computers

MDPI

*Review*

# Automated Machine Learning for Healthcare and Clinical Notes Analysis

**Akram Mustafa and Mostafa Rahimi Azghadi ***

College of Science and Engineering, James Cook University, Townsville, QLD 4811, Australia; akram.mohdmustafa@my.jcu.edu.au
* Correspondence: mostafa.rahimiazghadi@jcu.edu.au

**Abstract:** Machine learning (ML) has been slowly entering every aspect of our lives and its positive impact has been astonishing. To accelerate embedding ML in more applications and incorporating it in real-world scenarios, automated machine learning (AutoML) is emerging. The main purpose of AutoML is to provide seamless integration of ML in various industries, which will facilitate better outcomes in everyday tasks. In healthcare, AutoML has been already applied to easier settings with structured data such as tabular lab data. However, there is still a need for applying AutoML for interpreting medical text, which is being generated at a tremendous rate. For this to happen, a promising method is AutoML for clinical notes analysis, which is an unexplored research area representing a gap in ML research. The main objective of this paper is to fill this gap and provide a comprehensive survey and analytical study towards AutoML for clinical notes. To that end, we first introduce the AutoML technology and review its various tools and techniques. We then survey the literature of AutoML in the healthcare industry and discuss the developments specific to clinical settings, as well as those using general AutoML tools for healthcare applications. With this background, we then discuss challenges of working with clinical notes and highlight the benefits of developing AutoML for medical notes processing. Next, we survey relevant ML research for clinical notes and analyze the literature and the field of AutoML in the healthcare industry. Furthermore, we propose future research directions and shed light on the challenges and opportunities this emerging field holds. With this, we aim to assist the community with the implementation of an AutoML platform for medical notes, which if realized can revolutionize patient outcomes.

**Keywords:** AutoML; machine learning; natural language processing; clinical coding; clinical notes



**Citation:** Mustafa, A.; Rahimi Azghadi, M. Automated Machine Learning for Healthcare and Clinical Notes Analysis. *Computers* **2021**, *10*, 24. https://doi.org/10.3390/computers10020024

Academic Editors: Antonio Celesti, Ivanoe De Falco, Antonino Galletta and Giovanna Sannino

Received: 1 February 2021
Accepted: 17 February 2021
Published: 22 February 2021



## 1. Introduction

Health and well-being is, undoubtedly, one of the most fundamental concerns of human beings. This is demonstrated by the sheer size and the fast growth of global healthcare industries, which is projected to reach over 10 trillion dollars by 2022 [1]. One of the most promising technologies to advance this fast-growing industry is artificial intelligence (AI) [2] and its implementation with machine learning (ML). With the recent advances in ML technology comes an opportunity to improve healthcare and enhance patient outcomes.

ML has been extensively used in a variety of healthcare and medical applications including but not limited to cardiovascular risk and heart diseases identification [3,4], oral disease diagnosis and prediction [5], and discovering cancer tumors from radiology images [6]. Recently, AutoML [7–9] has been proposed to expand the application domain of ML algorithms and facilitate its deployment in many areas including healthcare [10,11]. Although still an emerging technology, AutoML has been already used in medical imaging [12], bioinformatics, translational medicine, diabetes diagnosis [13], Alzheimer diagnosis [14], and electronic health record (EHR) analysis. However, its use in processing clinical notes, which are a main category of EHR, has not been widely explored. In particular, most

of the previous works [15–18] on clinical notes have used standard ML to process them, e.g., for diagnosis [19,20]. These studies have deployed ML algorithms in combination with other techniques such as natural language processing (NLP) [21], concept extraction solutions [22], and optimization [23].

The main motivation of this paper is to fill the identified gap of the lack of comprehensive research in the area of AutoML applications for healthcare and in particular for clinical notes analysis. To that end, the scope of this paper is confined to surveying and analyzing the following study categories, which help in systematically covering the core motivation of this paper, i.e., AutoML for healthcare applications and clinical notes analysis.

- AutoML platforms: The papers in this category cover generic AutoML libraries and platforms, such as Google AutoML platform [24], and Auto-Sklearn [7].
- AutoML tools in the healthcare industry: The papers in this category cover AutoML tools that were built specifically for healthcare industry, such as JADBIO [13], and AutoPrognosis [25]. We also cover research papers that use existing general-purpose AutoML tools for medical purposes.
- Towards AutoML for clinical notes analysis: The papers in this category cover ML research to extract diagnoses from clinical notes. We discuss how previous ML methods used for medical notes analysis can be used towards AutoML for clinical notes diagnoses.

Figure 1 illustrates the organization of our paper. Section 1 provides the background and motivation of this paper. Section 2 covers the fundamental concept of AutoML and introduces its available tools and techniques. Section 3 surveys the use of AutoML technology in the healthcare industry. Section 4 provides details on different machine learning stages that are required to extract diagnoses from clinical notes. This Section surveys research on preprocessing, feature extraction and selection, algorithm selection and optimization, and evaluation stages of an AutoML platform for clinical notes. In this section, we also provide insight into future research directions for developing AutoML for clinical notes and discuss the challenges and opportunities this may bring. Finally, Section 5 provides concluding remarks of our paper.

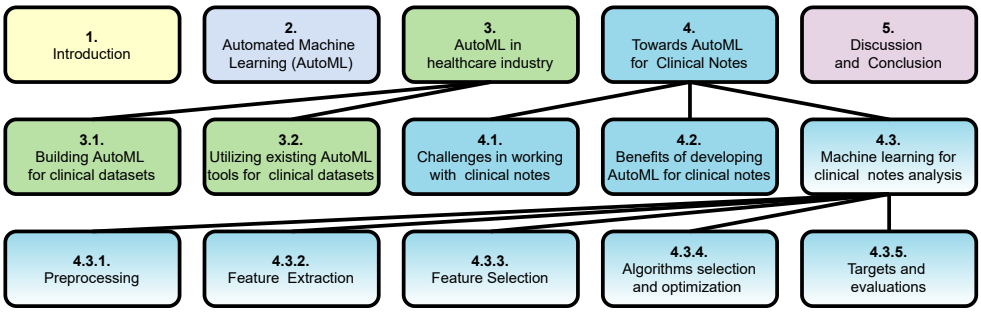


**Figure 1.** The structure of this paper at a glance.

## 2. Automated Machine Learning (AutoML)

Although ML algorithms do not require human interference while learning, preparing data that is going to be consumed by these algorithms, finding the right algorithm, and tweaking it to get the best results require skilled data scientists. Data scientists try different techniques for preprocessing and multiple ML algorithms to come up with the combination that is the most efficient. These processes are human dependent and require special skills in computer science, programming, mathematics, and statistics, in addition to business knowledge in the area of the processed data. Not all data scientists have these skills. The ones who may have all these skills are called "data science unicorns" as they are rare, therefore, many organizations recruit multiple data scientists, and data analysts to perform the complete ML processes. One of the ways to resolve this issue is by introducing AutoML, a process that is completely automated and reduces human interference to a minimum [8,26].

AutoML automates the main processes of ML as shown in Figure 2. The first step in these processes is data preparation, which includes data integration, data transformation, data cleaning, and data reduction. Data preparation is a lengthy process and takes most of a data engineer's time [27]. Feature extraction and selection is the next step, which selects a subset of dataset features that preserve the information in the dataset, while improving the learning generalization [28]. Algorithm selection is the next phase, in which a method is used to select the best algorithm that provides the most accurate results. Tweaking the algorithms' settings to enhance results further is called hyperparameter optimization [29]. AutoML systems use different methods and optimization techniques to achieve the desired accuracy and performance. Bayesian optimization is a technique that optimizes hyperparameters for ML algorithms based on a well-known theory in probabilities called Bayes' theorem [7,30,31]. Other simpler techniques are also used such as grid search and random search. Meta-Learning is another method for hyperparameter optimization, where the AutoML system learns from its own experience of applying machine learning; this is also called learning to learn [7–9].

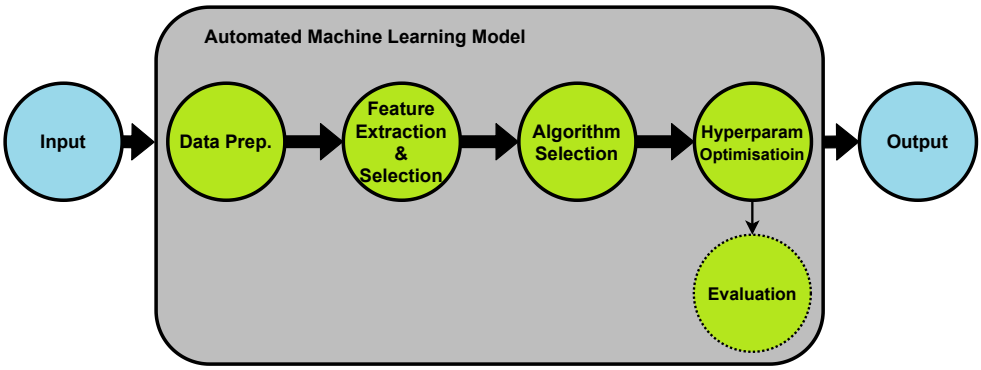


**Figure 2.** AutoML model main processes.

There are many AutoML platforms available. Some of these platforms are open-source while others are commercial. Table 1 shows a comparison among some of the most popular AutoML platforms, in terms of coding requirements, processing location, accepted input data, and cost. We have also included two popular platforms specifically developed for medical domain, which will be discussed in the next section.

One of the most popular open-source AutoML libraries is called Auto-Sklearn, which is short for automated science kit learn, a free open-source Python library that was developed by researchers from the University of Freiburg in Germany on top of the Sklearn library [7]. This library automates algorithm selection and hyperparameter optimization through using Bayesian optimization techniques and meta-learning. Another open-source platform is Auto-WEKA [31,32], which is short for automated Waikato environment for knowledge analysis and was developed at the University of British Columbia. It is similar to Auto-Sklearn but was built on top of Java's weka library, which was developed at the University of Waikato in New Zealand. Auto-WEKA uses Bayesian optimization for hyperparameter optimization. Both AutoML platforms use statistical algorithms and can only process structured data such as stock market prices, students' grades, hotels occupancy, etc.

**Table 1.** AutoML platforms comparison.

| AutoML Platform | Cost | Coding | Location | Dataset | Domain |
|---|---|---|---|---|---|
| Google AutoML | Chargeable | No coding | Cloud | Images<br>Text<br>Tabular | Generic |
| Apple Create ML | Free | Coding needed | Local | Images<br>Text<br>Tabular | Generic |
| Amazon AutoML | Chargeable | No coding | Cloud | Images<br>Text<br>Tabular | Generic |
| Microsoft AutoML | Chargeable | No coding | Cloud | Images<br>Text<br>Tabular | Generic |
| Auto-Sklearn | Free | Coding needed | Local | Tabular | Generic |
| Auto-WEKA | Free | Coding needed | Local | Tabular | Generic |
| Auto-Keras | Free | Coding needed | Local | Tabular | Generic |
| TPOT | Free | Coding needed | Local | Tabular | Generic |
| JADBIO | Chargeable | No coding | Cloud | Tabular | Medical |
| AutoPrognosis | Free | Coding needed | Local | Tabular | Medical |

A commercial AutoML platform example is RapidMiner [33], a form of guided AutoML where the whole machine learning process is automated from data preparation through to algorithm and hyperparameter selection. It is based on a pipeline that guides the process and decides which methods and algorithms to use, along with data analysis, visualization, and text mining. Google has its own cloud AutoML platform as well. The Google AutoML platform [24] is divided into different models based on the type of the input datasets. AutoML Table is used for structured data; AutoML Natural Language is used for text classification and entities identification; AutoML Translation is used for translations; AutoML Vision is for image classification and object detection; and finally AutoML Video Intelligence is dedicated to video classification and object tracking.

All these systems automate the machine learning process yet each one of them works differently, and targets different datasets, platforms, algorithms or users, while having unique advantages and disadvantages. For instance, Auto-Sklearn library can be embedded within a Python application but it only works for structured data and on Linux. Auto-WEKA has a graphical user interface (GUI) and can be used as a Java command, but it is limited to statistical algorithms. RapidMiner has data analysis capability but requires human guidance. Google AutoML covers many types of datasets, yet it is only cloud-based and charges for the data processing [7,8,24,31–33].

## 3. AutoML in Healthcare Industry

Although AutoML has been used in a variety of applications such as fraud detection [34] and disease diagnoses [12], there are many more applications that still use traditional ML processes rather than AutoML. This is due to the nature of these applications that require processes such as data cleaning and feature selection that are not supported by most AutoML platforms.

In healthcare industry, there have been a few studies focused on implementing AutoML systems that are specialized for health services. Considering the lack of funds for clinical coding [35] and high data scientist salaries [36], it is essential to find a cost-saving method that allows health organizations to benefit from machine learning capabilities without huge costs. More importantly, such a method may improve patient outcomes, which is of paramount importance in developing any healthcare tool. As an emerging

technology, AutoML can help achieve these goals for health organizations, especially for extracting diagnoses from clinical notes, which is the focus of this paper. This will not only save health workers' time, but it will also improve patient outcomes by accelerating patient treatment planning and improving the accuracy of diagnoses.

Overall, we found two general approaches that have been studied to use AutoML in the healthcare industry. The first approach is to build new AutoML tools for medical datasets, while the second approach is to use already existing AutoML libraries and platforms to perform predictive modeling or classification on a clinical dataset. Below, we discuss these approaches in more details.

### *3.1. Building AutoML for Clinical Datasets*

In this approach, an AutoML tool that is specialized in analyzing clinical data, for instance for proposing care management plan and predicting patient's clinical costs [37], or classifying medical records and images [12], is built. The datasets dealt with are either structured or unstructured. In a structured dataset the data is represented in the form of rows and columns and can be easily processed by computers. These datasets reside in databases, spread sheets, and other platforms that support tabular data format. An example of structured datasets is lab results, which consist of patient's information such as name, age, gender, and test results such as Hemoglobin, and Cholesterol level [38–40]. Unstructured datasets, on the other hand, are unformatted data. This includes text, images, videos, and documents that are not in a tabular format. Most of ML algorithms such as linear regression or support vector machine (SVM) need a transition process that converts unstructured data into a structured format. Around 90% of the available data is unstructured, and 90% of this unstructured data has not been used yet. Examples of unstructured datasets are clinical notes and medical images [41].

