to group the extracted features into related clusters. This grouping helps to reduce the dimensionality of the data and identify patterns that can be used for classification. A single representative label is then identified for each group, and labels that fall outside the selected threshold are eliminated from the clustering process.

After grouping and labeling, the model, as shown in Fig. 6 applies classification techniques to each selected label using the Auto-Sklearn library. The best model that provides the highest F1-score is then identified for each label. In certain instances, the Auto-Sklearn library is unable to identify a model that outperforms a dummy model [96]. In such cases, the proposed model utilizes the Random Forest algorithm. This algorithm is chosen due to its documented superior performance in previous studies [15, 37]. This

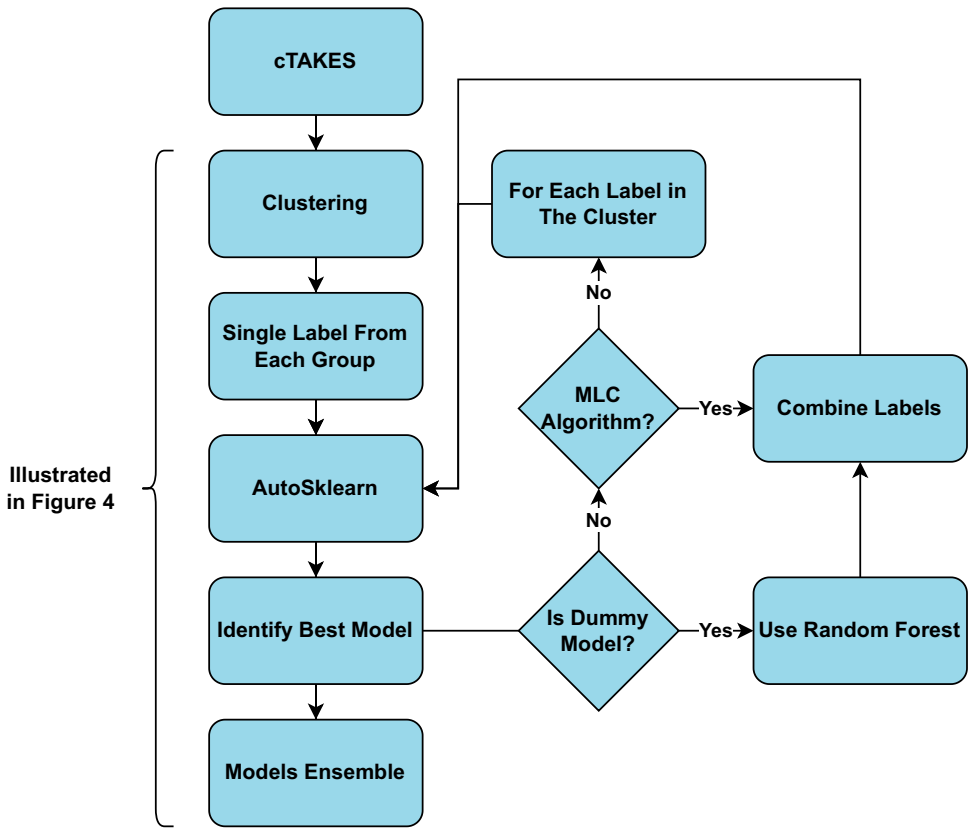

Fig. 6 Clustered automated machine learning model

step is crucial in achieving high classification accuracy and reducing computational time.

If the best model supports multi-label classification, all labels within the cluster are directly classified using that algorithm. If not, the model loops through each label in the cluster. Finally, all selected models from the aforementioned steps are combined into a model ensemble to achieve the highest possible accuracy.

The initial step in the conversion of unstructured medical reports involves the utilization of cTAKES. Section 4.1.1 elucidates that medical reports are transformed into a CUI format, and subsequently, the BOW method is employed to quantify these features. Meanwhile, the list of diagnoses is employed to create a label set using the binary relevance method, as outlined in Sect. 4.1.2. These preprocessing steps are common to both the clustered AutoML model and the native Auto-Sklearn model. The subsequent stage involves label clustering and grouping, as explicated in Sect. 4.2. The duration of this process and the number of clusters generated vary, depending on the size of the dataset. Table 4 presents the time spent on clustering and grouping for each dataset, alongside the number of clusters generated for each. Notably, the lengthiest duration spent on this process was less than 2.5 min, and the number of clusters ranged between one and ten. In more detail, the Neoplasms dataset labels were consolidated into a single cluster, whereas the Injury and Poisoning datasets were grouped into two to ten clusters. Supplementary datasets were grouped into two to six clusters, and All Level 2 datasets were grouped into five to eight clusters. The time required for clustering is directly linked to the dataset's size, ranging from 2 s for a Neoplasms Level 3 dataset with 1000 records to 141 s for an Injury & Poisoning Level 3 dataset with 16,000 records.

Table 4 Clustering time and number of clusters

Dataset name	Clustering time (Sec)	# Clusters
Neoplasms (L3-1k)	2	1
Neoplasms (L3-2k)	4	1
Neoplasms (L3-4k)	8	1
Neoplasms (L3-8k)	18	1
Neoplasms (L3-16k)	21	1
Injury & Poisoning (L3-1k)	3	10
Injury & Poisoning (L3-2k)	11	7
Injury & Poisoning (L3-4k)	31	4
Injury & Poisoning (L3-8k)	66	3
Injury & Poisoning (L3-16k)	141	2
Supplementary (L3-1k)	4	6
Supplementary (L3-2k)	10	4
Supplementary (L3-4k)	31	2
Supplementary (L3-8k)	65	3
Supplementary (L3-16k)	104	3
All (L2-1k)	7	6
All (L2-2k)	11	8
All (L2-4k)	28	6
All (L2-8k)	36	5
All (L2-16k)	77	7

5.4 Configurations

The present model relies solely on the threshold parameter, which serves as the key factor affecting its performance. Altering this parameter creates a trade-off between accuracy and performance, with higher thresholds leading to superior performance but lower accuracy and lower thresholds yielding inferior performance but higher accuracy. In the context of this model, the optimal threshold was determined to be 15% since it enabled high accuracy within a reasonable timeframe. The remaining parameters, while still relevant to the model, are not directly associated with it and instead pertain to the Auto-Sklearn library [97]. In this study, two Auto-Sklearn parameters were modified, namely, "time_left_for_this_task," which represents the time limit in seconds for Auto-Sklearn to identify the best model, and "per_run_time_limit," which indicates the time limit for each algorithm run in seconds. The values of these parameters were varied within the set 120, 240, 360 for time_left_for_this_task, while per_run_time_limit was restricted to 60 s due to limited resources. Given the large number of features and labels, these adjustments were deemed essential for ensuring optimal performance.

5.5 Comparison

The performance of the CAML model was compared with that of the Auto-Sklearn model for the task of multi-label classification, utilizing identical datasets (Table 3). Both models were allocated equal running time; however, it was observed that the Auto-Sklearn model used slightly more runtime than the CAML model. This discrepancy arises from the initial consideration of the time required for the CAML model to complete each experiment. Subsequently, the time_left_for_this_task parameter in Auto-Sklearn was set to the same duration. However, it should be noted that Auto-Sklearn does not consistently terminate precisely at the designated time_left_for_this_task limit; often, it exceeds this predetermined duration. As a result, the elapsed time for each experiment in Auto-Sklearn is slightly longer compared to the time consumed by CAML for the corresponding experiment.

In our experiments, a comparison of two different types of F1-scores was conducted: micro, and weighted F1-scores. The micro F1-score was computed by treating all discharge summaries and diagnoses in the dataset equally. This approach aimed to assess the overall performance of the models by assigning equal weight to each discharge summary and diagnosis. The weighted F1-score computed the F1-score for each diagnosis separately. However, it also took into consideration the number of discharge summaries associated with each diagnosis. In other words, it considered the dataset's imbalance and assigned a higher weight to diagnoses that appeared more frequently. This approach aimed to provide a better representation of each experiment's performance by accounting for the varying instances of diagnoses within the dataset.

5.6 Results

Both CAML and Auto-Sklearn were utilized to classify multi-label datasets, with Neoplasms (L3-1k) serving as the first dataset used (row 1 of Table 3). For the CAML model, the Neoplasms (L3-1k) dataset was configured for a time limit of 120 s (2 min) for time_left_for_this_task and a per_run_time_limit of 60 s (1 min), with a 15% model threshold. The 15% threshold was subsequently employed for the remaining datasets. The labels were categorized into a single group, and all distances, except for two labels, fell within the 15% threshold. The Secondary malignant neoplasm of respiratory and digestive systems (ICD: 197) and Secondary malignant neoplasm of other specified sites (ICD: 198) was 16.1% and 18.3% distant, respectively, from the closest label. Random Forest was the algorithm that generated the clustered labels list and served as the leader of the ICD 197 and ICD 198 labels model. The entire process took 6.5 min, with a 56.52% Micro F1-score (see row 1 of Table 6).

To compare the efficacy of the CAML model against the Auto-Sklearn model in classifying multi-label datasets with similar parameters, the time_left_for_this_task parameter was set to 8 min, and the per_run_time_limit was set to 4 min, reflecting a 2 : 1 ratio similar to the CAML model. The Auto-Sklearn model required a total of 8.2 min and produced a Micro F1-score of 38.33%, with BernoulliNB emerging as the most effective algorithm in generating these results (see row 1 of Table 10).

For the L3-2k dataset, AdaBoost was identified for both of the out-of-the-threshold labels and provided a 55.63% Micro F1-score in 6.5 min using the CAML model. In contrast, Auto-Sklearn using the MLP algorithm achieved a lower Micro F1-score of 36.46% in 8.2 min (see row 2 of Tables 6 and 6).

Similarly, the other dataset size comparisons are shown in Tables 6 and 6 for a ratio of 2:1. When the dataset size increased to 4000 records (L3-4k), 3 Random Forest algorithms took 6.9 min to produce a higher Micro F1-score of 60.66%. Auto-Sklearn using the Decision Tree algorithm achieved a lower Micro F1-score of 42.62%.

CAML model on Neoplasms (L3-8k) dataset demonstrated a slight reduction in the processing time of 6.4 min and achieved a Micro F1-score of 52.32% when utilizing 2 BernoulliNB and Decision Tree algorithms. In comparison, Auto-Sklearn yielded a Micro F1-score of 46.56% using the Decision Tree algorithm and required 8.7 min to complete processing.

When considering the Neoplasms (L3-16k) dataset, CAML achieved a Micro F1-score of 60.44% in 6.3 min by employing the Linear Support Vector Classifier (SVC) and BernoulliNB algorithms, while Auto-Sklearn yielded a Micro F1-score of 49.67% utilizing the Decision Tree algorithm, with a processing time of 10.4 min. It is noteworthy that in all datasets examined, CAML exhibited superior performance in terms of F1-score when compared to the Auto-Sklearn model.

The same experiments as explained above, were also conducted to investigate the efficacy of the CAML and Auto-Sklearn models on Injury & Poisoning, Supplementary, and All Level 2 datasets (see Table 3). These experiments were iterated only altering the initial 2:1 timing configurations, as 4:1 (time_left_for_this_task=4 min, and per_run_time_limit=1 min), 4:2 (time_left_for_this_task=4 min, and per_run_time_limit=2 min), and 6:1 (time_left_for_this_task=6 min, and per_run_time_limit=1 min). Tables 6 to 13 present the outcomes of these experiments.

Overall, these Tables show 73 experimental results across 20 different datasets of varying sizes and configurations, as shown in Table 3. It is worth noting that, out of the 80 possible experiments (shown in Tables 6 to 13 for the different dataset sizes and ratio settings), Seven experiments involving extended timing configurations could not be executed within the established experimental environment, as detailed in section 5.2. However, the experimental environment was not altered to maintain consistency in comparisons. Therefore, no results are shown for these experiments.

Both CAML and Auto-Sklearn were evaluated across the remaining 73 datasets. Results indicated that CAML outperformed Auto-Sklearn in the majority of cases, with 66 out of 73 cases demonstrating superior performance. This represents a success rate of over 90%. Furthermore, CAML consistently demonstrated higher F1-scores in all dataset sizes and configurations compared to Auto-Sklearn. This is evidenced by the Micro F1-score improvement ratio depicted in Fig. 7, which demonstrates a significant improvement in performance achieved by CAML compared to Auto-Sklearn. For datasets consisting of 1000 records, an average Micro-F1 improvement ratio of 23% was observed. This improvement

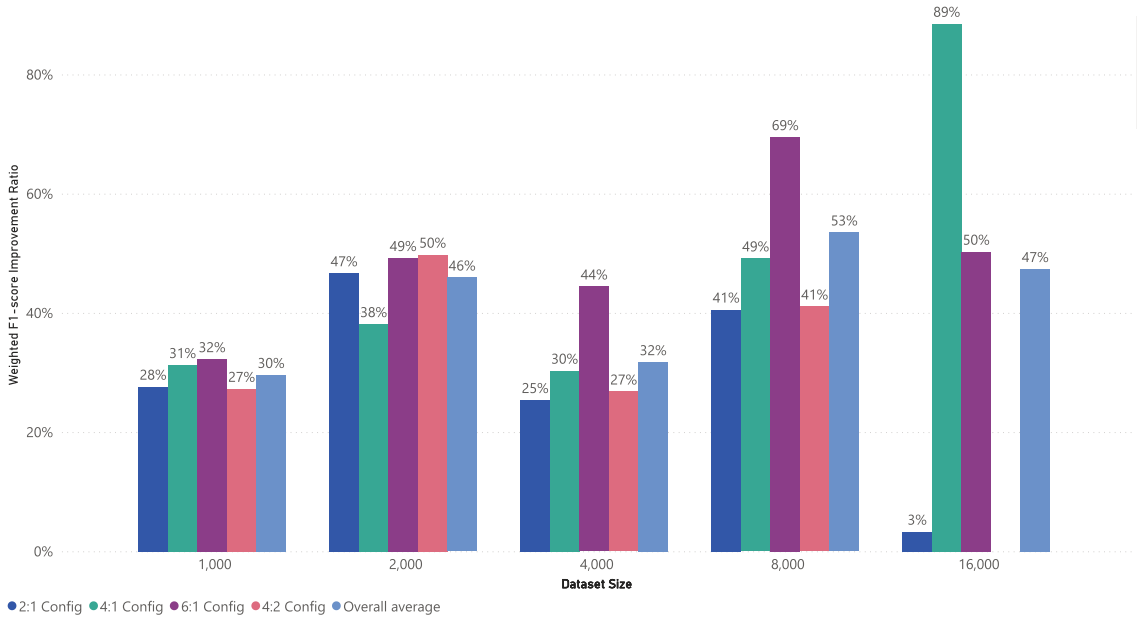

Fig. 7 CAML micro F1-score improvement ratio in comparison to Auto-Sklearn

is calculated by taking the average of the improvements in all five 1K dataset cases shown in Tables 6, 7, 8, 9, 10, 11, 12 and 13. The improvement in each of the five cases is calculated by dividing the difference in the F1-Score of CAML and Auto-Sklearn by the Auto-Sklearn F1-score value (i.e. the base value) for that case. For instance, the Micro F1-Score improved from 38.33% using Auto-Sklearn to 56.52% when utilizing CAML on the Neoplasms 1K dataset, achieving 47.46% improvement in this case. Similarly, for the Injury & Poisoning 1K dataset, an improvement ratio of 10.83% was observed, while the Supplementary 1K dataset showed an improvement ratio of 11.24%. When considering all level 2 1K datasets, the improvement ratio across the board amounts to 24.09%. Evaluating the average improvement ratio for the 2:1 configuration and a dataset size of 1K records then yields a value of 23.40%, which is shown as 23% in the most left bar of Fig. 7. All other improvement values shown in this figure are calculated similarly. Overall, comparing CAML to Auto-Sklearn across all datasets of various sizes reveals an average F1-Score improvement ratio of 35.15% for CAML.