Tsamardinos et al. [13] have built an AutoML system that is specialized in bioinformatic applications and translational medicine. Their platform, named just add data bio (JADBIO), also has built-in predictive and diagnostic clinical models. JADBIO works by using biosignatures of dataset features and can interpret and visualize results. It can work with a dataset including only small number of records, as few as 25, while being also capable of processing high-dimensional datasets of hundreds to thousands of features. Feature engineering, algorithm selection, and hyperparameter options and scopes are identified by algorithm and the hyperparameter space (AHPS) method. AHPS uses parameters such as dataset size, feature dimensionality, and targeted value type to identify a list of relevant algorithms, and to identify a list of methodologies for feature selection and data preprocessing, as well as to define hyperparameter scope.

AHPS's output is fed to a configuration generator (CG) to generate a list of pipelines with available hyperparameters. Then, configuration evaluation protocol (CEP) uses k-fold cross validation to determine the best data preprocessing methods, feature engineering algorithms, and hyperparameters, and to assess the model's performance. CEP selection is then applied to the original dataset and predictive models are built. Figure 3 shows how JADBIO AutoML model works.

Tsamardinos et al. have used multiple ML algorithms for building JADBIO. These include linear ridge regression, SVM, decision tree (DT), random forests (RF), and Gaussian kernel SVMs. JADBIO has been also compared to Auto-Sklearn system using 748 datasets. Auto-Sklearn failed to process around 39.44% of the datasets due to timeout and internal errors, yet JADBIO's performance was close to Auto-Sklearn's performance for the remaining datasets.

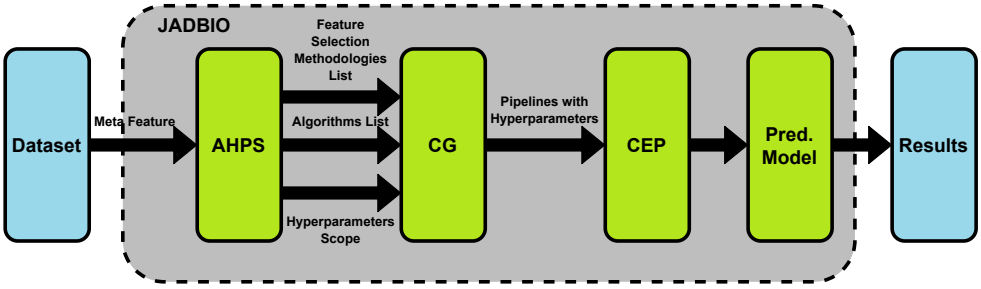


**Figure 3.** JADBIO AutoML [13]: Dataset meta features are fed into the algorithm and hyperparameters space (AHPS), which feeds the configuration generator (CG) with a list of feature selection and data preprocessing methodologies, relevant algorithms list, and hyperparameters scope, then configuration evaluation protocol (CEP) finds the best machine learning model with the best performance.

Luo et al. [37], proposed methods as part of an under-development AutoML system that can help healthcare experts perform predictions and classifications on big clinical data without data scientist involvement. They identified three hurdles that affect the AutoML process. The first is feature and algorithm selection, as well as hyperparameters optimization. Feature selection methods and models of selecting algorithms, and identifying optimum hyperparameters can lead to thousands of routes to try every possible combination. The second hurdle is grouping and accumulating data, which require data scientists. This issue is part of data preparation and feature-extraction processes. The third hurdle is generalizing machine learning models. Each medical dataset has its own characteristics and needs a different ML model to get good accuracy. One model cannot fit all datasets; therefore, a data scientist is needed to redesign a new model.

Luo et al. identified a few goals they plan to achieve in their project. The first goal is to craft a method to automate feature, algorithm, and hyperparameters selection. Another goal is to build a method for data accumulation. They also plan to validate their proposed model on nine modeling problems, and to estimate the significance of using their proposed AutoML in the USA, through simulation. They have studied features and algorithms selection, and hyperparameter optimization methods that were presented in [42]. They found that the methods surveyed in [42], i.e., combined algorithm selection and hyperparameter optimization (CASH), and sequential model-based algorithm configuration (SMAC), are not efficient for big data. Therefore, they created a new method that uses Bayesian optimization.

Other research on building an AutoML system for healthcare have also used Bayesian optimization, such as AutoPrognosis model [25] that automates clinical prognostic modeling, and FLASH model [43], which uses double layers of Bayesian optimization to effectively select algorithms and optimize hyperparameters.

Kim et al. [44] used an AutoML method called neural architecture search (NAS) to optimize neural network models for high resolution 3D medical images, which require lots of resources, computational power and large memory. The architecture was built based on U-Net, which is a convolutional neural network that is used to segment biomedical images [45]. Their method relied on alternating encoders and decoders and working with both discrete and continuous parameters using Gumbel SoftMax sampling and stochastic sampling algorithms, which were developed to reduce the number of used resources. Similarly, Weng et al. [46] used NAS to build their model called NAS-Unet for optimizing convolutional neural networks for 2D medical image segmentation based on U-Net architecture. They used mean intersection over union (MIOU) evaluation which is common for semantic image segmentation, and the results of NAS-Unet were better than U-Net architecture.

### *3.2. Using Existing AutoML Tools for Clinical Datasets*

Existing AutoML tools can be used for a variety of applications, including medical data analysis. Below, we review several recent works that used existing AutoML methods to address a clinical problem.

Borkowski et al. [12] have used and compared two AutoML platforms, i.e., Google's AutoML, and Apple's Create ML. They experimented on six small lung and colon cancer balanced image datasets. The size of these datasets was between 250 and 750 images. The prediction results were close, but Apple's Create ML had best recall results in four out of six datasets. Both Google and Apple tools have limitations in terms of the platform they work on. The Google AutoML tool incurs a cost to the medical users to process datasets and classify medical images. In addition, if higher computational resources are required, which is usually the case for large-scale medical data analysis, it creates additional costs. Furthermore, Google's model must store medical images and data on its cloud platform, which is undesired for sensitive patient data. Apple's model, on the other hand, is free of charge and can store data locally. On the downside, it is only available to iOS users.

In another study of using and evaluating available AutoML platforms, Ooms et al. [11] compared different AutoML libraries including TPOT [47], Auto-Keras [48], Auto-Sklearn [7], Auto-WEKA [31], AutoPrognosis [25], Flash [43], AlphaD3M [49], AutoNet [50], ATM [51], Hyperout-Sklearn [52], ML-Plan [53], PoSH Auto-Sklearn [54], RECIPE [55], Layered TPOT [56], and Autostacker [57]. They used these AutoML libraries to classify four binary datasets for breast cancer, diabetes diagnosis, and sick patient identification. As a result of their evaluation, they selected TPOT to perform the classification in the backend of their solution to support medical researchers.

Instead of using general AutoML platforms, some researchers have used the AutoML systems specifically built for medical data. For instance, Karaglani et al. [14] used JADBIO to diagnose Alzheimer disease using blood-based diagnostic biosignatures. Their datasets consisted of low-sample omics data with high-dimensional features. They tested seven datasets with different biosignatures: two metabolomic datasets, one proteomic dataset, and four transcriptomic datasets. Sample numbers were between 30 and 589, while the number of features was between 25 to 38,327 features. They used area under the curve (AUC) to evaluate predicted results and got AUC accuracy between 0.489 and 0.975, with an average AUC of 0.759.

Although a few significant works such as JADBIO have been dedicated to developing AutoML tools for medical datasets, most of the research found have used out-of-the-box AutoML tools for medical data. Examples include medical prediction and classification studies that used TPOT [58–60], Google AutoML vision [61–65], or Auto-WEKA [66]. Table 2 lists several previous works that have used different AutoML platforms for various medical applications. The table categorizes the surveyed research into two general groups based on the dataset type, which can be structured, i.e., tabular, or unstructured, which in this case are audio or images.

**Table 2.** Some of the AutoML platforms used in the healthcare domain research.

| Dataset Format | Dataset Type | Disease/ Speciality | Research | AutoML Platform | | |
|---|---|---|---|---|---|---|
| | | | | Commercial | Open Source | Health-Related |
| Unstructured | Audio | Hearing Aid | [67] | ✓ | × | × |
| | Images | Cancer | [12] | ✓ | × | × |
| | | | [61] | ✓ | × | × |
| | | Covid-19 | [62] | ✓ | × | × |
| | | Generic | [44] | × | × | ✓ |
| | | | [46] | × | × | ✓ |
| | | | [63] | ✓ | × | × |
| | | Liver Injury | [64] | ✓ | × | × |
| | | Pachychoroid | [65] | ✓ | × | × |
| Structured | Tabular | Alzheimer | [14] | × | × | ✓ |
| | | BioSignature | [13] | × | ✓ | ✓ |
| | | Brain Age | [58] | × | ✓ | × |
| | | Brain Tumor | [59] | × | ✓ | × |
| | | Cardiac | [25] | × | ✓ | ✓ |
| | | Diabetes | [66] | × | ✓ | × |
| | | Generic | [37] | × | ✓ | ✓ |
| | | | [11] | × | ✓ | ✓ |
| | | Metabolic | [60] | × | ✓ | × |

As the table shows, most of the unstructured data is processed using commercial AutoML tools, while structured data can be usually processed using open-source tools and medical AutoML platforms. The main reason for this is that dealing with structured data is easier due to their process-ready nature. Much research is available on these AutoML platforms as most were built by academic institutions such as the University of British Columbia that developed Auto-WEKA, University of Freiburg that designed Auto-Sklearn, and the University of Pennsylvania that created TPOT [68]. It seems that due to the complex nature of the unstructured data, they cannot be readily handled by the limited open-source tools. On the other hand, commercial companies that achieve financial benefits by providing AutoML platforms have developed more advanced tools that can process images and text. Examples include Google's AutoML, Microsoft's AutoML, Apple's Create ML, and Amazon's Rekognition. These companies invested heavily on AutoML platform developments. Therefore, their tools became very easy to use with almost no coding required for most of their products [63].

From a clinical practitioner's perspective, these AutoML tools can be very useful as they do not require any machine learning or coding experience. Therefore, medical professionals can use one or more of these tools to identify different diseases based on available datasets of imaging, lab results, symptoms, medical history, etc. Most of AutoML platforms are disease agnostic, which means they can work on any diseases provided in the training dataset. A general AutoML tool recommendation for clinical practitioners based on their different clinical situations and data is as follows:

- Google AutoML Vision would be the easiest tool to work with medical images. It does not require tool installation or coding. Clinical practitioner can upload image datasets to train their model and then use it for diagnoses. Previous examples include cancer [12] and pneumonia [63] detection based on X-ray images.

- For a small number of records in high-dimensional datasets, JADBIO would be the recommended tool to use to identify diseases. For instance, it was used to identify Alzheimer [14] and Parkinson [69] diseases.
- For assessing patients risks using biometrics and patient's medical history, AutoPrognosis can be used. Use case examples include prognosis of cardiovascular disease [25].

## 4. Towards AutoML for Clinical Notes

As shown in Table 2, previous research on developing AutoML in the healthcare domain are mainly for medical imaging (unstructured data), or for tabular structured datasets. However, while there is an abundance of research applying conventional ML models on clinical notes for classification and diagnoses, prior research that implements AutoML on clinical notes, i.e., unstructured medical text, is very limited. This is mainly due to the challenging nature of clinical notes. Below, we first explain the challenges faced when working with clinical notes, and then review machine learning research to process clinical notes in a stage-wise manner. We also provide motivations and insight for developing AutoML for clinical notes and challenges and opportunities at each stage of an AutoML platform.

### *4.1. Challenges in Working with Clinical Notes*

Clinical notes are a form of unstructured data with useful information hidden in the clinical text. In general, the linguistics of these notes need special analyses, as they contain a level of ambiguity that requires the use of different NLP methodologies to clarify. In particular, clinical text contains many of medical terminologies and non-clinical words that can confuse computers about how to classify them. For example, the word "BAD" can mean bad, i.e., the opposite of good, or it can mean bipolar affective disorder [70].

Furthermore, it is a challenging problem to decide which NLP techniques to use for the data preparation stage of ML or AutoML. Therefore, it requires a data scientist. After the data is prepared, many features from clinical text can be generated. This large number can affect model's performance significantly. In addition, considering that AutoML models apply multiple algorithms and methodologies to the features set, it greatly affects the processing time of the AutoML model.

Medical abbreviation is another challenge, as it is commonly used in clinical notes but get filtered out by stop words (which will be discussed later) removal algorithms, because the abbreviations can be similar to stop words. Examples of these stop works include "AND" which means allowing natural death, and "IS" which means incentive spirometry.

Typos in the medical reports can be another challenge as spelling a word wrong could indicate another disease or diagnoses. Examples include "clot" and "blot", "ADHD" which means attention deficit hyperactivity disorder and "ADHF" which means acute decompensated heart failure. Also, the large number of targeted codes, which for example count around 70,000 codes for international classification of diseases version 10 Australian modification (ICD10AM), affects model accuracy.

### *4.2. Benefits of Developing AutoML for Clinical Notes*

If the aforementioned difficulties are addressed, remarkable benefits for patients and the medical system will be achieved. This will happen by minimizing the need for skilled clinical coders, which will significantly help with the lack of funds for clinical coding in the medical system [35]. More importantly, significant improvement in patient outcomes can be achieved by faster and more accurate diagnoses and prognoses.

Although each healthcare organization has different structure and size of clinical notes, a potential AutoML tool will find the best algorithm and settings that provide best accuracy without the need for human interference, which will save funds by reducing the number of clinical coders in the hospitals.

In addition, such tool will assist medical practitioners to better manage their patients and use their valuable time to deliver better outcomes for patients' health. It can also

contribute to improved medical resource management and reduce the burden on national and international medical systems. Moreover, it will help hospitals increase clinical coding team efficiency, reduce coding errors, improve coding quantity and quality, and assist clinical coders through significantly cutting the time needed for processing notes. This means improving the quality and quantity of coded reports since the AutoML model can process thousands of records per hour.

Furthermore, clinical coding AutoML helps junior clinical coders with undeveloped skills or non-medical workers to better identify illnesses without the need to understand medical terms. Moreover, it saves the time needed by machine learning experts to prepare datasets manually and to wait for the training process of one algorithm to finish to try another one. In addition, such AutoML tool can help build more AutoML platforms for medical language processing research in the future.