The results presented in Tables 6, 7, 8, 9, 10, 11, 12 and 13 demonstrate instances where the CAML model performed less effectively compared to the Auto-Sklearn model in seven specific cases. Among these cases, two involved datasets on Injury & Poisoning with 8000 and 16,000 records, respectively, using a 2:1 configuration. In these instances, Auto-Sklearn exhibited significantly superior Micro F1-score results of 37.89% and 35.38% as opposed to CAML's results of 27.22% and 22.19%. Furthermore, the Supplementary datasets with 4000, 8000, and 16,000 records, employing a 2:1 configuration, exhibited three cases where Auto-Sklearn outperformed CAML. The Micro F1-score results for Auto-Sklearn were significantly higher at 33.20%, 33.91%, and 22.12% compared to CAML's results of 28.20%, 27.20%, and 13.56%, respectively. Additionally, there was one case involving 8000 records in the Injury & Poisoning datasets with a 4:1 configuration, where Auto-Sklearn displayed slightly better Micro F1-score of 47.79% in contrast to CAML's 46.81%. Lastly, for the 4000 records in the Supplementary datasets using a 4:2 configuration, Auto-Sklearn showcased marginally superior Micro F1-score results of 33.31% compared to CAML's results of 32.96%. The reason for these findings can be attributed to providing Auto-Sklearn with extended time to explore a wider range of algorithms, particularly considering the substantial availability of algorithms that facilitate single-label classification.

When considering the Weighted F1-score, compared to the micro F1-score, our performance analysis still demonstrated the superiority of CAML over Auto-Sklearn. Specifically, CAML achieved a higher Weighted F1-score in 63 cases out of the overall 73 experiments. The weighted F1-score improvement ratios for the various experimental dataset sizes and configurations are shown in Fig. 8. These values, which were calculated in a similar manner to those in

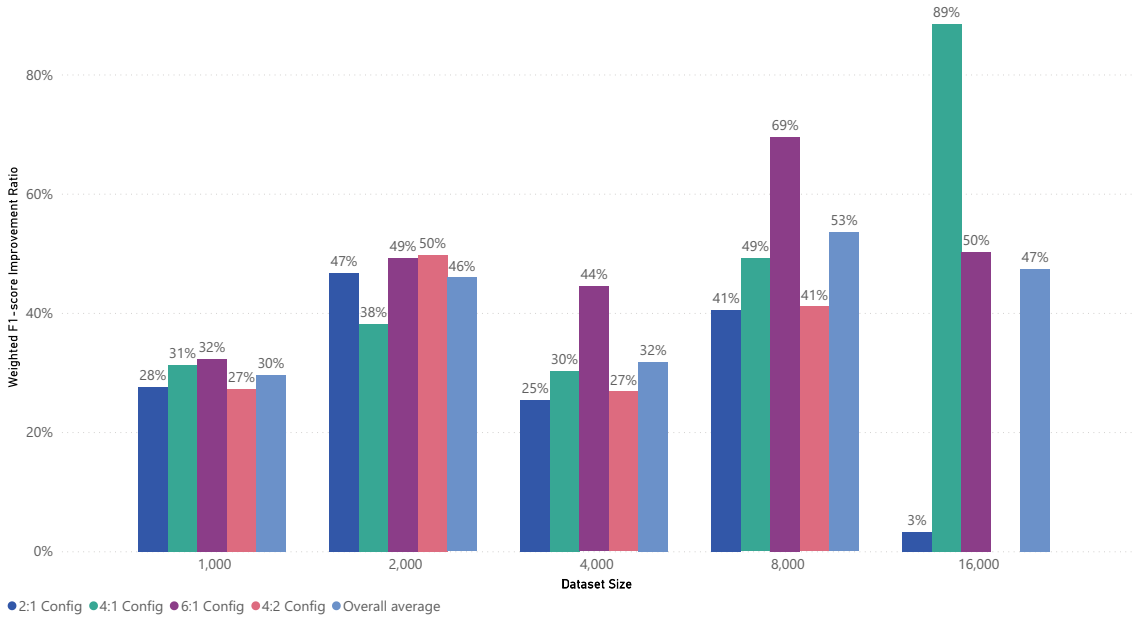

Fig. 8 CAML weighted F1-score improvement ratio in comparison to Auto-Sklearn

Table 5 CAML ensemble algorithms

Algorithm	Appearance count
Random forest (RF)	67
Ada boost (Ada)	27
Decision tree (DT)	27
Extra trees (ET)	26
Gradient boosting (GB)	21
Passive aggressive (PA)	18
Library for linear SVC (LSVC)	14
Bernoulli Naive Bayes (BNB)	11
Stochastic Gradient Descent (SGD)	11
K Nearest Neighbors (KNN)	9
Multi-layer Perceptron (MLP)	6
Library for SVC (LSVMC)	5
Gaussian Naive Bayes (GNB)	3
Linear Discriminant Analysis (LDA)	1

Fig. 7, result in an average weighted F1-score improvement ratio of 40.56%.

To gain a comprehensive understanding of the algorithms employed by CAML and Auto-Sklearn models, a thorough investigation was conducted. Random Forest (RF) is the most frequently utilized algorithm in CAML, appearing in 67 CAML ensembles, followed by Ada Boost (Ada) and Decision Tree (DT), each appearing in 27 ensembles. Extra Trees (ET) is also utilized frequently, appearing in 26 ensembles. Table 5 provides a breakdown of the frequency of algorithm utilization in these ensembles, with a total of 14 distinct algorithms employed. Conversely, Auto-Sklearn models employ only three algorithms, with Bernoulli Naive Bayes (BNB) appearing 40 times, Decision Tree (DT) appearing 22 times, and Multi-layer Perceptron (MLP) appearing 11 times across our experiments.

The superior performance of CAML models over Auto-Sklearn models can be attributed, in part, to the diversity of algorithms utilized in their construction. While Auto-Sklearn exclusively employs algorithms that natively support multi-label classification, CAML models are able to utilize any classification algorithm, thus increasing the likelihood of identifying an algorithm that yields a higher F1-score. Ensembles, which consist of a combination of multiple algorithms ranging from a single algorithm to as many as nine, are commonly utilized in CAML models and can be another reason for CAML's improved performance.

The F1 score exhibits a consistently low performance across various tested datasets. This issue arises from the inherent difficulty associated with handling a large set of labels in Multi-Label Classification models. The inclusion of a large label set significantly amplifies the model complexity, subsequently diminishing its accuracy [98, 99]. Many prior research endeavors have sought to address this complexity by constraining the clinical coding classification to

a more manageable scenario involving a single label, often limited to a specific set [26, 27, 33]. While this strategy has proven to enhance accuracy in those studies, it comes at the cost of overlooking the intricacies and nuances present in datasets with a broader range of labels. In contrast to these limitations, our model operates with a more expansive label set, spanning from 72 to 192 labels, encompassing diverse codes. Furthermore, the nature of multi-label datasets introduces an additional layer of complexity through the inherent imbalance among labels. This imbalance is evident as certain labels are more prevalent than others, with instances where one label may be more frequently observed than its counterparts [100].

In summary, the study employed both CAML and Auto-Sklearn to classify multi-label datasets, starting with the Neoplasms (L3-1k) dataset. CAML demonstrated superior performance, achieving a Micro F1-score of 56.52% in 6.5 min, compared to Auto-Sklearn's 38.33% in 8.2 min. This trend continued across various dataset sizes and configurations, with CAML consistently outperforming Auto-Sklearn in 66 out of 73 cases, representing a success rate of over 90%. Notably, CAML exhibited higher F1-scores across all dataset sizes, with an average improvement ratio of 35.15% compared to Auto-Sklearn. Despite a few cases where Auto-Sklearn performed better, the study attributes CAML's superior performance to its diverse use of algorithms in constructing ensembles, enabling a broader exploration of classification algorithms. Overall, CAML's versatility and ability to create ensembles contribute to its improved performance in multi-label classification tasks.

6 Discussion and conclusion

The process of clinical coding is a labor-intensive endeavor, requiring coders to manually extract and categorize patients' diseases. In an effort to enhance this process, various algorithms have been tested to assist clinical coders and medical practitioners. However, the majority of these algorithms are designed for the classification of individual diseases, and those capable of accommodating multi-label classification lack full automation.

In this research study, an innovative approach was presented to automate the clinical coding procedure utilizing the AutoML library Auto-Sklearn. A dataset comprising approximately 64,000 discharge summary reports from the MIMIC III database was subjected to cTAKES to extract CUI codes. BOW was employed for the quantification of CUI codes and for the development of the feature-set. Given the existence of more than 28,000 distinct CUIs, the analysis was limited to the top 10,000 CUIs to maintain a manageable feature-set. The dataset's other component was the creation of the label-set, derived from the MIMIC III dataset which contained multiple diagnoses utilizing ICD9 codes, and these were classified into two hierarchical levels, specifically Level 2 and Level 3. Level 3 codes were utilized for the Neoplasms dataset, Injury & Poisoning dataset, and Injury & Poisoning dataset, while Level 2 was applied across all records.

The computations for this study were executed on Tinaroo, a high-performance computing system boasting 12-core CPUs and 128 GB of RAM. In total, 73 distinct models were run, and these models facilitated a comprehensive comparison between the CAML model and Auto-Sklearn. The CAML model engages in multi-label classification by organizing these labels into separate groups based on their label-to-label distances. Auto-Sklearn is then employed for each of these label groups as a single-label classification dataset. The selected algorithms collaboratively form an ensemble of methods aimed at delivering the utmost precision in terms of F1-scores accuracy. In our comparative analysis, CAML consistently outperformed Auto-Sklearn in the majority of experiments, yielding superior Micro and Weighted F1-scores, with an average improvement ratio of 35.15% and 40.56%, respectively.

Our proposed methodology can be integrated within the Auto-Sklearn framework to enhance its proficiency in multi-label classification tasks. Additionally, the CAML model demonstrates potential for extension beyond the confines of Auto-Sklearn, offering integration with other AutoML libraries. This expansion could grant access to a broader array of algorithms, particularly those harnessing neural networks such as CNNs, RNNs, and LSTM networks. Such an extension holds promise for even more precise and efficient disease identification in the clinical coding domain, representing a focal point of our forthcoming research endeavors.

7 Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the author(s) used ChatGPT in order to improve language and readability. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

Appendix A: Experiments results

See Tables 6, 7, 8, 9, 10, 11, 12 and 13.

Table 6 CAML results for 2:1 configurations

Dataset name	Process time (min)	Micro F1	Weighted F1	Ensemble algorithms
Neoplasms (L3-1k)	6.5	56.52	54.31	RF
Neoplasms (L3-2k)	6.5	55.63	55.05	PA, Ada
Neoplasms (L3-4k)	6.9	60.66	60.30	RF
Neoplasms (L3-8k)	6.4	52.32	51.68	BNB, DT
Neoplasms (L3-16k)	6.3	60.44	60.41	BNB, LSVC
Injury & Poisoning (L3-1k)	32.6	43.53	40.53	GNB, GB, Ada, RF
Injury & Poisoning (L3-2k)	25.2	43.08	38.75	RF, MLP, GB, ET, Ada
Injury & Poisoning (L3-4k)	17.2	44.63	39.44	RF, MLP
Injury & Poisoning (L3-8k)	15.3	27.22	20.49	RF, BNB, ET
Injury & Poisoning (L3-16k)	12.5	22.19	15.15	RF, GNB
Supplementary (L3-1k)	17.6	34.95	30.84	GB, Ada, RF, ET
Supplementary (L3-2k)	15.8	40.16	34.90	KNN, RF, ET
Supplementary (L3-4k)	10.8	28.20	25.32	ET, RF
Supplementary (L3-8k)	12.9	27.20	18.91	RF, LSVC
Supplementary (L3-16k)	13.0	13.56	6.72	RF, DT
All (L2-1k)	50.2	57.92	52.51	GB, RF, KNN, Ada, PA
All (L2-2k)	55.3	60.35	55.39	Ada, RF, ET
All (L2-4k)	51.7	61.34	56.25	GB, DT, RF
All (L2-8k)	49.8	56.26	46.37	RF, DT, BNB
All (L2-16k)	52.7	48.13	37.82	BNB, RF, DT, GNB, PA

Table 7 CAML results for 4:1 configurations

Dataset name	Process time (min)	Micro F1	Weighted F1	Ensemble algorithms
Neoplasms (L3-1k)	12.2	57.75	56.43	Ada, RF
Neoplasms (L3-2k)	12.3	54.55	53.66	ET, LSVC, RF
Neoplasms (L3-4k)	13.3	62.09	61.93	RF
Neoplasms (L3-8k)	12.6	65.96	65.85	DT, BNB
Neoplasms (L3-16k)	12.4	65.66	65.30	LSVC, RF
Injury & Poisoning (L3-1k)	59.8	42.21	38.17	Ada, GB, ET, RF
Injury & Poisoning (L3-2k)	45.9	42.33	39.13	RF, MLP
Injury & Poisoning (L3-4k)	33.2	45.87	41.98	RF, DT, Ada, LSVC
Injury & Poisoning (L3-8k)	29.3	46.81	41.14	RF, PA, ET, BNB
Injury & Poisoning (L3-16k)	25.1	49.03	45.10	DT, RF, PA, LSVC
Supplementary (L3-1k)	33.4	42.19	38.20	GB, Ada, RF, DT
Supplementary (L3-2k)	29.0	39.43	34.69	KNN, ET, RF, LSVM
Supplementary (L3-4k)	21.5	34.09	31.96	RF, ET
Supplementary (L3-8k)	25.3	32.38	27.75	RF, DT, PA
Supplementary (L3-16k)	25.1	43.62	35.05	RF, PA
All (L2-1k)	95.9	58.56	53.65	ET, Ada, GB, PA, RF, ET, LSVM, KNN
All (L2-2k)	105.6	61.11	56.64	RF, DT, Ada, ET
All (L2-4k)	100.4	62.67	58.24	RF, GB, DT
All (L2-8k)	96.7	64.42	60.20	DT, PA, RF, SGD, GB, KNN, BNB, LSVC
All (L2-16k)	102.8	61.20	54.31	RF, DT, PA, LSVC, SGD, BNB