### *4.3. Machine Learning for Clinical Notes Analysis*

To achieve the aforementioned benefits, an effective AutoML system for clinical notes must be developed. In such a system, the required stages in a typical AutoML platform should be carefully designed and integrated. In the following subsections, we survey how previous works addressed various stages of AutoML for clinical text, but not in an AutoML setting. At the end of each subsection, we discuss the stages from the perspective of developing AutoML for clinical notes, raise some research questions, and propose future directions in developing AutoML for clinical notes analysis.

#### 4.3.1. Preprocessing

Machine learning algorithms run on computers, which cannot understand human languages and can only process structured data. Therefore, it is necessary to transform unstructured clinical notes into structured data so the machine learning algorithm can process it. This transformation process requires applying NLP techniques to quantify clinical notes words and terminologies, which are the features of the medical notes dataset.

To achieve such transformation, clinical notes first need to be cleaned up and prepared in a step that is called preprocessing. Preprocessing is essential in any machine learning processes, but is of great importance when working with text and especially medical text. It removes unnecessary words and group those with the same roots before feature extraction. Removing and grouping these words reduce the number of extracted features in later stages and helps in improving the model's performance and accuracy.

The techniques used for preprocessing text can vary based on the industry, text format, targeted results, and many other factors. To the best of our knowledge, there is no previous work that has implemented preprocessing of clinical notes in an AutoML setting. Therefore, here we survey common techniques that can be integrated in a future AutoML tool for clinical notes.

One of the basic text preparation techniques is word tokenization which splits text into separate words where each word becomes a token. This method allows similar words to be counted instead of processing each word separately. For example, in the sentence "patient reported nausea, vomiting and headaches" each word is a token by itself "patient", "reported", "nausea", "vomiting", "and", and "headaches" [21,71].

Another technique is stop-word removal. A corpus is normally congested with many words that can affect a model's performance and accuracy negatively such as "the", "at", "on", "you", etc. In medical texts, there are more generic words that are repetitive such as "disease", "disorder", "chronic" [72]. Therefore, removing these words from clinical notes can improve performance and accuracy.

Word stemming and lemmatization techniques are converting words into their roots by removing suffixes and returning the words into their dictionary forms. An example is words such as "following", "follower" and "followed" which all become "follow" after stemming. This helps in reducing the number of distinct tokens and improve model performance and accuracy. There exist available libraries that identify stop words and do

the tokenization, stemming and lemmatization such as the natural language toolkit (NLTK) library in Python, and spaCy library for Python and Cython programming languages. For the health sector, some other methods are used to identify words and prepare medical text.

One of the popular methods is regular expressions (Regex), which is a search pattern technique that finds words that match certain expression. For example, in this technique, the token "ing$" means all words that end with "ing", and "go+gle" means all words that contain gogle, google, gooogle, etc. should be grouped together [73]. In addition, regular expression techniques are used to identify and tokenize words such as "diabet[a-z]*". This expression can identify words that are similar to diabetes, such as diabetic and diabetically. Once these words are identified, they can be grouped within the same root, so diabetic and diabetically should count under diabetes. Regex is implemented in many programming languages such as Java, C#, and Python [74,75] and has been used in various research. For example, it was used in [76] to extract information from radiology and other EHR text, where Regex `(?<=\|)\d*/\d*/\d*` was used to find report dates.

Another method to prepare medical texts is developing rule-based algorithms. These algorithms implement specific rules that do not involve computer learning. Examples include stop words removal rules, such as removing the stop-word "and"

$$\text{: if word} = \text{“and” then remove word,} \tag{1}$$

tokenizing word rules, such as grouping "Sz" under the general "seizure" word group,

$$\text{: if word} = \text{“Sz” then “seizure”,} \tag{2}$$

or word correction rules, such as exchanging English spelling with American one,

$$\text{: if word} = \text{“localised” then “localized”.} \tag{3}$$

Other rule-based algorithms can also be developed for identifying term rules, expanding a term to its complete definition rules, word correction rules, etc. [75]. An example is [77], where specific rule-based text processing algorithms such as symbolic approach linguistic rule, to find the relation between medical terminologies, was developed to retrieve information from biomedical ontology texts.

In addition to the above deterministic algorithmic techniques, machine learning has proven to be efficient in text preparation. Processes such as removing stop words, tokenization, and expanding a term to its complete definition, can be performed using machine learning algorithms. For instance, SVM has been used to clean clinical notes from all unnecessary words while improving preparation process performance [75]. Some previous works have developed ML-based libraries to assist in the text preparation process. Examples include the snowball stemmer method that is built-in scikit-learn and weka libraries [78].

Both rule-based techniques and machine learning techniques can be combined to prepare clinical texts [75]. For instance, Pakhomov et al. [79] have used the combined technique. First, they used a rule-based method to identify diagnoses in the clinical notes under processing. This method accurately classified 86% of the notes. The rest, which were unsuccessfully classified, were labeled unknown. Next, they used a neural network algorithm and Naive Bayes on the unknown labeled records. These algorithms classified the unknown records with 91.61% accuracy. This made their proposed hybrid model's accuracy 98.78%.

Preprocessing in AutoML for Clinical Notes

Clinical note preparation is the first step in the AutoML process. By applying the above-mentioned preparation methods, a dataset becomes ready for next steps. Tokenizing, removing stop words, stemming and lemmatizing clinical text are all processes that are applied to clinical text one after the other. The order of this application may change, some steps may not be used, or even some new steps could be added. The techniques used in

each process can also vary. For example, lemmatization can be a rule-based or a machine learning method, or a hybrid one using both rules and ML as in [79]. The choice of which stop-word dictionary is used can vary too. Automating the chosen processes and the order at which these techniques are applied to clinical notes will be the initial step in building an AutoML for diagnoses classification in clinical notes.

In a previous work, Estevez-Velarde et al. [80] have split this stage into six steps. The steps start with removing punctuation symbols and accents and continues with tokenization, combining single words, adding dependency features, stemming and stopwords removal. Even though all these steps are implemented in [80], there was flexibility in removing some of these steps such as removing stopwords and stemming.

Implementing preprocessing in AutoML also depends on the nature of dataset and targeted results. In addition, other factors can affect the steps and techniques to be chosen for this stage. For instance, in [80] some techniques were implemented for working with Spanish language, such as removing the accents. This suggests that the preprocessing is very challenging to be automated due to the variety of techniques, libraries, dictionaries, and even languages.

In Auto-Sklearn, the approach Feurer et al. [81] used was to reduce the preprocessing space by limiting the preprocessing stage to essential techniques only. The main reason they used this approach was to save processing time, because in Auto-Sklearn, the user decides the running time of each model, which could be very limited for trialing all various preprocessing steps. Although Auto-Sklearn is a generic AutoML library that is built for tabular datasets, the same approach can work for AutoML for clinical coding.

The key point here is that a possible AutoML tool for clinical notes analysis should use the above-mentioned preprocessing techniques, which heavily depend on the time limitation of the AutoML tool, the dataset at hand, and the performance required. In this way, lessons learned in previous AutoML implementations applied to other domains should be considered when working with the more challenging clinical notes.

#### 4.3.2. Feature Extraction

Once clinical notes are preprocessed and cleaned, their features need to be extracted and selected. Feature extraction and selection is the process of quantifying words and medical terminologies in the notes. The most-used methods are bag of words (BOW) and term frequency–inverse document frequency (TF-IDF). BOW is counting the repetition of each word in the prepared clinical text independently from other notes in the dataset. On the other hand, TF-IDF considers word appearance in other instances of the dataset text, beside the repetition on the current record. These methods produce high-dimensional sparse features. They also do not respect word order and that could lead to an error in classification of the report. For example, a sentence such as "broke his arm then fainted" has the exact representation as "fainted then broke his arm" if the features are extracted by either BOW or TF-IDF.

TF-IDF [20,82–86] and BOW [18,20,75,87–89] have been extensively used in previous research. They can work on a single word at one time or combine multiple words together. When working with several words, the technique is called n-gram, and it normally uses a combination of a small number of words such as 2-gram or 3-gram. These words are then treated as a single entity to be quantified in BOW or TF-IDF.

Word2vec is another method that is used for feature extraction in clinical notes. It pretrains word vectors using a neural network model and external corpus such as medical information mart for intensive care dataset (MIMIC) and Merriam-Webster medical thesaurus to represent each word in a vector [90].

Word2vec uses two algorithms to build its vectors. The first is continuous bag of words (CBOW) and the other is skip-gram. These techniques have been extensively used in previous research [18,20,74,87,91,92]. Similar to word2vec, global vectors (GloVe) is another word representation method. Although word2vec uses word appearance within local context, GloVe uses word appearance within the whole corpus [90,92–95].

In addition, some other works have developed new techniques or a combination of the above-mentioned algorithms (TF-IDF, BOW, and Word2vec) to extracts features from medical texts. For instance, Weng et al. [71] used BOW, TF-IDF, and word embedding along with SVM, Naive Bayes and convolutional neural networks (CNN) algorithms to classify medical subdomains. They combined BOW features along with unified medical language system (UMLS) concept extracted features. Concept features were extracted using the clinical text analysis and knowledge extraction system (cTAKES) [96], which is used to extract a limited number of concepts. In their model, they used different combinations of features including using BOW features alone, BOW with UMLS concepts limited to five semantic groups, BOW with UMLS concepts limited to 15 semantic groups, and BOW with all UMLS concepts. They also evaluated each feature method solely. The best results were for the method that had a combination of BOW and UMLS concepts limited to five semantic groups. For this, clinicians extracted the most related concepts from the clinical notes. This reduced the number of concepts, which led to better performance and better accuracy.

In early research, there was a need to extract medical terms from clinical corpus in what is called concept extraction or annotation [97]. Now, there are clinical specialized tools that can improve clinical note analysis and identify clinical concepts much better than standard NLP techniques. cTAKES is an open-source system that uses a combination of rules and machine learning algorithms to identify medical terminologies in clinical notes [96]. It prepares text and uses external dictionaries such as UMLS, systematized nomenclature of medicine—clinical terms (Snomed CT), and normalized names and codes for clinical drugs (RxNorm) to identify diseases, symptoms, medications, procedures, and body anatomy [22]. Further dictionaries such as international classification of diseases version 10 (ICD10) or older versions such as ICD9 can be also added to cTAKES. Moreover, clinical concepts can be represented in concept unique identifiers (CUI), which are codes generated by UMLS and can be features of a clinical note dataset. cTAKES can identify concepts even if words do not match, for example, symptoms such as "shortness of breath" and "dyspnea", which both represent the same concept are both represented as CUI number "C0013404". Besides medical annotation, cTAKES can identify term affirmation status, therefore, some phrases can be affirmed, negated or uncertain. Examples of cTAKES affirmation are: "she reports having a cough" is affirmed, "no acute distress" is negated, and "may represent atelectasis" is uncertain. Separating negated CUIs from affirmed ones can improve text classification [15].

The cTaKES system has been extensively used in the literature for clinical notes feature extraction [15,71,98–102]. For instance, Gehrmann et al. [15] used different methodologies for applying machine learning on clinical notes to classify a list of diseases, one disease at a time. After data preparation, they used cTAKES to extract medical concepts. Then used the extracted complete CUIs and applied cTAKES CUI output into BOW and TF-IDF models. They also used 2-gram and 3-gram models to prepare data for the machine learning process.

Another tool that is used for clinical concept extraction is MetaMap that uses NLP and computational linguistic techniques. Similar to cTAKES, MetaMap uses UMLS as a dictionary to extract clinical concepts in a form of CUIs. MetaMap aggregates CUIs to avoid concept duplications [103,104].

Reategui et al. [104] have done a comparison between MetaMap and cTAKES against obesity classification challenge data [105]. Results of the comparison were very close to each other in most disease identifications with cTAKES having a better average result than MetaMap. Suominen et al. [106] have used MetaMap to build an augmented feature list with UMLS CUIs along with BOW. In addition to CUIs, they added parent hypernyms of the CUIs. For example, a main category "Dystonia" was added to represent "Blepharospasm" and "Spasmodic Torticollis" diseases. This technique helped Suominen's team to secure the third place in the computational medicine center's challenge for medical NLP in 2007.

Each model studied has different set of features that depend on the clinical notes used and the targeted diseases. Tools such as cTAKES extract features in form of CUIs or Snomed codes, while BOW can build large number of features based on medical and

non-medical words such as "bed", "cough" or "morning". Table 3 summarizes several feature-extraction methods from literature. The table shows that MetaMap has been only used a few times, while the other methods are of similar popularity in the literature. In addition, the table demonstrates that some of the previous works have combined multiple feature-extraction methods [18,20,71,87].

**Table 3.** Feature-extraction methods for clinical notes.

| Research | Word Weighting | | Word Embedding | | Medical Analytics Tools | |
|---|---|---|---|---|---|---|
| | TF-IDF | BOW | Word2vec | GloVe | cTAKES | MetaMap |
| ML & NLP for clinical notes classification [71] | ✓ | ✓ | ✓ | × | ✓ | × |
| Deep learning evaluation for ICD [20] | ✓ | ✓ | ✓ | × | × | × |
| Clinical text classification [82] | ✓ | × | × | × | × | × |
| Labeling clinical text [83] | ✓ | × | × | × | × | × |
| Identifying alcohol use [84] | ✓ | × | × | × | × | × |
| Automated ICD coding [85] | ✓ | × | × | × | × | × |
| Indexing biomedical literature [86] | ✓ | × | × | × | × | × |
| Multi-label classification [18] | × | ✓ | ✓ | × | × | × |
| Mental status automated detection [87] | × | ✓ | ✓ | × | × | × |
| ML & NLP for clinical coding [75] | × | ✓ | × | × | × | × |
| ML approach on encoding [89] | × | ✓ | × | × | × | × |
| Feature selection from BOW [88] | × | ✓ | × | × | × | × |
| Medication extraction [74] | × | × | ✓ | × | × | × |
| ICD encoding using deep learning [91] | × | × | ✓ | × | × | × |
| ML models for clinical coding [92] | × | × | ✓ | ✓ | × | × |
| Medical notes classification [93] | × | × | × | ✓ | × | × |
| Embeddings learning from medical notes [94] | × | × | × | ✓ | × | × |
| AI for classifying diagnosis [95] | × | × | × | ✓ | × | × |
| Oncologt patients pre-screening [99] | × | × | × | × | ✓ | × |
| Rules and deep learning comparison [15] | × | × | × | × | ✓ | × |
| Ontology feature engineering [100] | × | × | × | × | ✓ | × |
| Medical notes knowledge extraction [98] | × | × | × | × | ✓ | × |
| Cancer information text mining [101] | × | × | × | × | ✓ | × |
| NLP of health text [102] | × | × | × | × | ✓ | × |
| Drugs indications extraction [103] | × | × | × | × | × | ✓ |
| Radiology reports codes assignment [106] | × | × | × | × | × | ✓ |

Feature Extraction in AutoML for Clinical Notes

The main methods studied here are BOW, TF-IDF, Word2Vec, GloVe, cTAKES and MetaMap, which are critical NLP methods for feature extraction from clinical notes. These can all be investigated in an AutoML platform at its feature-extraction stage.