Table 8 CAML results for 6:1 configurations

Dataset name	Process time (min)	Micro F1	Weighted F1	Ensemble algorithms
Neoplasms (L3-1k)	18.3	55.88	53.16	Ada, SGD
Neoplasms (L3-2k)	18.5	61.25	61.03	RF
Neoplasms (L3-4k)	18.6	65.27	64.78	RF
Neoplasms (L3-8k)	19.6	67.63	67.74	PA, RF, SGD
Neoplasms (L3-16k)	18.8	66.14	65.84	DT, LSVC, SGD
Injury & Poisoning (L3-1k)	86.5	42.07	38.27	RF, DT, GB, BNB, Ada
Injury & Poisoning (L3-2k)	68.5	45.44	41.21	RF, MLP
Injury & Poisoning (L3-4k)	50.8	49.14	46.06	RF, DT
Injury & Poisoning (L3-8k)	43.0	53.61	51.63	RF, PA, GB, LSVC
Injury & Poisoning (L3-16k)	36.6	44.22	37.46	DT, RF, SGD
Supplementary (L3-1k)	49.4	45.19	41.19	RF, SGD, Ada, GB
Supplementary (L3-2k)	44.3	41.80	36.83	RF, ET, LSVM, Ada
Supplementary (L3-4k)	31.6	40.05	37.41	RF, LSVC
Supplementary (L3-8k)	36.9	42.00	38.63	KNN, RF, SGD, LSVC, PA
Supplementary (L3-16k)	36.7	37.20	30.46	RF, DT, PA
All (L2-1k)	142.7	59.28	55.38	GB, LDA, DT, PA, RF, ET, Ada, BNB, KNN
All (L2-2k)	155.3	60.95	56.61	RF, DT, GB, ET, Ada
All (L2-4k)	148.2	63.29	59.32	RF, GB, DT, Ada
All (L2-8k)	143.7	64.72	60.83	MLP, RF, ET, DT, GB, Ada, SGD, PA
All (L2-16k)	153.2	64.49	60.14	DT, RF, Ada, SGD, LSVC, PA

Table 9 CAML results for 4:2 configurations

Dataset name	Process time (min)	Micro F1	Weighted F1	Ensemble algorithms
Neoplasms (L3-1k)	12.9	54.81	54.06	Ada, RF
Neoplasms (L3-2k)	13.0	60.53	59.98	ET, LSVC, RF
Neoplasms (L3-4k)	13.2	66.32	65.99	RF
Neoplasms (L3-8k)	12.5	65.69	65.77	RF
Neoplasms (L3-16k)	–	–	–	–
Injury & Poisoning (L3-1k)	58.8	42.31	37.92	Ada, GB, ET, RF
Injury & Poisoning (L3-2k)	46.5	46.13	42.50	RF, MLP
Injury & Poisoning (L3-4k)	34.2	48.18	44.83	DT, RF, ET
Injury & Poisoning (L3-8k)	–	–	–	–
Injury & Poisoning (L3-16k)	–	–	–	–
Supplementary (L3-1k)	33.8	40.00	36.06	GB, Ada, RF, DT
Supplementary (L3-2k)	30.0	42.02	37.21	KNN, ET, RF, LSVM
Supplementary (L3-4k)	21.9	32.96	29.88	RF, ET
Supplementary (L3-8k)	–	–	–	–
Supplementary (L3-16k)	–	–	–	–
All (L2-1k)	97.1	58.53	52.85	ET, Ada, GB, PA, RF, LSVM, KNN
All (L2-2k)	104.3	61.19	57.04	RF, Ada, DT, ET
All (L2-4k)	100.1	64.40	60.65	SGD, Ada, RF, GB, ET
All (L2-8k)	–	–	–	–
All (L2-16k)	–	–	–	–

Table 10 Auto-Sklearn results for 2:1 configurations

Dataset name	Process time (min)	Micro F1	Weighted F1	Top algorithm
Neoplasms (L3-1k)	8.2	38.33	32.24	BNB
Neoplasms (L3-2k)	8.2	36.46	33.30	MLP
Neoplasms (L3-4k)	8.6	42.62	38.24	DT
Neoplasms (L3-8k)	8.7	46.56	42.62	DT
Neoplasms (L3-16k)	10.4	49.67	46.24	DT
Injury & Poisoning (L3-1k)	34.2	39.28	36.05	BNB
Injury & Poisoning (L3-2k)	26.2	39.35	31.89	BNB
Injury & Poisoning (L3-4k)	18.4	43.64	40.96	BNB
Injury & Poisoning (L3-8k)	16.9	37.89	31.48	BNB
Injury & Poisoning (L3-16k)	12.3	35.38	31.56	DT
Supplementary (L3-1k)	18.4	31.42	27.02	BNB
Supplementary (L3-2k)	16.6	34.18	28.62	BNB
Supplementary (L3-4k)	12.3	33.20	30.64	BNB
Supplementary (L3-8k)	14.7	33.91	27.64	BNB
Supplementary (L3-16k)	14.4	22.12	16.47	DT
All (L2-1k)	52.2	46.68	45.49	BNB
All (L2-2k)	56.4	35.21	31.15	BNB
All (L2-4k)	54.4	34.04	34.11	BNB
All (L2-8k)	53.2	19.35	15.08	DT
All (L2-16k)	57.2	24.42	19.53	MLP

Table 11 Auto-Sklearn results for 4:1 configurations

Dataset name	Process time (min)	Micro F1	Weighted F1	Top algorithm
Neoplasms (L3-1k)	14.2	40.85	35.51	DT
Neoplasms (L3-2k)	14.4	44.81	41.71	DT
Neoplasms (L3-4k)	14.3	46.81	44.80	BNB
Neoplasms (L3-8k)	13.9	50.38	46.25	DT
Neoplasms (L3-16k)	14.5	49.51	45.95	DT
Injury & Poisoning (L3-1k)	61.1	40.34	35.64	MLP
Injury & Poisoning (L3-2k)	46.8	39.46	32.03	BNB
Injury & Poisoning (L3-4k)	34.3	43.11	38.42	BNB
Injury & Poisoning (L3-8k)	30.6	47.79	44.24	BNB
Injury & Poisoning (L3-16k)	28.5	38.69	34.66	DT
Supplementary (L3-1k)	34.1	34.45	27.05	MLP
Supplementary (L3-2k)	30.1	32.16	28.46	BNB
Supplementary (L3-4k)	22.3	33.71	31.43	BNB
Supplementary (L3-8k)	26.4	27.85	32.16	BNB
Supplementary (L3-16k)	26.4	26.92	20.19	DT
All (L2-1k)	96.4	46.78	45.55	BNB
All (L2-2k)	106.4	32.96	31.51	BNB
All (L2-4k)	102.3	35.32	33.86	BNB
All (L2-8k)	98.3	28.40 21.90	DT	
All (L2-16k)	104.5	22.67	17.62	DT

Table 12 Auto-Sklearn results for 6:1 configurations

Dataset name	Process time (min)	Micro F1	Weighted F1	Top algorithm
Neoplasms (L3-1k)	20.4	41.30	35.73	DT
Neoplasms (L3-2k)	20.2	39.52	36.02	BNB
Neoplasms (L3-4k)	20.3	47.03	43.00	DT
Neoplasms (L3-8k)	21.7	53.01	48.83	DT
Neoplasms (L3-16k)	20.2	57.15	53.34	MLP
Injury & Poisoning (L3-1k)	88.2	38.81	35.46	BNB
Injury & Poisoning (L3-2k)	69.2	40.32	33.77	MLP
Injury & Poisoning (L3-4k)	51.8	43.21	38.00	BNB
Injury & Poisoning (L3-8k)	45.4	39.49	38.94	BNB
Injury & Poisoning (L3-16k)	38.4	41.79	37.96	DT
Supplementary (L3-1k)	51.1	31.59	27.39	BNB
Supplementary (L3-2k)	45.2	32.54	28.52	BNB
Supplementary (L3-4k)	33.3	34.65	31.40	BNB
Supplementary (L3-8k)	38.3	33.39	30.46	BNB
Supplementary (L3-16k)	39.4	29.60	22.54	DT
All (L2-1k)	144.2	46.65	45.46	BNB
All (L2-2k)	156.3	37.41	32.12	MLP
All (L2-4k)	150.5	35.94	31.77	BNB
All (L2-8k)	145.7	29.04	21.74	MLP
All (L2-16k)	155.8	30.51	24.66	DT

Table 13 Auto-Sklearn results for 4:2 configurations

Dataset name	Process time (min)	Micro F1	Weighted F1	Top algorithm
Neoplasms (L3-1k)	14.2	40.56	35.27	DT
Neoplasms (L3-2k)	14.3	42.96	38.65	MLP
Neoplasms (L3-4k)	14.8	55.25	52.54	MLP
Neoplasms (L3-8k)	14.1	51.03	46.62	DT
Neoplasms (L3-16k)	–	–	–	–
Injury & Poisoning (L3-1k)	60.2	40.85	35.07	MLP
Injury & Poisoning (L3-2k)	48.4	39.42	32.06	BNB
Injury & Poisoning (L3-4k)	36.3	43.01	40.70	BNB
Injury & Poisoning (L3-8k)	–	–	–	–
Injury & Poisoning (L3-16k)	–	–	–	–
Supplementary (L3-1k)	34.9	31.93	27.41	BNB
Supplementary (L3-2k)	32.6	31.92	28.19	BNB
Supplementary (L3-4k)	22.4	33.31	31.01	BNB
Supplementary (L3-8k)	–	–	–	–
Supplementary (L3-16k)	–	–	–	–
All (L2-1k)	98.3	46.70	45.54	BNB
All (L2-2k)	106.2	33.25	31.81	BNB
All (L2-4k)	102.3	33.34	34.64	BNB
All (L2-8k)	–	–	–	–
All (L2-16k)	–	–	–	–

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Data availability The data used for this work is available upon reasonable request.

References

1. Huang C, Wang J, Wang S, Zhang Y (2023) A review of deep learning in dentistry. Neurocomputing 126629
2. Huang C, Murugiah K, Mahajan S, Li S-X, Dhruva SS, Haimovich JS, Wang Y, Schulz WL, Testani JM, Wilson FP et al (2018) Enhancing the prediction of acute kidney injury risk after percutaneous coronary intervention using machine learning techniques: A retrospective cohort study. PLoS Med 15(11):1002703
3. Arora G, Joshi J, Mandal RS, Shrivastava N, Virmani R, Sethi T (2021) Artificial intelligence in surveillance, diagnosis, drug discovery and vaccine development against COVID-19. Pathogens 10(8):1048
4. Awotunde JB, Oluwabukonla S, Chakraborty C, Bhoi AK, Ajamu GJ (2022) Application of artificial intelligence and big data for fighting COVID-19 pandemic. Decision Sciences for COVID-19: Learning Through Case Studies 3–26
5. Azghadi MR, Lammie C, Eshraghian JK, Payvand M, Donati E, Linares-Barranco B, Indiveri G (2020) Hardware implementation of deep network accelerators towards healthcare and biomedical applications. IEEE Trans Biomed Circuits Syst 14(6):1138–1159
6. Eysenbach G et al (2023) The role of chatgpt, generative language models, and artificial intelligence in medical education: a conversation with chatgpt and a call for papers. JMIR Med Educ 9(1):46885
7. Scherr R, Halaseh FF, Spina A, Andalib S, Rivera R (2023) Chatgpt interactive medical simulations for early clinical education: case study. JMIR Med Educ 9:49877
8. Cheong RCT, Pang KP, Unadkat S, Mcneillis V, Williamson A, Joseph J, Randhawa P, Andrews P, Paleri V (2023) Performance of artificial intelligence chatbots in sleep medicine certification board exams: Chatgpt versus google bard. Eur Arch Oto-Rhino-Laryngol 1–7
9. King DR, Nanda G, Stoddard J, Dempsey A, Hergert S, Shore JH, Torous J (2023) An introduction to generative artificial intelligence in mental health care: considerations and guidance. Curr Psychiatry Reports 1–8
10. Campbell S, Giadresco K (2020) Computer-assisted clinical coding: a narrative review of the literature on its benefits, limitations, implementation and impact on clinical coding professionals. Health Inf Manage J 49(1):5–18
11. Stanfill MH, Williams M, Fenton SH, Jenders RA, Hersh WR (2010) A systematic literature review of automated clinical coding and classification systems. J Am Med Inform Assoc 17(6):646–651
12. Erickson N, Mueller J, Shirkov A, Zhang H, Larroy P, Li M, Smola A (2020) Autogluon-tabular: robust and accurate automl for structured data. arXiv preprint arXiv:2003.06505
13. asmgx: Can Autosklearn handle multi-class/multi-label classification and which classifiers will it use? (2022). https://github.com/automl/auto-sklearn/issues/1429 Accessed 2023-02-22
14. Sheikhalishahi S, Miotto R, Dudley JT, Lavelli A, Rinaldi F, Osmani V et al (2019) Natural language processing of clinical notes on chronic diseases: systematic review. JMIR Med Inform 7(2):12239
15. Obeid JS, Weeda ER, Matuskowitz AJ, Gagnon K, Crawford T, Carr CM, Frey LJ (2019) Automated detection of altered mental status in emergency department clinical notes: a deep learning approach. BMC Med Inform Decis Mak 19:1–9
16. Yogarajan V, Montiel J, Smith T, Pfahringer B (2020) Seeing the whole patient: using multi-label medical text classification techniques to enhance predictions of medical codes. arXiv preprint arXiv:2004.00430
17. Savova GK, Masanz JJ, Ogren PV, Zheng J, Sohn S, Kipper-Schuler KC, Chute CG (2010) Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. J Am Med Inform Assoc 17(5):507–513
18. Kong H-J (2019) Managing unstructured big data in healthcare system. Healthcare Inform Res 25(1):1–2
19. Shah V, Goswami R, Kumar V, Shah B, Shah H (2018) Automated clinical documentation improvement. In: 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp 1544–1547. IEEE
20. Giannangelo K, Fenton S (2008) Ehr's effect on the revenue cycle management coding function. J Healthc Inf Manag 22(1):26–29
21. 3M™ 360 Encompass™ System — 3M Health Information Systems. MMM-ext (2011). https://www.3m.com/3M/en_US/health-information-systems-us/improve-revenue-cycle/360-encompass-system/
22. DeepMed: DeepMed [+], Code Doctor (2023). https://deepmed.com.au/codedoc.aspx Accessed 2023-02-20
23. Servais C (1992) Computer assisted coding quality management. J AHIMA 63(1):42–49
24. Benson LO, Kuelbs E, Marc L, Lock C (1996) Implementing and evaluating computer-assisted coding of adverse events. Drug Inf J 30(3):799–809
25. Terry K (2010) Is computer-assisted coding ready for inpatient use? Early indications are that it can improve productivity without sacrificing accuracy. Healthcare Inform 27(7):22–24
26. Gehrmann S, Dernoncourt F, Li Y, Carlson ET, Wu JT, Welt J, Foote Jr J, Moseley ET, Grant DW, Tyler PD et al (2017) Comparing rule-based and deep learning models for patient phenotyping. arXiv preprint arXiv:1703.08705
27. Garla V, Re VL III, Dorey-Stein Z, Kidwai F, Scotch M, Womack J, Justice A, Brandt C (2011) The yale ctakes extensions for document classification: architecture and application. J Am Med Inform Assoc 18(5):614–620
28. Li M, Fei Z, Zeng M, Wu F-X, Li Y, Pan Y, Wang J (2018) Automated icd-9 coding via a deep learning approach. IEEE/ACM Trans Comput Biol Bioinf 16(4):1193–1202
29. Spasic I, Krzeminski D, Corcoran P, Balinsky A et al (2019) Cohort selection for clinical trials from longitudinal patient records: text mining approach. JMIR Med Inform 7(4):15980
30. Huang J, Osorio C, Sy LW (2019) An empirical evaluation of deep learning for ICD-9 code assignment using mimic-iii clinical notes. Comput Methods Programs Biomed 177:141–153