However, to the best of our knowledge, there is not much previous published research on the implementation of AutoML for NLP. Most of the previous research for text analysis compared feature-extraction methods manually through evaluating each method in a separate model and comparing the results.

Looking at natural language classification models applied in other domains, they usually use word weighing and embedding methods as their main feature-extraction

techniques, which are not enough for clinical notes analysis. An acceptable AutoML tool for medical notes requires trying various feature-extraction methods in complete models to identify the best. This is not the only challenge, each one of these methods has its own parameters that increases the complexity of identifying the best method. For instance, BOW and TF-IDF can work on single words, 2-gram, or 3-gram options. Similarly, Word2Vec and GloVe can be trained on MIMIC database, Merriam-Webster medical thesaurus or any other medical datasets. These makes feature extraction a very challenging step for medical text, which requires extensive future research efforts.

One of the approaches to consider is to use cTAKES, which has proved efficient in clinical notes features extraction. It has been used in many previous studies and provides a list of medical concepts such as symptoms, procedures, anatomy, diseases, and medications, which are focused features and have valuable information. On top of that, cTAKES does not require the preprocessing stage. On the other hand, it is computationally expensive as it uses rules-based and ML-based algorithms to extract medical concepts. Moreover, cTAKES is a complete solution that requires installation and configuration by an expert and cannot be embedded within a programming language such as Python or R, or be easily integrated in a customized AutoML tool.

Looking at all these methods and tools raises several important questions to be answered in future research. These include but are not limited to

- Which feature-extraction method or combination of methods should be used in an AutoML for clinical notes?
- Which methods should be used to compare and decide the best feature-extraction techniques?
- Can cTAKES be easily integrated in an AutoML tool for clinical notes analysis? What are the hurdles?

These questions could have multiple answers, such as using a baseline model to compare feature-extraction methods results, or using an exhaustive approach by testing all methods on a small subset, or applying and comparing all the methods on the complete dataset, which will be time consuming but will most likely be more accurate. To address these questions, much further research should be undertaken in this unexplored research domain.

#### 4.3.3. Feature Selection

NLP feature-extraction methods, and concept extraction tools generate many features. The number of generated features can reach thousands to hundreds of thousands depending on the dataset and clinical text size. The high-dimensional features affect machine learning model's accuracy and performance, and can cause overfitting too [100,107]. Therefore, it is essential to reduce the number of features and select features that carry more useful information for the learning process.

Feature selection methods are algorithms that work together to select a subset of features to improve model accuracy or performance. The ones used in clinical text analysis can be divided into filter and wrapper methods [108]. The filter method ranks features using algorithms such as Chi square ($\chi^2$) [109–111], information gain (IG) [111,112], mutual information (MI) [113], symmetrical uncertainty (SU) [21,114], etc. Next, a subset of these ranked features is selected by increasing the correlation with the targeted classes and reducing redundancy between selected features. Different algorithms are used in this process such as genetic algorithm (GA) [21], fast correlation-based filter (FCBF) [108,115,116], and algorithms that use forward and backward search [117].

FCBF is a filter method that ranks features by symmetrical uncertainty (SU) with the targeted value $y$, then, sequentially remove redundant features by calculating SU for all available features in the dataset and comparing them to SU for the feature and targeted value, as illustrated in Figure 4. It is suitable for high-dimensionality datasets as it can reduce the number of selected features and improve model accuracy. The complexity of

this algorithm is $O(MNlog(N))$ where $M$ is the number of dataset instances and $N$ is the number of dataset features [118,119].

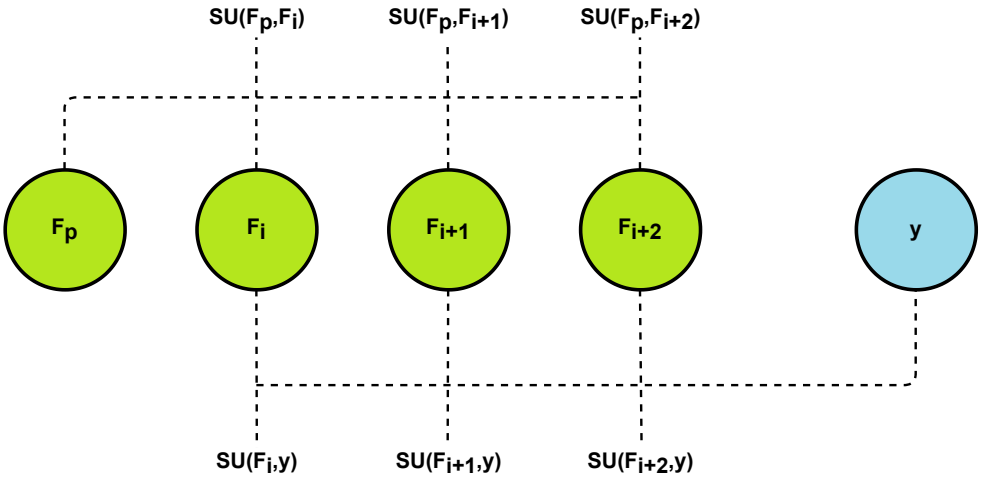


**Figure 4.** FCBF compares symmetrical uncertainty $SU(F_p, F_i)$ of each feature $F_i$ with the first feature $F_p$. If $US(F_p, F_i) \geqslant US(F_i, y)$, $F_i$ gets removed.

Genetic algorithm (GA) is another technique for feature selection. As its name suggests, it operates in a similar way to genes and chromosomes; it passes features from a parent subset to an offspring subset. GA selects a set of random subsets, then evaluates each one of these subsets using a prediction algorithm. Applying the concept of "survival of the fittest", subsets with high accuracy are "mated" to pass their features to a new offspring. Randomly, "mutations" or small changes to the selected features can happen by adding features from outside the parent subsets. The process stops when the subset's evaluation improvement reaches a certain target [120].

Another feature selection technique is the Wrapper method that uses the prediction algorithm (such as KNN or SVM) of the used machine learning model to evaluate subsets. The wrapper method uses different techniques to select subsets to be evaluated [121]. Figure 5 shows how the wrapper method works.

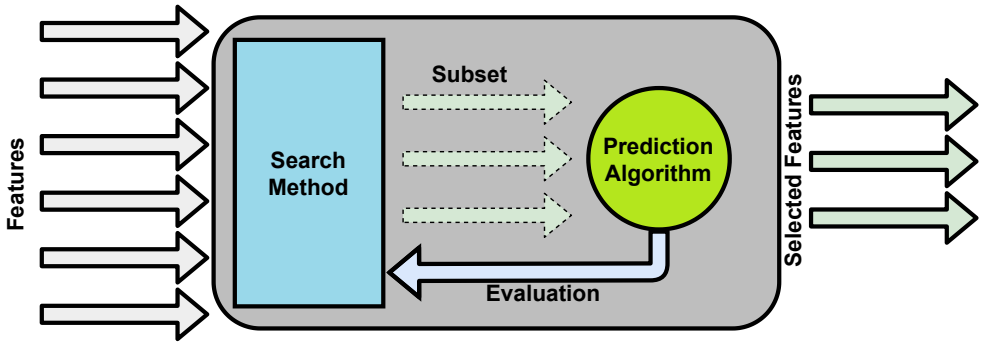


**Figure 5.** The wrapper feature selection method. Here, the search method evaluates the feature subsets using the prediction algorithm and then selects the subset with the best result.

One of the wrapper methods is leave one out (LOO) cross validation [122]. LOO method splits datasets with $N$ number of records into $N$ subsets, each with $N-1$ records to train the prediction algorithm and one record to test and evaluate the model. Although this method requires lots of resources and computational power, it achieves the best trained model as it uses all the $N$ records in the training process. In addition to the above-mentioned methods, there are other techniques such as bootstrap resampling validation [123] and kernel entropy inference validation [124], for feature selection.

Soguero-Ruiz et al. [88] aimed to detect anastomosis leakage (AL) that could occur after colorectal cancer surgery via analyzing EHR, which included clinical notes in text format. Their main focus was reducing dimensionality by selecting features from clinical notes using the BOW and SVM algorithm. They tested their method using three different validation methods: LOO, bootstrap resampling validation, and kernel entropy inference validation.

They also used three evaluation methods to test the outcome of each feature selection method. The comparison between the three feature selection methods showed that bootstrap resampling method has the best performance and reaches the smallest number of selected features, i.e., 196. LOO method resulted in a significantly higher number of selected features (6896), while bootstrap resampling and kernel entropy reached 212 features.

Garla et al. [100] found that cTAKES generates many CUIs, and not all these CUIs are related to their targeted task for multiple obesity-related disease classification. They, therefore, used automated isolation hotspot passage (AutoHP) that improved selection of only features that were useful for their application. AutoHP uses IG algorithm to rank features and uses BOW to quantify text around highly ranked features.

Feature Selection in AutoML for Clinical Notes

Table 4 shows a list of previous papers that investigated feature selection methods for clinical notes. This table confirms the no free lunch theory [125], which states that there is no one algorithm that can provide best results for all datasets. A naive exhaustive approach in AutoML may include testing all feature selection methods to find the best algorithm. However, this is computationally expensive especially that methods such as LOO take up lots of resources. Therefore, limiting feature selection methods to the ones that have proved useful for different applications can reduce resource consumption and improve performance compared to the naive exhaustive approach. Similar to feature extraction, in an AutoML setting, other approaches such as using a baseline model or naive exhaustive search on smaller subsets can be used for feature selection. Furthermore, an AutoML tool may investigate and use previously proposed feature selection techniques such as those in [37,126].

In [37], Luo et al. have proposed some new techniques for feature selection. One of them is to use multiple feature selection methods simultaneously, and remove features that do not have positive effects on target classification. Another technique is to set a ranking criteria for feature selection methods. For this, some selection algorithms such as $\chi^2$, IG, and MI, which are considered feature ranking methods, are used. Then a threshold is set for the number of test cases that each feature selection method should use before ranking features. In an AutoML scenario, the techniques proposed in [37] may be implemented and used. However, algorithm optimization is necessary, otherwise, the AutoML feature selection method may not be computationally viable.

In [126], Escalante et al. expanded the particle swarm optimization (PSO) [127] method to search in a combined pool of preprocessing methods, feature selection methods, learning algorithms and hyperparameter optimization methods. Their proposed particle swarm model selection benefited from the simplicity and light computation weight of PSO to find the combination of methods at each machine learning stage that provides the highest model accuracy. This method can be investigated to find out if it is suitable to be implemented in an AutoML tool for clinical notes analysis.

Another interesting future study can compare the methods proposed in [37,126]. Such study will need to perform an extensive test for these methods on clinical notes datasets within an AutoML platform to identify which one works better in high-dimensionality dataset and can improve AutoML accuracy and performance.

**Table 4.** Feature selection methods for clinical notes.

| Research | Filter | | | | | | | Wrapper | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\chi^2$ | IG | MI | SU | GA | FCBF | Forward & Backward Search | LOO | Bootstrap Resampling | Kernel Entropy Inference |
| Clinical coding survey [110] | ✓ | × | × | × | × | × | × | × | × | × |
| Autopsy reports classification [111] | ✓ | ✓ | × | × | × | × | × | × | × | × |
| Clinical coding feature selection [112] | × | ✓ | × | × | × | × | × | × | × | × |
| Ontology feature engineering [100] | × | ✓ | × | × | × | × | × | × | × | × |
| Assigning clinical codes [113] | × | × | ✓ | × | × | × | × | × | × | × |
| Clinical coding with EHR data [114] | ✓ | ✓ | × | ✓ | × | ✓ | × | × | × | × |
| Clinical narrative [21] | ✓ | ✓ | × | ✓ | ✓ | × | × | × | × | × |
| Health data interoperability [108] | × | × | × | × | × | ✓ | × | × | × | × |
| ML diseases profiling [117] | × | × | × | × | × | × | ✓ | × | × | × |
| Feature selection from BOW [88] | × | × | × | × | × | × | × | ✓ | ✓ | ✓ |

#### 4.3.4. Algorithms Selection and Optimization

Once the dataset is prepared, the features are extracted, and a feature set is selected, the next step is to select a machine learning algorithm and to train it on the available dataset, from which features are extracted and selected. The dataset will also be used to evaluate how accurate the model is and how well it performs. This performance can be a good indication on how the trained model may perform on a new dataset. Below, we review several previous works that used various ML algorithms and how their hyperparameters got optimized for clinical notes processing.

Gehrmann et al. [15] applied multiple algorithms including convolutional neural networks (CNN), RF, and logistic regression (LR) on a binary classification task for ten disease diagnoses. The dataset feature engineering was performed using cTAKES and n-gram methods as mentioned earlier. Both 2-gram and 3-gram were used with LR for feature extraction. CNN hyperparameters were chosen and optimized manually. The CNN algorithm achieved the best results for all classified diagnoses. It also provided the best positive predictive value (PPV) for 50% of diagnosis classification models.

Nigam [16] mainly focused on recurrent neural network (RNN) algorithms. She compared several algorithms including LR and neural network models such as long short-term memory networks (LSTM), and gated recurrent units (GRU) used with BOW for multi-label diagnoses on a large dataset for more than 30,000 patients. Nigam's research focused on both accuracy and performance. She used two datasets with different numbers of diagnostics. The first dataset was limited to ten diagnoses and the other dataset had a hundred diagnoses. GRUs showed the best result when applied to the dataset with ten diagnoses, while RNNs showed the best performance for the dataset with 100 diagnoses.

Venkataraman et al. studied three algorithms [17], i.e., DT, RF, and LSTM, which processed clinical text sequentially. Their study used two models, one targeting top level ICD9 codes such as mental disorders and neoplasms, while the second targeted diagnosis codes.