31. Wang Y, Sohn S, Liu S, Shen F, Wang L, Atkinson EJ, Amin S, Liu H (2019) A clinical text classification paradigm using weak supervision and deep representation. BMC Med Inform Decis Mak 19:1–13
32. Ni Y, Wright J, Perentesis J, Lingren T, Deleger L, Kaiser M, Kohane I, Solti I (2015) Increasing the efficiency of trial-patient matching: automated clinical trial eligibility pre-screening for pediatric oncology patients. BMC Med Inform Decis Mak 15:1–10
33. Garla VN, Brandt C (2012) Ontology-guided feature engineering for clinical text classification. J Biomed Inform 45(5):992–998
34. Mujtaba G, Shuib L, Raj RG, Rajandram R, Shaikh K, Al-Garadi MA (2017) Automatic icd-10 multi-class classification of cause of death from plaintext autopsy reports through expert-driven feature selection. PLoS One 12(2):0170242
35. Scheurwegs E, Cule B, Luyckx K, Luyten L, Daelemans W (2017) Selecting relevant features from the electronic health record for clinical code prediction. J Biomed Inform 74:92–103
36. Soguero-Ruiz C, Hindberg K, Rojo-Alvarez JL, Skrøvseth SO, Godtliebsen F, Mortensen K, Revhaug A, Lindsetmo R-O, Augestad KM, Jenssen R (2014) Support vector feature selection for early detection of anastomosis leakage from bag-of-words in electronic health records. IEEE J Biomed Health Inform 20(5):1404–1415
37. Venkataraman GR, Pineda AL, Bear Don't Walk IV OJ, Zehnder AM, Ayyar S, Page RL, Bustamante CD, Rivas MA (2020) Fastag: automatic text classification of unstructured medical narratives. PLoS One 15(6):0234647
38. Nigam P (2016) Applying deep learning to icd-9 multi-label classification from medical records. Technical report, Technical report, Stanford University
39. Perotte A, Pivovarov R, Natarajan K, Weiskopf N, Wood F, Elhadad N (2014) Diagnosis code assignment: models and evaluation metrics. J Am Med Inform Assoc 21(2):231–237
40. Mustafa A, Rahimi Azghadi M (2021) Automated machine learning for healthcare and clinical notes analysis. Computers 10(2):24
41. Zheng T, Xie W, Xu L, He X, Zhang Y, You M, Yang G, Chen Y (2017) A machine learning-based framework to identify type 2 diabetes through electronic health records. Int J Med Inform 97:120–127
42. Mani S, Chen Y, Elasy T, Clayton W, Denny J (2012) Type 2 diabetes risk forecasting from emr data using machine learning. In: AMIA Annual Symposium Proceedings, vol. 2012, p. 606. American Medical Informatics Association
43. Liu X, Chen Y, Bae J, Li H, Johnston J, Sanger T (2019) Predicting heart failure readmission from clinical notes using deep learning. In: 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp 2642–2648. IEEE
44. Wei Q, Ji Z, Li Z, Du J, Wang J, Xu J, Xiang Y, Tiryaki F, Wu S, Zhang Y et al (2020) A study of deep learning approaches for medication and adverse drug event extraction from clinical text. J Am Med Inform Assoc 27(1):13–21
45. Walsh CG, Ribeiro JD, Franklin JC (2017) Predicting risk of suicide attempts over time through machine learning. Clin Psychol Sci 5(3):457–469
46. Shi H, Xie P, Hu Z, Zhang M, Xing EP (2017) Towards automated icd coding using deep learning. arXiv preprint arXiv:1711.04075
47. Kocbek S, Kocbek P, Zupanic T, Stiglic G, Gabrys B (2019) Using (automated) machine learning and drug prescription records to predict mortality and polypharmacy in older type 2 diabetes mellitus patients. In: Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings, Part IV 26, pp 624–632. Springer
48. Gonzalez-Hernandez G, Sarker A, O'Connor K, Savova G (2017) Capturing the patient's perspective: a review of advances in natural language processing of health-related text. Yearb Med Inform 26(01):214–227
49. Jolley RJ, Quan H, Jetté N, Sawka KJ, Diep L, Goliath J, Roberts DJ, Yipp BG, Doig CJ (2015) Validation and optimisation of an icd-10-coded case definition for sepsis using administrative health data. BMJ Open 5(12):009487
50. Kaur R, Ginige JA (2018) Comparative analysis of algorithmic approaches for auto-coding with icd-10-am and achi. Stud Health Technol Inform 252:73–79
51. Steindel SJ (2010) International classification of diseases, clinical modification and procedure coding system: descriptive overview of the next generation hipaa code sets. J Am Med Inform Assoc 17(3):274–282
52. Harrison JE, Weber S, Jakob R, Chute CG (2021) Icd-11: an international classification of diseases for the twenty-first century. BMC Med Inform Decis Mak 21(6):1–10
53. International Classification of Diseases and Injuries: 11th Issue Launched By WHO (2018). https://www.eurosafe.eu.com/news/international-classification-of-diseases-and-injuries-11th-issue-launched-by-who
54. ICD-11 2022 release. World Health Organization (2022). https://www.who.int/news/item/11-02-2022-icd-11-2022-release
55. Kaur R, Ginige JA (2018) Comparative analysis of algorithmic approaches for auto-coding with icd-10-am and achi. Stud Health Technol Inform 252:73–79
56. Cartwright DJ (2013) ICD-9-CM to ICD-10-CM codes: what? why? how? Mary Ann Liebert, Inc. 140 Huguenot Street, 3rd Floor New Rochelle, NY 10801 USA
57. Jolley RJ, Quan H, Jetté N, Sawka KJ, Diep L, Goliath J, Roberts DJ, Yipp BG, Doig CJ (2015) Validation and optimisation of an icd-10-coded case definition for sepsis using administrative health data. BMJ Open 5(12):009487
58. Organization WH et al (1978) International classification of diseases: [9th] ninth revision, basic tabulation list with alphabetic index. World Health Organization
59. Karmakar A (2018) Classifying medical notes into standard disease codes using machine learning. arXiv preprint arXiv:1802.00382
60. Yogarajan V, Montiel J, Smith T, Pfahringer B (2020) Seeing the whole patient: using multi-label medical text classification techniques to enhance predictions of medical codes. arXiv preprint arXiv:2004.00430
61. Clinic® AC (2005) ICD-9-CM Coding Advice for Healthcare Encounters in Hurricane Aftermath. AHA Coding Clinic®
62. Yogarajan V (2022) Domain-specific language models for multi-label classification of medical text. PhD thesis, The University of Waikato
63. Harbecke D, Chen Y, Hennig L, Alt C (2022) Why only micro-f1? class weighting of measures for relation classification. arXiv preprint arXiv:2205.09460
64. Catling F, Spithourakis GP, Riedel S (2018) Towards automated clinical coding. Int J Med Inform 120:50–61
65. Yan C, Fu X, Liu X, Zhang Y, Gao, Y, Wu J, Li Q (2022) A survey of automated icd coding: development, challenges, and applications. Intelligent Medicine
66. Singh A, Guntu M, Bhimireddy AR, Gichoya JW, Purkayastha S (2020) Multi-label natural language processing to identify diagnosis and procedure codes from mimic-iii inpatient notes. arXiv preprint arXiv:2003.07507
67. Xu K, Lam M, Pang J, Gao X, Band C, Mathur P, Papay F, Khanna AK, Cywinski JB, Maheshwari K et al (2019) Multimodal machine learning for automated icd coding. In: Machine Learning for Healthcare Conference, pp 197–215. PMLR