They extracted features from clinical text in two ways using MetaMap and GloVe methods. They tested different training and validation datasets, along with several feature-extraction and target options, which resulted in 28 different combinations of classification models. Out of these 28 models, LSTM provided the best F1 score results in 23 of them. RF had the best F1 score in four models and DT scored the best F1 in one model. To achieve the best F1 scores, LSTM was tested with different sets of hyperparameters. This shows the importance of using AutoML in hyperparameter optimization, as these parameters can affect results and optimizing them can lead to better performance. Moreover, the effort of testing multiple sets can be preserved by using AutoML hyperparameter optimization techniques.

Huang et al. [20] evaluated multiple algorithms to diagnose discharge summaries for the top 10 and top 50 ICD codes and top 10 and 50 ICD categories. After tokenizing text, they used TF-IDF and Word2vec methods to extract features. Targeted diagnoses in their dataset were converted into binary values; if a discharge summary is classified as "diabetes" and "pneumonia", then their model will treat them as 2 targeted values: diabetes is 1 in the first case and pneumonia is 1 in the second.

This conversion made it possible to use LR and RF algorithms, which use a single classification per discharge summary per model. On the other hand, they used neural network algorithms, which do not need to be trained for each diagnosis separately. Neural networks can classify multiple diagnoses in one model. The algorithms they used were conventional feed-forward neural networks (FNN), CNN, the basic version of RNN, and other versions of RNN, i.e., LSTM, and GRU. There were many combinations of parameter options the authors had to test to get the best model of each algorithm.

For example, for LR, they had to test multiple iterations between the range of 5 and 100 to get the iteration that provided them with the best results. RF has multiple parameters to be configured too. They configured tree depth in a range between 5 and 30. For FNN, they tried different neuron sizes, and different activation functions such as ReLU, and sigmoid. For CNN, the number of layers were changed between three and ten, for different layer sizes. For RNN, layer sizes of 64, 128, and 256 were tried. All these options were tested manually, and the results of each case were compared to get the best configuration that gives the best F1 score model. Best results were not consistent in terms of algorithm, feature-extraction method, and parameter configuration. For top 10 ICD codes, GRU algorithm with self-trained Word2vec feature extraction provided the best F1 score result of 0.696. Similarly, for the top 10 ICD categories GRU had the best F1 score of 0.723. Logistic regression had the best F1 score for top 50 ICD codes and categories of 0.366 and 0.430, respectively.

This study shows that for the same dataset different results can be achieved. It shows that there is not a single method of feature extraction, feature selection, algorithm selection and hyperparameter optimization that work for all trials and the best combinations of steps should be found.

This simply cannot be performed manually, as it requires significant time and efforts. This further justifies the need for an AutoML platform that can automate the process of applying different combinations of these methods for similar clinical notes as those processed in [20]. Therefore, an opportunity exists to automate this process so computers can identify best methods that provide the most accurate diagnoses for a dataset.

Pineda et al. [128] compared seven different algorithms to detect influenza from clinical notes. They compared results of Naive Bayes, LR, Bayesian network, efficient Bayesian multivariate classification, RF, SVM, and artificial neural network algorithms. Their results were close for most of the algorithms. NB, LR, and ANN showed the best area under curve (AUC) of 0.93. However, NB showed the best performance with five milliseconds required to perform the detection, while this time was around four and half minutes for ANN. NB algorithm had the best AUC results even when the method of filling missing values in the test dataset was changed. In this case, the other algorithms performed randomly.

There are other techniques in the literature that were used to extract diagnoses from clinical notes. For instance, Chen [129] used BERT [130] transformer technique in classifying clinical notes based on ICD9 codes on MIMIC dataset and compared it with other research results based on CNN, GRU and convolutional attention for multi-label classification (CAML) [23]. The comparison was not in favor of BERT as F1 score and AUC of BERT were lower than CNN and CAML.

Algorithm Selection and Hyperparameter Optimization in AutoML for Clinical Notes

The algorithms that Gehrmann et al. [15], Nigam [16], Venkataraman et al. [17], Huang et al. [20], and Pineda et al. [128] used have been extensively employed in the literature as shown in Table 5. The table shows a list of the most popular algorithms in clinical note processing. As demonstrated, SVM, CNN, RF, and LR are the most-used algorithms for patient diagnoses through analyzing clinical notes. Other algorithms that are used and showed good results include LSTM and DT.

**Table 5.** ML algorithms used for clinical notes.

| Research | Statistical Algorithms | | | | | | Neural Networks | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **RF** | **LR** | **DT** | **SVM** | **NB** | **KNN** | **CNN** | **RNN** | **LSTM** | **GRU** |
| Medical notes classification [93] | × | ✓ | × | × | × | × | ✓ | × | ✓ | × |
| Medication extraction [74] | × | × | × | ✓ | × | × | ✓ | ✓ | ✓ | × |
| Automated ICD coding [85] | × | × | ✓ | × | × | × | ✓ | × | × | × |
| Deep transfer learning for ICD coding [131] | × | × | × | ✓ | × | × | ✓ | × | × | × |
| ICD coding via deep learning [132] | × | × | × | ✓ | × | × | ✓ | × | × | × |
| Medical codes explainable prediction [23] | × | ✓ | × | ✓ | × | × | ✓ | × | × | ✓ |
| ML models for clinical coding [92] | ✓ | ✓ | ✓ | ✓ | × | × | ✓ | × | ✓ | × |
| Deep learning evaluation for ICD [20] | ✓ | ✓ | × | × | × | × | ✓ | ✓ | ✓ | ✓ |
| Rules and deep learning comparison [15] | ✓ | ✓ | × | × | × | × | ✓ | × | × | × |
| AI for classifying diagnosis [95] | ✓ | × | × | ✓ | × | × | ✓ | × | × | × |
| Mental status automated detection [87] | ✓ | × | ✓ | ✓ | ✓ | × | ✓ | × | × | × |
| Automated text classification [17] | ✓ | × | ✓ | × | × | × | × | × | ✓ | × |
| ML for ICD term encoding [89] | ✓ | × | × | × | × | × | × | × | × | × |
| Eye disease classification with ML [133] | ✓ | × | ✓ | × | ✓ | × | × | × | × | × |
| Autopsy reports classification [111] | ✓ | × | ✓ | ✓ | ✓ | ✓ | × | × | × | × |
| ML classifiers comparison [128] | ✓ | ✓ | × | ✓ | ✓ | × | × | × | × | × |
| Crohn's case definition using NLP [134] | × | ✓ | × | × | × | × | × | × | × | × |
| Multi-label classification [18] | × | ✓ | × | × | × | ✓ | × | × | × | × |
| Privacy-preserving data enrichment [135] | × | ✓ | × | × | × | × | × | × | × | × |
| Multi-label classification with DL [16] | × | ✓ | × | × | × | × | × | ✓ | ✓ | ✓ |
| Rule-based ICD coding [136] | × | × | ✓ | × | × | × | × | × | × | × |
| Diagnosis code assignment [35] | × | × | × | ✓ | × | × | × | × | × | × |
| Ontology feature engineering [100] | × | × | × | ✓ | × | × | × | × | × | × |
| Matching codes to diagnoses [19] | × | × | × | ✓ | × | × | × | × | × | × |
| Medical notes knowledge extraction [98] | × | × | × | ✓ | ✓ | ✓ | × | × | × | × |
| ML for ICD encoding [78] | × | × | × | × | ✓ | × | × | × | × | × |
| Determining modification of diagnoses [79] | × | × | × | × | ✓ | × | × | × | × | × |

Testing all these algorithms manually would require data science expertise along with a large pool of hyperparameter sets to be tried one by one to get the best results for a dataset. Applying AutoML algorithm selection and hyperparameter optimization techniques help in finding the best model with the most efficient results. However, the challenge at this stage is to find hyperparameters that give the highest accuracy for each one of the considered learning algorithms. This problem has been studied in many existing AutoML platforms such as Auto-Sklearn [7], and Auto-WEKA [31]. Both these platforms have used Bayesian optimization for hyperparameters because it has shown the best results and resulted in the highest performance.

Although Auto-Sklearn and Auto-WEKA are statistical algorithm-based AutoML platforms, Auto-Keras [137] which is a neural network-based platform also uses Bayesian optimization. This is to lead the AutoML network morphism based on NAS model. Auto-Keras brings benefits from hardware perspective too, because it uses both CPUs and GPUs to improve its performance.

In another paper [137], the authors proved that random search can outperform Bayesian optimization, with the price of doubling the processing time. An interesting future research direction is to investigate an AutoML model that allows the use of both Bayesian optimization and random search. In such as tool, if the priority is better performance, random search can be used, while if speed is the priority, Bayesian optimization will be preferred.

#### 4.3.5. Targets and Evaluations

In any machine learning development task, after the features are extracted and selected, the ML algorithm is designed and optimized based on defined evaluation metrics. For clinical notes they are usually classified for identification of a disease, symptom, or behavior. This classification, similar to other machine learning areas, can be categorized into three different domains including binary, multi-class, and multi-label multi-class classifications.

Binary classification in the context of clinical notes is to confirm, for example, if a patient has a disease or not. This is the simplest form of classification, where the targeted value is either one or zero. This form of classification can be used to identify if a patient has a certain disease such as diabetes or not, or have a certain behavior such as smoking or not. In binary classification, the only options available to predict are limited to two (one and zero).

Multi-class classification is where the ML prediction can be one class in a list of three or more classes of diseases, symptoms, etc. The classes can be a list of diseases such as ICD9 codes, for examples 297.1 Paranoia or 775.4 Hypocalcemia, or a list of blood types O, AB, A, and B.

In multi-label multi-class classification, there are multiple targeted classes for each instance, and the number of targeted values can differ from one record to another. For example, the prescription prediction for one patient can be citalopram, and fluoxetine only, while another patient may get vilazodone, benzonatate, acetaminophen, and citalopram. The variable number of targets requires some techniques such as converting the multi-label targets into multi-class targets by combining all targets into one target or transforming targets into binary targets [138].

Regardless of the classification problem at hand, machine learning algorithms are evaluated based on standard evaluation metrics that provide a measure of the performance of the developed algorithm. Similar to other machine learning domains, the standard evaluation metrics are defined based on true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values [139]. These values are used to measure different performance metrics, as described in [139]. One of the main metrics when evaluating clinical NLP systems is

$$Recall = TP/(TP + FN), \quad (4)$$

that shows the model sensitivity, which is the percentage of truly identified positives from the actual positives. Another metric is

$$Precision = TP/(TP + FP), \tag{5}$$

that shows the percentage of truly identified positives to all identified positives. The next metric is

$$F1 = 2(Precision.Recall)/(Precision + Recall), \tag{6}$$

which is a measure that uses both recall and precision to calculate the model's accuracy.

$$Specificity = TN/(TN + FP), \tag{7}$$

shows the percentage of truly identified negatives to all identified negatives. Another important metric is

$$Error = (FP + FN)/(TP + FP + FN), \tag{8}$$

which is the percentage of all wrongly identified values of all classified values except true negatives. Finally,

$$Accuracy = (TP + TN)/(TP + FP + FN + TN), \tag{9}$$

is the percentage of all truly identified values to all values. Another metric that was not covered in [139], yet has been extensively used in the literature [20,71,85,87,91,95,134,135,140] is the area under the receiver operating characteristic curve (AUC). This measure illustrates the relationship between TP rate, which is defined as recall shown in Equation (4), and FP rate, which is defined as,

$$FPR = FP/(FP + TN), \tag{10}$$

that shows the percentage of all wrongly identified positive values of all falsely classified cases and true negatives. AUC, therefore, is representative of a curve in the area between [0, 1] and [1, 0]. The area under this curve therefore cannot exceed 1. A higher AUC number means a higher model accuracy. Below, we survey some of the previous works on clinical notes prediction and explain how they have used the above-mentioned classification methods and performance metrics in their ML model evaluations.

Gehrmann et al. [15] used binary classification to predict depression from their medical notes dataset. They compared the results of seven different algorithms and feature engineering methods. They used the MIMIC dataset to build a smaller dataset of 1610 clinical notes including 460 depression cases. The dataset was divided into three parts, 70% to train models, 10% for validation and the remaining 20% for testing the models. To evaluate the studied models, they used recall, precision, and F1 score measures. The most accurate precision they got was 91%, best recall was 76%, and best F1 score was 83% all using the CNN model. They then repeated the same process for other diseases including advanced metastatic cancer, advanced heart disease, advanced lung disease, chronic neurological dystrophies, chronic pain, alcohol abuse, substance abuse, obesity, and psychiatric disorder.

Since the main objective of [15] was to predict the presence of a disease from processing the available medical notes, binary classification is the obvious choice. However, should the authors decided to predict if a patient has more than one disease, they should have used multi-class and/or multi-label classification. Regarding the performance metrics used, usually recall and precision are used for binary classification tasks, but they may not represent a true picture of the model performance. Therefore, it is recommended to show F1 score, which better represents how the model performs in both detecting all the instances of the disease and how precise it is in detecting only the disease and not returning false positives.

In another study [141] cTAKES is used in conjunction with a hybrid machine learning and rule-based method to improve smoking status identification accuracy. This status is

defined as either of the following classes: past smoker, current smoker, smoker, non-smoker, and unknown status. A combination of 3 models is built to first identify if it is known that the patient is a smoker or not. If it is known that the patient is a smoker, the second model identifies if the patient is a smoker or non-smoker, while the third model identifies the smoking status, i.e., if the patient is a past smoker, a current smoker, or a smoker. The first model is a rule-based model that checks for certain words such as nicotine, cigarette. If they were not detected then the note will be classified as "unknown". The second model is rule-based too where it tries to find a negation for the keywords that were found in the first model, e.g., "patient does not smoke" or "non-smoker". The third model is an SVM model that extracts smoking words from clinical notes. The F1 score, precision and recall are used to evaluate the proposed hybrid model. The micro average accuracy reported is 0.967 for all three evaluation methods. Since this work is classifying patients into more than two categories, it is a multi-class classification problem. It uses more than one model to classify the clinical notes in a hierarchical manner. The performance measures reported include F1, which represents the model is performing well.

In [19], the author used different approaches to classify clinical notes. She studied clinical notes in the Bulgarian language and built a model that diagnosed these reports. She used ICD-10 codes for diagnosing discharge summaries of 1300 records, focusing on endocrine and metabolic diseases that had codes starting with the letter "E", but some of these diseases started with other letters, mostly "D", "G", "H", "I", "K", "M", and "N". She prepared text using NLP methods such as tokenization, abbreviation expansion, feature extraction, etc. Then she fed NLP output features into SVM. SVM then predicts codes as binary and gives each code a rank. A method called winning strategy was then used to select the code with the highest rank to be the predicted diagnosis. Boytcheva used three common evaluation metrics. The reported results were 74.68%, 97.3%, and 84.5% for recall, precision, and F1 score, respectively, for all endocrine and metabolic diseases, and an F1 score of 81.53% for the "E" coded diseases and the ones in the common clusters "D", "G", "H", "I", "K", "M", and "N".