68. Cheung BM, Li C (2012) Diabetes and hypertension: Is there a common metabolic pathway? Curr Atheroscler Rep 14:160–166
69. Ceylan Z, Pekel E (2017) Comparison of multi-label classification methods for prediagnosis of cervical cancer. Graph Models 21:22
70. Read J, Pfahringer B, Holmes G, Frank E (2011) Classifier chains for multi-label classification. Mach Learn 85:333–359
71. Wang R, Ye S, Li K, Kwong S (2021) Bayesian network based label correlation analysis for multi-label classifier chain. Inf Sci 554:256–275
72. Konkle BA, Fletcher SN (2022) Hemophilia a. GeneReviews®[Internet]
73. Wever M, Tornede A, Mohr F, Hüllermeier E (2021) Automl for multi-label classification: overview and empirical evaluation. IEEE Trans Pattern Anal Mach Intell 43(9):3037–3054
74. Moyano Murillo JM (2020) Multi-label classification models for heterogeneous data: an ensemble-based approach
75. Yapp EK, Li X, Lu WF, Tan PS (2020) Comparison of base classifiers for multi-label learning. Neurocomputing 394:51–60
76. Tsoumakas G, Vlahavas I (2007) Random k-labelsets: an ensemble method for multilabel classification. In: Machine Learning: ECML 2007: 18th European Conference on Machine Learning, Warsaw, Poland, September 17-21, 2007. Proceedings 18, pp 406–417. Springer
77. Zhang M-L, Zhou Z-H (2005) A k-nearest neighbor based algorithm for multi-label classification. In: 2005 IEEE International Conference on Granular Computing, vol. 2, pp 718–721. IEEE
78. Lim H, Lee J, Kim D-W (2017) Optimization approach for feature selection in multi-label classification. Pattern Recogn Lett 89:25–30
79. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3(Mar):1157–1182
80. Yu L, Liu H (2003) Feature selection for high-dimensional data: a fast correlation-based filter solution. In: Proceedings of the 20th International Conference on Machine Learning (ICML-03), pp 856–863
81. Kaur R, Ginige JA (2018) Comparative analysis of algorithmic approaches for auto-coding with icd-10-am and achi. Stud Health Technol Inform 252:73–79
82. Su X, Chen N, Sun H, Liu Y, Yang X, Wang W, Zhang S, Tan Q, Su J, Gong Q et al (2020) Automated machine learning based on radiomics features predicts h3 k27m mutation in midline gliomas of the brain. Neuro Oncol 22(3):393–401
83. Tsoumakas G, Katakis I (2007) Multi-label classification: an overview. Int J Data Warehous Min (IJDWM) 3(3):1–13
84. Truong A, Walters A, Goodsitt J, Hines K, Bruss CB, Farivar R (2019) Towards automated machine learning: evaluation and comparison of automl approaches and tools. In: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), pp 1471–1479. IEEE
85. AlGhanem H, Mustafa A, Abdallah S (2020) Knowledge and human development authority in Dubai (khda) open data: What do researchers want? In: Information Systems: 16th European, Mediterranean, and Middle Eastern Conference, EMCIS 2019, Dubai, United Arab Emirates, December 9–10, 2019, Proceedings 16, pp 58–70. Springer
86. Thornton C, Hutter F, Hoos HH, Leyton-Brown K (2013) Auto-weka: combined selection and hyperparameter optimization of classification algorithms. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 847–855
87. Kotthoff L, Thornton C, Hoos HH, Hutter F, Leyton-Brown K (2019) Auto-weka: automatic model selection and hyperparameter optimization in Weka. Automated machine learning: methods, systems, challenges, 81–95
88. Read J, Reutemann P, Pfahringer B, Holmes G (2016) Meka: a multi-label/multi-target extension to weka
89. Sá AG, Freitas AA, Pappa GL (2018) Automated selection and configuration of multi-label classification algorithms with grammar-based genetic programming. In: Parallel Problem Solving from Nature–PPSN XV: 15th International Conference, Coimbra, Portugal, September 8–12, 2018, Proceedings, Part II, pp 308–320. Springer
90. Feurer M, Klein A, Eggensperger K, Springenberg J, Blum M, Hutter F (2015) Efficient and robust automated machine learning. In: Cortes C, Lawrence N, Lee D, Sugiyama M, Garnett R (eds) Advances in neural information processing systems, vol. 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2015/file/11d0e6287202fced83f79975ec59a3a6-Paper.pdf
91. Feurer M, Eggensperger K, Falkner S, Lindauer M, Hutter F (2020) Auto-sklearn 2.0: the next generation. arXiv preprint arXiv:2007.04074 24
92. Feurer M, Eggensperger K, Falkner S, Lindauer M, Hutter F (2022) Auto-sklearn 2.0: Hands-free automl via meta-learning. J Mach Learn Res 23(1):11936–11996
93. McKenzie K, Walker S, Dixon-Lee C, Dear G, Moran-Fuke J (2004) Clinical coding internationally: a comparison of the coding workforce in Australia, America, Canada and England. In: The 14th International Federation of Health Records Organizations (IFHRO) Congress and the 76th AHIMA National Convention Proceedings, pp 52–64. American Health Information Management Association
94. Scikit-learn.org: 1.12. Multiclass and multilabel algorithms-scikit-learn 0.21.3 documentation (2009). https://scikit-learn.org/stable/modules/multiclass.html
95. Queensland TU (2023) Research Computing Centre-the University of Queensland, Australia. https://rcc.uq.edu.au/tinaroo Accessed 2023-02-23
96. Feurer M (2020) Remove warning "No models better than random - using Dummy Score!" fix 739 762. https://github.com/automl/auto-sklearn/pull/762 Accessed 2023-02-22
97. APIs - AutoSklearn 0.15.0 documentation. https://automl.github.io/auto-sklearn/master/api.html Accessed 2023-02-23
98. Zhang M-L, Zhou Z-H (2013) A review on multi-label learning algorithms. IEEE Trans Knowl Data Eng 26(8):1819–1837
99. Sajid NA, Rahman A, Ahmad M, Musleh D, Basheer Ahmed MI, Alassaf R, Chabani S, Ahmed MS, Salam AA, AlKhulaifi D (2023) Single vs. multi-label: the issues, challenges and insights of contemporary classification schemes. Appl Sci 13(11):6804
100. Tarekegn AN, Giacobini M, Michalak K (2021) A review of methods for imbalanced multi-label classification. Pattern Recogn 118:107965