In another multi-class classification study for clinical notes [82], the authors that used SVM, RF, and CNN and evaluated their results using F1 scores. Furthermore, [71] converted the multi-class model into a binary model and used AUC for evaluation.

In a multi-label classification study for clinical coding [18], classifying medical text can result in multiple coding. These codes could be unrelated to each other or correlated such as diabetes and hypertension. Often when a patient has diabetes, they have hypertension too, but still they can have one without the other. [18] studied different approaches for their multi-label classification problem. The first approach was binary relevance, where each code is compared to all other codes. To resolve the aforementioned coding correlation, once any clinical code has been classified, it becomes a new feature to predict the next clinical code. This approach is called classifier chain. Using multiple classifier chains in random orders of clinical codes is another approach that is called ensemble of classifier chains. A multi-label version of KNN algorithm (MLKNN) can identify a set of labels that are near targeted records. Neural networks can be used for multi-label classification, too. In [18], an open-source tool named MEKA [142] was used for neural network-based multi-label classification.

F1 score was used to compare different classification methodologies. Using ensemble of classifier chains with logistic regression (ECC-LR) gave best F1 score in most of the tested cases. ICD codes were split into 18 groups and ECC-LR resulted in the best F1 score in 12 groups. By adjusting the used algorithms and hyperparameters, the best F1 score varied between 41% and 93.3% based on ICD group.

In another multi-label classification study, Nigam [16] diagnosed ICD9 codes through analyzing clinical notes from the MIMIC dataset. Each clinical note can have multiple diagnoses and therefore multiple ICD codes. The dataset used had 6985 distinct ICD codes with an average of 14 diagnoses for each patient, therefore, 2 sub-datasets were created from the original dataset. The first had the top 100 codes only, while the second include the

top 10. Nigam used recall, precision and F1 scores to evaluate results for each of the used algorithm in this study. The algorithms used included LR, NN, RNN, LSTM and GRU. GRU achieved the best F1 score of 42.03% for the top 10 codes dataset, and RNN had the best F1 score of 24.39% for the top 100 ICD codes.

In addition to the above, specificity has also been used as an evaluation metric in clinical notes processing [134]. However, it is mainly used in clinical annotation [143], and in structured medical data forecasting [144].

### Evaluation Metrics in AutoML for Clinical Notes

Most of the previous research have used the above-mentioned evaluation methods as listed in Table 6. The table depicts that F1 score has been extensively used because it covers both recall and precision and can give a better reflection of performance.

**Table 6.** Evaluation methods used in clinical research.

| Research | Recall | Precision | F1 Score | Accuracy | AUC |
|---|---|---|---|---|---|
| Multi-label classification with DL [16] | ✓ | ✓ | ✓ | × | × |
| Automatic recognition of disorders [145] | ✓ | ✓ | ✓ | × | × |
| Diagnosis code assignment [35] | ✓ | ✓ | ✓ | × | × |
| Rules and deep learning comparison [15] | ✓ | ✓ | ✓ | × | × |
| Deep learning evaluation for ICD [20] | ✓ | ✓ | ✓ | ✓ | × |
| Crohn's case definition using NLP [134] | ✓ | ✓ | × | × | × |
| Drug side effect extraction [146] | ✓ | ✓ | ✓ | × | × |
| Genetic studies informatics leveraging [147] | ✓ | ✓ | ✓ | × | × |
| Smoking status classification [141] | ✓ | ✓ | ✓ | × | × |
| Ontology feature engineering [100] | × | × | ✓ | × | × |
| Multi-label classification [18] | × | × | ✓ | × | × |
| Medication extraction [74] | × | × | ✓ | × | × |
| Matching codes to diagnoses [19] | ✓ | ✓ | ✓ | × | × |
| Automated text classification [17] | × | × | ✓ | × | × |
| Clinical text classification [82] | ✓ | ✓ | ✓ | × | × |
| AI for classifying diagnosis [95] | × | × | ✓ | × | ✓ |
| Automated ICD coding [85] | × | × | ✓ | × | ✓ |
| Suicide attempts prediction [148] | ✓ | ✓ | × | × | ✓ |
| Rule-based ICD coding [136] | ✓ | ✓ | ✓ | × | × |
| Radiology reports codes assignment [106] | × | × | ✓ | × | × |
| ICD coding via deep learning [132] | ✓ | ✓ | ✓ | × | × |
| Eye disease classification with ML [133] | ✓ | ✓ | ✓ | ✓ | × |
| Medical codes explainable prediction [23] | × | × | ✓ | × | ✓ |
| ML classifiers comparison [128] | ✓ | ✓ | × | ✓ | ✓ |
| Symptom extraction [149] | ✓ | ✓ | ✓ | × | × |
| Medical problems extraction [139] | ✓ | ✓ | ✓ | ✓ | × |
| ML models for clinical coding [92] | ✓ | ✓ | ✓ | × | × |
| Disease name extraction [150] | ✓ | ✓ | ✓ | × | × |
| Autopsy reports classification [111] | ✓ | ✓ | ✓ | ✓ | ✓ |

In the context of AutoML, Auto-Sklearn, Auto-WEKA, and Auto-Keras have mainly used error rate and accuracy as evaluation methods in their decision of the best machine learning model [7] in ML competitions such as Chalearn [151]. This shows that error rate and accuracy should be among the initial evaluation metrics that are investigated for various new AutoML platforms including those for clinical notes.

In health-related AutoML platforms such as JADBIO and AutoPrognosis, AUC is used to evaluate the models for multi-class classification [13,25]. In an AutoML tool for clinical notes, since the focus will be mostly on multi-label multi-class classification, F1, and AUC seem more suitable to evaluate related models and should be evaluated at the first steps.

The good news is that evaluation methods are very light in terms of their required computational power. Hence, in an AutoML platform, various methods can be used to select the one that provides the most reliable outcome.

## 5. Conclusions

To enhance patient outcomes and improve the healthcare industry, new techniques and technologies are being developed continuously. One of these technologies, which has shown great promise in the healthcare domain is ML. However, ML usually requires human knowledge in its training process and depends on the expertise of the human designers to achieve good performance. This heavy dependence to humans results in lower adoption in the healthcare industry despite ML's great potential to improve patient outcome and save their times and money while reducing the load on the medical system. One approach that may help in reducing human's involvement in ML is automating the process of ML design and learning. This automatic machine learning development is called AutoML, which is an emerging technology that has already shown great promise in the healthcare domain.

In this paper, we surveyed previous works on AutoML in healthcare. We then discussed that AutoML has not been used in analyzing clinical notes, which contain critical information about patient, but cannot be processed automatically and take a significant time to process by humans. To advance the knowledge in the field of AutoML for clinical notes, we then surveyed the literature on ML works for processing clinical notes. In doing so, we analyzed the literature from an AutoML development perspective and discussed challenges and opportunities it brings.

We conclude that for an AutoML platform to be developed for clinical notes, several important research questions should be addressed, and several hurdles must be overcome. We hope that this paper serves the ML- and AutoML-related healthcare industry and researchers in developing a powerful tool that can improve the quality of life for humanity by significantly enhancing patient outcomes through better diagnoses, reduced costs, and shortened treatment time.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** M.R.A. acknowledges the support of a JCU Rising Star ECR fellowship.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References


1. Azghadi, M.R.; Lammie, C.; Eshraghian, J.K.; Payvand, M.; Donati, E.; Linares-Barranco, B.; Indiveri, G. Hardware Implementation of Deep Network Accelerators Towards Healthcare and Biomedical Applications. *IEEE Trans. Biomed. Circuits Syst.* **2020**, *14*, *6*, 1138–1159.
2. Rong, G.; Mendez, A.; Assi, E.B.; Zhao, B.; Sawan, M. Artificial Intelligence in Healthcare: Review and Prediction Case Studies. *Engineering* **2020**, *6*, 291–301.
3. Beam, A.L.; Kohane, I.S. Big data and machine learning in health care. *JAMA* **2018**, *319*, 1317–1318.
4. Li, J.P.; Haq, A.U.; Din, S.U.; Khan, J.; Khan, A.; Saboor, A. Heart Disease Identification Method Using Machine Learning Classification in E-Healthcare. *IEEE Access* **2020**, *8*, 107562–107582.
5. Leite, A.F.; Vasconcelos, K.d.F.; Willems, H.; Jacobs, R. Radiomics and machine learning in oral healthcare. *Proteom. Clin. Appl.* **2020**, *14*, 1900040.
6. Esteva, A.; Robicquet, A.; Ramsundar, B.; Kuleshov, V.; DePristo, M.; Chou, K.; Cui, C.; Corrado, G.; Thrun, S.; Dean, J. A guide to deep learning in healthcare. *Nat. Med.* **2019**, *25*, 24–29.
7. Feurer, M.; Klein, A.; Eggensperger, K.; Springenberg, J.; Blum, M.; Hutter, F. Efficient and robust automated machine learning. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 2962–2970.
8. Hutter, F.; Kotthoff, L.; Vanschoren, J. *Automated Machine Learning: Methods, Systems, Challenges*; Springer: Cham, Switzerland, 2019.
9. Yao, Q.; Wang, M.; Chen, Y.; Dai, W.; Li, Y.F.; Tu, W.W.; Yang, Q.; Yu, Y. Taking human out of learning applications: A survey on automated machine learning. *arXiv* **2018**, arXiv:1810.13306.
10. Waring, J.; Lindvall, C.; Umeton, R. Automated machine learning: Review of the state-of-the-art and opportunities for healthcare. *Artif. Intell. Med.* **2020**, *104*, 101822.
11. Ooms, R.; Spruit, M. Self-Service Data Science in Healthcare with Automated Machine Learning. *Appl. Sci.* **2020**, *10*, 2992.
12. Borkowski, A.A.; Wilson, C.P.; Borkowski, S.A.; Thomas, L.B.; Deland, L.A.; Grewe, S.J.; Mastorides, S.M. Google Auto ML versus Apple Create ML for Histopathologic Cancer Diagnosis; Which Algorithms Are Better? *arXiv* **2019**, arXiv:1903.08057.
13. Tsamardinos, I.; Charonyktakis, P.; Lakiotaki, K.; Borboudakis, G.; Zenklusen, J.C.; Juhl, H.; Chatzaki, E.; Lagani, V. Just Add Data: Automated Predictive Modeling and BioSignature Discovery. *bioRxiv* **2020**, doi:10.1101/2020.05.04.075747 .
14. Karaglani, M.; Gourlia, K.; Tsamardinos, I.; Chatzaki, E. Accurate Blood-Based Diagnostic Biosignatures for Alzheimer's Disease via Automated Machine Learning. *J. Clin. Med.* **2020**, *9*, 3016.
15. Gehrmann, S.; Dernoncourt, F.; Li, Y.; Carlson, E.T.; Wu, J.T.; Welt, J.; Foote, J., Jr.; Moseley, E.T.; Grant, D.W.; Tyler, P.D. Comparing rule-based and deep learning models for patient phenotyping. *arXiv* **2017**, arXiv:1703.08705.
16. Nigam, P. *Applying Deep Learning to ICD-9 Multi-Label Classification from Medical Records*; Technical report; Stanford University: Stanford, CA, USA, 2016.
17. Venkataraman, G.R.; Pineda, A.L.; Bear Don't Walk IV, O.J.; Zehnder, A.M.; Ayyar, S.; Page, R.L.; Bustamante, C.D.; Rivas, M.A. FasTag: Automatic text classification of unstructured medical narratives. *PLoS ONE* **2020**, *15*, e0234647.
18. Yogarajan, V.; Montiel, J.; Smith, T.; Pfahringer, B. Seeing The Whole Patient: Using Multi-Label Medical Text Classification Techniques to Enhance Predictions of Medical Codes. *arXiv* **2020**, arXiv:2004.00430.
19. Boytcheva, S. Automatic matching of ICD-10 codes to diagnoses in discharge letters. In Proceedings of the Second Workshop on Biomedical Natural Language Processing, Hissar, Bulgaria, 15 September 2011; pp. 11–18.
20. Huang, J.; Osorio, C.; Sy, L.W. An empirical evaluation of deep learning for ICD-9 code assignment using MIMIC-III clinical notes. *Comput. Methods Programs Biomed.* **2019**, *177*, 141–153.
21. Zheng, J.; Chapman, W.W.; Miller, T.A.; Lin, C.; Crowley, R.S.; Savova, G.K. A system for coreference resolution for the clinical narrative. *J. Am. Med. Inform. Assoc.* **2012**, *19*, 660–667.
22. Liu, H.; Wagholikar, K.B.; Jonnalagadda, S.; Sohn, S. Integrated cTAKES for Concept Mention Detection and Normalization. In Proceedings of the 2013 Cross Language Evaluation Forum Conference, Valencia, Spain, 23–26 September 2013.
23. Mullenbach, J.; Wiegreffe, S.; Duke, J.; Sun, J.; Eisenstein, J. Explainable prediction of medical codes from clinical text. *arXiv* **2018**, arXiv:1802.05695.
24. Bisong, E.; Google AutoML: Cloud Vision. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 581–598.
25. Alaa, A.M.; van der Schaar, M. Autoprognosis: Automated clinical prognostic modeling via bayesian optimization with structured kernel learning. *arXiv* **2018**, arXiv:1802.07207.
26. Baškarada, S.; Koronios, A. Unicorn data scientist: the rarest of breeds. *Program* **2017**, *51*, 65–74.
27. Zhang, S.; Zhang, C.; Yang, Q. Data preparation for data mining. *Appl. Artif. Intell.* **2003**, *17*, 375–381.
28. Khalid, S.; Khalil, T.; Nasreen, S. A survey of feature selection and feature extraction techniques in machine learning. In Proceedings of the 2014 Science and Information Conference, London, UK, 27–29 August 2014; pp. 372–378.
29. Yang, C.; Akimoto, Y.; Kim, D.W.; Udell, M. OBOE: Collaborative filtering for AutoML model selection. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 1173–1183.

30. Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R.P.; De Freitas, N. Taking the human out of the loop: A review of Bayesian optimization. *Proc. IEEE* **2015**, *104*, 148–175.
31. Thornton, C.; Hutter, F.; Hoos, H.H.; Leyton-Brown, K. 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, Chicago, IL, USA, 11–14 August 2013; pp. 847–855.
32. Kotthoff, L.; Thornton, C.; Hoos, H.H.; Hutter, F.; Leyton-Brown, K. Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA. *J. Mach. Learn. Res.* **2017**, *18*, 826–830.
33. Jungermann, F. Information extraction with rapidminer. Available online: https://duepublico2.uni-due.de/servlets/MCRFileNodeServlet/duepublico_derivate_00038023/Tagungsband_GSCLSYMP2009_final_6.pdf (accessed on 20 February 2021).
34. Gosiewska, A.; Bakala, M.; Woznica, K.; Zwolinski, M.; Biecek, P. EPP: interpretable score of model predictive power. *arXiv* **2019**, arXiv:1908.09213.
35. Perotte, A.; Pivovarov, R.; Natarajan, K.; Weiskopf, N.; Wood, F.; Elhadad, N. Diagnosis code assignment: models and evaluation metrics. *J. Am. Med. Inf. Assoc.* **2014**, *21*, 231–237.
36. King, J.; Magoulas, R. *2015 Data Science Salary Survey*; O'Reilly Media, Incorporated: Sebastopol, CA, USA, 2015.
37. Luo, G.; Stone, B.L.; Johnson, M.D.; Tarczy-Hornoch, P.; Wilcox, A.B.; Mooney, S.D.; Sheng, X.; Haug, P.J.; Nkoy, F.L. Automating construction of machine learning models with clinical big data: proposal rationale and methods. *JMIR Res. Protoc.* **2017**, *6*, e175.
38. Baars, H.; Kemper, H.G. Management support with structured and unstructured data—An integrated business intelligence framework. *Inf. Syst. Manag.* **2008**, *25*, 132–148.
39. Zhang, D.; Yin, C.; Zeng, J.; Yuan, X.; Zhang, P. Combining structured and unstructured data for predictive models: A deep learning approach. *BMC Med. Inform. Decis. Mak.* **2020**, *20*, 1–11.
40. Miiro, F.; Nääs, M. SQL and NoSQL Databases: A Case Study in the Azure Cloud. Bachelor' s Thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2015.
41. Barrenechea, M.J.; Jenkins, T. Enterprise Information Management: The Next Generation of Enterprise Software. *OpenText Waterloo (Can.)* **2013**.
42. Luo, G. A review of automatic selection methods for machine learning algorithms and hyper-parameter values. *Netw. Model. Anal. Health Inform. Bioinform.* **2016**, *5*, 18.
43. Zhang, Y.; Bahadori, M.T.; Su, H.; Sun, J. FLASH: fast Bayesian optimization for data analytic pipelines. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 13–16 August 2016; pp. 2065–2074.
44. Kim, S.; Kim, I.; Lim, S.; Baek, W.; Kim, C.; Cho, H.; Yoon, B.; Kim, T. Scalable neural architecture search for 3d medical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 220–228.
45. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
46. Weng, Y.; Zhou, T.; Li, Y.; Qiu, X. Nas-unet: Neural architecture search for medical image segmentation. *IEEE Access* **2019**, *7*, 44247–44257.
47. Olson, R.S.; Moore, J.H. TPOT: A tree-based pipeline optimization tool for automating machine learning. *Proc. Mach. Learn. Res.* **2016**, *64*, 66–74.
48. Jin, H.; Song, Q.; Hu, X. Auto-keras: An efficient neural architecture search system. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 1946–1956.
49. Drori, I.; Krishnamurthy, Y.; Rampin, R.; Lourenço, R.; One, J.; Cho, K.; Silva, C.; Freire, J. AlphaD3M: Machine learning pipeline synthesis. In Proceedings of the AutoML Workshop at ICML, Stockholm, Sweden, 14 July 2018.
50. Mendoza, H.; Klein, A.; Feurer, M.; Springenberg, J.T.; Hutter, F. Towards automatically-tuned neural networks. In Proceedings of the Workshop on Automatic Machine Learning, New York, NY, USA, 24 June 2016; pp. 58–65.
51. Swearingen, T.; Drevo, W.; Cyphers, B.; Cuesta-Infante, A.; Ross, A.; Veeramachaneni, K. ATM: A distributed, collaborative, scalable system for automated machine learning. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 151–162.
52. Komer, B.; Bergstra, J.; Eliasmith, C. Hyperopt-sklearn: Automatic hyperparameter configuration for scikit-learn. In Proceedings of the Scientific Computing with Python, Austin, TX, USA, 6–12 July 2014.
53. Mohr, F.; Wever, M.; Hüllermeier, E. ML-Plan: Automated machine learning via hierarchical planning. *Mach. Learn.* **2018**, *107*, 1495–1515.
54. Feurer, M.; Eggensperger, K.; Falkner, S.; Lindauer, M.; Hutter, F. Practical automated machine learning for the automl challenge 2018. In Proceedings of the International Workshop on Automatic Machine Learning at ICML, Stockholm, Sweden, 14 July 2018; pp. 1189–1232.
55. de Sá, A.G.; Pinto, W.J.G.; Oliveira, L.O.V.; Pappa, G.L. RECIPE: A grammar-based framework for automatically evolving classification pipelines. In *Proceedings of the European Conference on Genetic Programming*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 246–261.

56. Gijsbers, P.; Vanschoren, J.; Olson, R.S. Layered TPOT: Speeding up tree-based pipeline optimization. *arXiv* **2018**, arXiv:1801.06007.
57. Chen, B.; Wu, H.; Mo, W.; Chattopadhyay, I.; Lipson, H. Autostacker: A compositional evolutionary learning system. In Proceedings of the Genetic and Evolutionary Computation Conference, Kyoto, Japan, 15–19 July 2018; pp. 402–409.
58. Dafflon, J.; Pinaya, W.H.; Turkheimer, F.; Cole, J.H.; Leech, R.; Harris, M.A.; Cox, S.R.; Whalley, H.C.; McIntosh, A.M.; Hellyer, P.J. An automated machine learning approach to predict brain age from cortical anatomical measures. *Hum. Brain Mapp.* **2020**, doi:10.1002/hbm.25028.
59. Su, X.; Chen, N.; Sun, H.; Liu, Y.; Yang, X.; Wang, W.; Zhang, S.; Tan, Q.; Su, J.; Gong, Q. Automated machine learning based on radiomics features predicts H3 K27M mutation in midline gliomas of the brain. *Neuro-oncology* **2020**, *22*, 393–401.
60. Orlenko, A.; Moore, J.H.; Orzechowski, P.; Olson, R.S.; Cairns, J.; Caraballo, P.J.; Weinshilboum, R.M.; Wang, L.; Breitenstein, M.K. Considerations for Automated Machine Learning in Clinical Metabolic Profiling: Altered Homocysteine Plasma Concentration Associated with Metformin Exposure. *Biocomputing* **2018**, *23*, 460–471.
61. Zeng, Y.; Zhang, J. A machine learning model for detecting invasive ductal carcinoma with Google Cloud AutoML Vision. *Comput. Biol. Med.* **2020**, *122*, 103861.
62. Mantas, J. Setting up an Easy-to-Use Machine Learning Pipeline for Medical Decision Support: A Case Study for COVID-19 Diagnosis Based on Deep Learning with CT Scans. *Importance Health Inform. Public Health Pandemic* **2020**, *272*, 13.
63. Faes, L.; Wagner, S.K.; Fu, D.J.; Liu, X.; Korot, E.; Ledsam, J.R.; Back, T.; Chopra, R.; Pontikos, N.; Kern, C. Automated deep learning design for medical image classification by health-care professionals with no coding experience: a feasibility study. *Lancet Digit. Health* **2019**, *1*, e232–e242.
64. Puri, M. Automated machine learning diagnostic support system as a computational biomarker for detecting drug-induced liver injury patterns in whole slide liver pathology images. *Assay Drug Dev. Technol.* **2020**, *18*, 1–10.
65. Kim, I.K.; Lee, K.; Park, J.H.; Baek, J.; Lee, W.K. Classification of pachychoroid disease on ultrawide-field indocyanine green angiography using auto-machine learning platform. *Br. J. Ophthalmol.* **2020**, doi:10.1136/bjophthalmol-2020-316108.
66. Kocbek, S.; Kocbek, P.; Zupanic, T.; Stiglic, G.; Gabrys, B. Using (Automated) Machine Learning and Drug Prescription Records to Predict Mortality and Polypharmacy in Older Type 2 Diabetes Mellitus Patients. In *Proceedings of the International Conference on Neural Information Processing*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 624–632.
67. Bhat, G.S.; Shankar, N.; Panahi, I.M. Automated machine learning based speech classification for hearing aid applications and its real-time implementation on smartphone. *Annu Int Conf IEEE Eng Med Biol Soc.* **2020**, 956–959, doi:10.1109/EMBC44109.2020.9175693.
68. Truong, A.; Walters, A.; Goodsitt, J.; Hines, K.; Bruss, C.B.; Farivar, R. Towards automated machine learning: Evaluation and comparison of automl approaches and tools. In Proceedings of the 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), Portland, OR, USA, 4–6 November 2019; pp. 1471–1479.
69. Tsanas, A.; Little, M.; McSharry, P.; Ramig, L. Accurate telemonitoring of Parkinson's disease progression by non-invasive speech tests. *Nat. Preced.* **2009**, doi:10.1038/npre.2009.3920.1.
70. Khan, W.; Daud, A.; Nasir, J.A.; Amjad, T. A survey on the state-of-the-art machine learning models in the context of NLP. *Kuwait J. Sci.* **2016**, *43*, 95–113.
71. Weng, W.H.; Wagholikar, K.B.; McCray, A.T.; Szolovits, P.; Chueh, H.C. Medical subdomain classification of clinical notes using a machine learning-based natural language processing approach. *BMC Med. Inform. Decis. Mak.* **2017**, *17*, 1–13.
72. Gupta, S.; MacLean, D.L.; Heer, J.; Manning, C.D. Induced lexico-syntactic patterns improve information extraction from online medical forums. *J. Am. Med. Inform. Assoc.* **2014**, *21*, 902–909.
73. Li, Y.; Krishnamurthy, R.; Raghavan, S.; Vaithyanathan, S.; Jagadish, H. Regular expression learning for information extraction. In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, HI, USA, 25–27 October 2008; pp. 21–30.
74. Wei, Q.; Ji, Z.; Li, Z.; Du, J.; Wang, J.; Xu, J.; Xiang, Y.; Tiryaki, F.; Wu, S.; Zhang, Y. A study of deep learning approaches for medication and adverse drug event extraction from clinical text. *J. Am. Med. Inform. Assoc.* **2020**, *27*, 13–21.
75. Kaur, R. A comparative analysis of selected set of natural language processing (NLP) and machine learning (ML) algorithms for clinical coding using clinical classification standards. *Stud. Health Technol. Inform.* **2018**, *252*, 73–79.
76. Cai, T.; Giannopoulos, A.A.; Yu, S.; Kelil, T.; Ripley, B.; Kumamaru, K.K.; Rybicki, F.J.; Mitsouras, D. Natural language processing technologies in radiology research and clinical applications. *Radiographics* **2016**, *36*, 176–191.
77. Liu, K.; Hogan, W.R.; Crowley, R.S. Natural language processing methods and systems for biomedical ontology learning. *J. Biomed. Inform.* **2011**, *44*, 163–179.
78. Medori, J.; Fairon, C. Machine learning and features selection for semi-automatic ICD-9-CM encoding. In Proceedings of the NAACL HLT 2010 Second Louhi Workshop on Text and Data Mining of Health Documents, Los Angeles, CA, USA, 5 June 2010; pp. 84–89.
79. Pakhomov, S.; Chute, C.G. A Hybrid Approach to Determining Modification of Clinical Diagnoses. In Proceedings of the AMIA Annual Symposium Proceedings, Washington, DC, USA, 11–15 November 2006; American Medical Informatics Association: Bethesda, MD, USA, 2006; p. 609.
80. Estevez-Velarde, S.; Gutiérrez, Y.; Montoyo, A.; Almeida-Cruz, Y. AutoML strategy based on grammatical evolution: A case study about knowledge discovery from text. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 4356–4365.

81. Feurer, M.; Eggensperger, K.; Falkner, S.; Lindauer, M.; Hutter, F. Auto-sklearn 2.0: The next generation. *arXiv* **2020**, arXiv:2007.04074.
82. Wang, Y.; Sohn, S.; Liu, S.; Shen, F.; Wang, L.; Atkinson, E.J.; Amin, S.; Liu, H. A clinical text classification paradigm using weak supervision and deep representation. *BMC Med. Inform. Decis. Mak.* **2019**, *19*, 1.
83. Trivedi, H.M.; Panahiazar, M.; Liang, A.; Lituiev, D.; Chang, P.; Sohn, J.H.; Chen, Y.Y.; Franc, B.L.; Joe, B.; Hadley, D. Large scale semi-automated labeling of routine free-text clinical records for deep learning. *J. Digit. Imaging* **2019**, *32*, 30–37.
84. Alzoubi, H.; Ramzan, N.; Alzubi, R.; Mesbahi, E. An Automated System for Identifying Alcohol Use Status from Clinical Text. In Proceedings of the 2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE), Southend, UK, 16–17 August 2018; pp. 41–46.
85. Xu, K.; Lam, M.; Pang, J.; Gao, X.; Band, C.; Mathur, P.; Papay, F.; Khanna, A.K.; Cywinski, J.B.; Maheshwari, K. Multimodal machine learning for automated ICD coding. In Proceedings of the Machine Learning for Healthcare Conference, PMLR, Ann Arbor, MI, USA, 8–10 August 2019; pp. 197–215.
86. Aronson, A.R.; Bodenreider, O.; Demner-Fushman, D.; Fung, K.W.; Lee, V.K.; Mork, J.G.; Névéol, A.; Peters, L.; Rogers, W.J. From indexing the biomedical literature to coding clinical text: experience with MTI and machine learning approaches. In Proceedings of the Biological, Translational, and Clinical Language Processing, Prague, Czech Republic, 29 June 2007; pp. 105–112.
87. Obeid, J.S.; Weeda, E.R.; Matuskowitz, A.J.; Gagnon, K.; Crawford, T.; Carr, C.M.; Frey, L.J. Automated detection of altered mental status in emergency department clinical notes: A deep learning approach. *BMC Med. Inform. Decis. Mak.* **2019**, *19*, 164.
88. Soguero-Ruiz, C.; Hindberg, K.; Rojo-Álvarez, J.L.; Skrøvseth, S.O.; Godtliebsen, F.; Mortensen, K.; Revhaug, A.; Lindsetmo, R.O.; Augestad, K.M.; Jenssen, R. Support vector feature selection for early detection of anastomosis leakage from bag-of-words in electronic health records. *IEEE J. Biomed. Health Inform.* **2014**, *20*, 1404–1415.
89. Atutxa, A.; Pérez, A.; Casillas, A. Machine learning approaches on diagnostic term encoding with the ICD for clinical documentation. *IEEE J. Biomed. Health Inform.* **2017**, *22*, 1323–1329.
90. Kalyan, K.S.; Sangeetha, S. Secnlp: A survey of embeddings in clinical natural language processing. *J. Biomed. Inform.* **2020**, *101*, 103323.
91. Shi, H.; Xie, P.; Hu, Z.; Zhang, M.; Xing, E.P. Towards automated ICD coding using deep learning. *arXiv* **2017**, arXiv:1711.04075.
92. Polignano, M.; Suriano, V.; Lops, P.; de Gemmis, M.; Semeraro, G. A study of Machine Learning models for Clinical Coding of Medical Reports at CodiEsp 2020. In Proceedings of the Working Notes of Conference and Labs of the Evaluation (CLEF) Forum, CEUR Workshop Proceedings, Thessaloniki, Greece, 2–25 September 2020.
93. Karmakar, A. Classifying medical notes into standard disease codes using Machine Learning. *arXiv* **2018**, arXiv:1802.00382.
94. Dubois, S.; Romano, N. Learning effective embeddings from medical notes. *arXiv* **2017**, arXiv:1705.07025.
95. Lin, C.; Hsu, C.J.; Lou, Y.S.; Yeh, S.J.; Lee, C.C.; Su, S.L.; Chen, H.C. Artificial intelligence learning semantics via external resources for classifying diagnosis codes in discharge notes. *J. Med. Internet Res.* **2017**, *19*, e380.
96. Savova, G.K.; Masanz, J.J.; Ogren, P.V.; Zheng, J.; Sohn, S.; Kipper-Schuler, K.C.; Chute, C.G. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *J. Am. Med. Inform. Assoc.* **2010**, *17*, 507–513.
97. Torii, M.; Wagholikar, K.; Liu, H. Using machine learning for concept extraction on clinical documents from multiple data sources. *J. Am. Med. Inform. Assoc.* **2011**, *18*, 580–587.
98. Cobb, R.; Puri, S.; Wang, D.Z.; Baslanti, T.; Bihorac, A. Knowledge extraction and outcome prediction using medical notes. In Proceedings of the ICML Workshop on Role of Machine Learning in Transforming Healthcare, Atlanta, GA, USA, 20–21 June 2013.
99. Ni, Y.; Wright, J.; Perentesis, J.; Lingren, T.; Deleger, L.; Kaiser, M.; Kohane, I.; Solti, I. Increasing the efficiency of trial-patient matching: automated clinical trial eligibility pre-screening for pediatric oncology patients. *BMC Med. Inform. Decis. Mak.* **2015**, *15*, 28.
100. Garla, V.N.; Brandt, C. Ontology-guided feature engineering for clinical text classification. *J. Biomed. Inform.* **2012**, *45*, 992–998.
101. Spasić, I.; Livsey, J.; Keane, J.A.; Nenadić, G. Text mining of cancer-related information: review of current status and future directions. *Int. J. Med. Inform.* **2014**, *83*, 605–623.
102. Gonzalez-Hernandez, G.; Sarker, A.; O'Connor, K.; Savova, G. Capturing the patient's perspective: A review of advances in natural language processing of health-related text. *Yearb. Med. Inform.* **2017**, *26*, 214.
103. Khare, R.; Wei, C.H.; Lu, Z. Automatic extraction of drug indications from FDA drug labels. In Proceedings of the AMIA Annual Symposium Proceedings, Washington, DC, USA, 15–19 November 2014; Volume 2014, p. 787.
104. Reátegui, R.; Ratté, S. Comparison of MetaMap and cTAKES for entity extraction in clinical notes. *BMC Med. Inform. Decis. Mak.* **2018**, *18*, 74.
105. Uzuner, Ö. Recognizing obesity and comorbidities in sparse data. *J. Am. Med. Inform. Assoc.* **2009**, *16*, 561–570.
106. Suominen, H.; Ginter, F.; Pyysalo, S.; Airola, A.; Pahikkala, T.; Salanter, S.; Salakoski, T. Machine learning to automate the assignment of diagnosis codes to free-text radiology reports: A method description. In Proceedings of the ICML/UAI/COLT Workshop on Machine Learning for Health-Care Applications, Helsinki, Filand, 9 July 2008.
107. Mwangi, B.; Tian, T.S.; Soares, J.C. A review of feature reduction techniques in neuroimaging. *Neuroinformatics* **2014**, *12*, 229–244.
108. Ngwenya, M. Health Systems Data Interoperability and Implementation. Master's Thesis, University of South Africa, Pretoria, South Africa, 2018.

109. Mujtaba, G.; Shuib, L.; Idris, N.; Hoo, W.L.; Raj, R.G.; Khowaja, K.; Shaikh, K.; Nweke, H.F. Clinical text classification research trends: Systematic literature review and open issues. *Expert Syst. Appl.* **2019**, *116*, 494–520.
110. Sehjal, R.; Harries, V. Awareness of clinical coding: A survey of junior hospital doctors. *Br. J. Healthc. Manag.* **2016**, *22*, 310–314.
111. Mujtaba, G.; Shuib, L.; Raj, R.G.; Rajandram, R.; Shaikh, K.; Al-Garadi, M.A. Automatic ICD-10 multi-class classification of cause of death from plaintext autopsy reports through expert-driven feature selection. *PLoS ONE* **2017**, *12*, e0170242.
112. Scheurwegs, E.; Cule, B.; Luyckx, K.; Luyten, L.; Daelemans, W. Selecting relevant features from the electronic health record for clinical code prediction. *J. Biomed. Inform.* **2017**, *74*, 92–103.
113. Scheurwegs, E.; Luyckx, K.; Luyten, L.; Goethals, B.; Daelemans, W. Assigning clinical codes with data-driven concept representation on Dutch clinical free text. *J. Biomed. Inform.* **2017**, *69*, 118–127.
114. Ferrão, J.C.; Oliveira, M.D.; Janela, F.; Martins, H.M.; Gartner, D. Can structured EHR data support clinical coding? A data mining approach. *Health Syst.* **2020**, 1–24, doi:10.1080/20476965.2020.1729666.
115. Balakrishnan, S.; Narayanaswamy, R. Feature selection using fcbf in type ii diabetes databases. *Int. J. Comput. Internet Manag.* **2009**, *17*, 50–8.
116. Zhang, W.; Tang, J.; Wang, N. Using the machine learning approach to predict patient survival from high-dimensional survival data. In Proceedings of the 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Sehnzhen, China, 15–18 December 2016; pp. 1234–1238.
117. Buettner, R.; Klenk, F.; Ebert, M. A systematic literature review of machine learning-based disease profiling and personalized treatment. In Proceedings of the 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, 13–17 July 2020.
118. Yu, L.; Liu, H. Feature selection for high-dimensional data: A fast correlation-based filter solution. In Proceedings of the 20th International Conference on Machine Learning (ICML-03), Washington, DC, USA, 21–24 August, 2003; pp. 856–863.
119. Raza, M.S.; Qamar, U. *Understanding and Using Rough Set Based Feature Selection: Concepts, Techniques and Applications*; Springer: Berlin/Heidelberg, Germany, 2017.
120. Goldberg, D.E. *Genetic Algorithms*; Pearson Education India: Chennai, India, 2006.
121. Chandrashekar, G.; Sahin, F. A survey on feature selection methods. *Comput. Electr. Eng.* **2014**, *40*, 16–28.
122. Vehtari, A.; Gelman, A.; Gabry, J. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Stat. Comput.* **2017**, *27*, 1413–1432.
123. Schumacher, M.; Holländer, N.; Sauerbrei, W. Resampling and cross-validation techniques: a tool to reduce bias caused by model building? *Stat. Med.* **1997**, *16*, 2813–2827.
124. Saito, T.; Rehmsmeier, M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE* **2015**, *10*, e0118432.
125. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82.
126. Escalante, H.J.; Montes, M.; Sucar, L.E. Particle swarm model selection. *J. Mach. Learn. Res.* **2009**, *10*, 405–440.
127. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of ICNN'95-International Conference on Neural Networks, Perth, Austrilia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
128. Pineda, A.L.; Ye, Y.; Visweswaran, S.; Cooper, G.F.; Wagner, M.M.; Tsui, F.R. Comparison of machine learning classifiers for influenza detection from emergency department free-text reports. *J. Biomed. Inform.* **2015**, *58*, 60–69.
129. Chen, Y. Predicting ICD-9 Codes from Medical Notes–Does the Magic of BERT Applies Here? Available online:https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1204/reports/custom/report25.pdf (accessed on 20 Febraury 2021)
130. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
131. Zeng, M.; Li, M.; Fei, Z.; Yu, Y.; Pan, Y.; Wang, J. Automatic ICD-9 coding via deep transfer learning. *Neurocomputing* **2019**, *324*, 43–50.
132. Li, M.; Fei, Z.; Zeng, M.; Wu, F.X.; Li, Y.; Pan, Y.; Wang, J. Automated ICD-9 coding via a deep learning approach. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2018**, *16*, 1193–1202.
133. Malik, S.; Kanwal, N.; Asghar, M.N.; Sadiq, M.A.A.; Karamat, I.; Fleury, M. Data Driven Approach for Eye Disease Classification with Machine Learning. *Appl. Sci.* **2019**, *9*, 2789.
134. Ananthakrishnan, A.N.; Cai, T.; Savova, G.; Cheng, S.C.; Chen, P.; Perez, R.G.; Gainer, V.S.; Murphy, S.N.; Szolovits, P.; Xia, Z. Improving case definition of Crohn's disease and ulcerative colitis in electronic medical records using natural language processing: a novel informatics approach. *Inflamm. Bowel Dis.* **2013**, *19*, 1411–1420.
135. Vukicevic, M.; Radovanovic, S.; Stiglic, G.; Delibasic, B.; Van Poucke, S.; Obradovic, Z. A data and knowledge driven randomization technique for privacy-preserving data enrichment in hospital readmission prediction. In Proceedings of the 5th Workshop on Data Mining for Medicine and Healthcare, Miami, FL, USA, 7 May 2016; Volume 10.
136. Farkas, R.; Szarvas, G. Automatic construction of rule-based ICD-9-CM coding systems. *BMC Bioinform.* **2008**, *9*, S10.
137. Li, L.; Jamieson, K.; DeSalvo, G.; Rostamizadeh, A.; Talwalkar, A. Hyperband: A novel bandit-based approach to hyperparameter optimization. *J. Mach. Learn. Res.* **2017**, *18*, 6765–6816.
138. Tsoumakas, G.; Katakis, I. Multi-label classification: An overview. *Int. J. Data Warehous. Min. (IJDWM)* **2007**, *3*, 1–13.
139. Meystre, S.; Haug, P.J. Natural language processing to extract medical problems from electronic clinical documents: Performance evaluation. *J. Biomed. Inform.* **2006**, *39*, 589–599.

140. Miotto, R.; Li, L.; Kidd, B.A.; Dudley, J.T. Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. *Sci. Rep.* **2016**, *6*, 1–10.
141. Sohn, S.; Savova, G.K. Mayo clinic smoking status classification system: extensions and improvements. In Proceedings of the AMIA Annual Symposium Proceedings, San Francisco, CA, USA, 14–18 November 2009; American Medical Informatics Association: Bethesda, MD, USA 2009; Volume 2009, p. 619.
142. Read, J.; Reutemann, P.; Pfahringer, B.; Holmes, G. Meka: A multi-label/multi-target extension to weka. *J. Mach. Learn. Res.* **2016**, *17*, 667–671.
143. Pfaff, E.R.; Crosskey, M.; Morton, K.; Krishnamurthy, A. Clinical Annotation Research Kit (CLARK): Computable Phenotyping Using Machine Learning. *JMIR Med. Inform.* **2020**, *8*, e16042.
144. Mani, S.; Chen, Y.; Elasy, T.; Clayton, W.; Denny, J. Type 2 diabetes risk forecasting from EMR data using machine learning. In Proceedings of the AMIA Annual Symposium Proceedings, Chicago, IL, USA, 3–7 November 2012; American Medical Informatics Association: Bethesda, MD, USA 2012; Volume 2012, p. 606.
145. Skeppstedt, M.; Kvist, M.; Nilsson, G.H.; Dalianis, H. Automatic recognition of disorders, findings, pharmaceuticals and body structures from clinical text: An annotation and machine learning study. *J. Biomed. Inform.* **2014**, *49*, 148–158.
146. Sohn, S.; Kocher, J.P.A.; Chute, C.G.; Savova, G.K. Drug side effect extraction from clinical narratives of psychiatry and psychology patients. *J. Am. Med. Inform. Assoc.* **2011**, *18*, i144–i149.
147. Kullo, I.J.; Fan, J.; Pathak, J.; Savova, G.K.; Ali, Z.; Chute, C.G. Leveraging informatics for genetic studies: use of the electronic medical record to enable a genome-wide association study of peripheral arterial disease. *J. Am. Med. Inform. Assoc.* **2010**, *17*, 568–574.
148. Walsh, C.G.; Ribeiro, J.D.; Franklin, J.C. Predicting risk of suicide attempts over time through machine learning. *Clin. Psychol. Sci.* **2017**, *5*, 457–469.
149. Divita, G.; Luo, G.; Tran, L.; Workman, T.E.; Gundlapalli, A.V.; Samore, M.H. General Symptom Extraction from VA Electronic Medical Notes. *Stud. Health Technol. Inform.* **2017**, *245*, 356.
150. Ghiasvand, O. Disease Name Extraction from Clinical Text Using Conditional Random Fields. Master's Thesis, University of Wisconsin-Milwaukee, Milwaukee, WI, USA, May 2014.
151. Guyon, I.; Bennett, K.; Cawley, G.; Escalante, H.J.; Escalera, S.; Ho, T.K.; Macia, N.; Ray, B.; Saeed, M.; Statnikov, A. Design of the 2015 chalearn automl challenge. In Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 12–16 July 2015; pp. 1–8.