IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 23, 2020, accepted November 22, 2020, date of publication December 8, 2020, date of current version January 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3043221

Ensembling Classical Machine Learning and Deep Learning Approaches for Morbidity Identification From Clinical Notes

VIVEK KUMAR[1], **DIEGO REFORGIATO RECUPERO**[1], **DANIELE RIBONI**[1], **(Member, IEEE), AND RIM HELAOUI**[2]

[1]Department of Mathematics and Computer Science, University of Cagliari, 09124 Cagliari, Italy
[2]Philips Research, 5656 Eindhoven, The Netherlands

Corresponding author: Vivek Kumar (vivek.kumar@unica.it)

This work was supported by the EU's Marie Curie training network PhilHumans—Personal Health Interfaces Leveraging Human–Machine Natural Interactions under Agreement 812882.

ABSTRACT The past decade has seen an explosion of the amount of digital information generated within the healthcare domain. Digital data exist in the form of images, video, speech, transcripts, electronic health records, clinical records, and free-text. Analysis and interpretation of healthcare data is a daunting task, and it demands a great deal of time, resources, and human effort. In this paper, we focus on the problem of co-morbidity recognition from patient's clinical records. To this aim, we employ both classical machine learning and deep learning approaches. We use word embeddings and bag-of-words representations, coupled with feature selection techniques. The goal of our work is to develop a classification system to identify whether a certain health condition occurs for a patient by studying his/her past clinical records. In more detail, we have used pre-trained word2vec, domain-trained, GloVe, fastText, and universal sentence encoder embeddings to tackle the classification of sixteen morbidity conditions within clinical records. We have compared the outcomes of classical machine learning and deep learning approaches with the employed feature representation methods and feature selection methods. We present a comprehensive discussion of the performances and behaviour of the employed classical machine learning and deep learning approaches. Finally, we have also used ensemble learning techniques over a large number of combinations of classifiers to improve the single model performance. For our experiments, we used the n2c2 natural language processing research dataset, released by Harvard Medical School. The dataset is in the form of clinical notes that contain patient discharge summaries. Given the unbalancedness of the data and their small size, the experimental results indicate the advantage of the ensemble learning technique with respect to single classifier models. In particular, the ensemble learning technique has slightly improved the performances of single classification models but has greatly reduced the variance of predictions stabilizing the accuracies (i.e., the lower standard deviation in comparison with single classifiers). In real-life scenarios, our work can be employed to identify with high accuracy morbidity conditions of patients by feeding our tool with their current clinical notes. Moreover, other domains where classification is a common problem might benefit from our approach as well.

INDEX TERMS Deep learning, machine learning, multimorbidity, natural language processing, classifiers, word embeddings, healthcare.

I. INTRODUCTION

In the last years, we have observed a rise in life expectancy, which has also increased the risk of long-term diseases such as diabetes, cognitive impairment, and many other severe health issues [1]–[4]. A further downside of a longer lifespan is that people can be affected by more than one disease at a time, leading to the likelihood of under-standard quality of life. An individual with long-term diabetes, for example, has a higher risk of hypertension, high cholesterol levels, blockage of the arteries or veins. According to the World Health Organization report [5], 40% of the population is

The associate editor coordinating the review of this manuscript and approving it for publication was Sabah Mohammed.

exposed to at least one long-term health condition, and 25% of the population suffers from multimorbidity in a developed country. In addition, the report also emphasizes the directly proportional relationship between the high incidence of multimorbidity and middle and low-income countries because they do not have funds that should be invested to enhance primary care of the population [6]. There is, therefore, the need to continuously track medical information.

With the introduction of information technology systems, more and more clinical records are constantly being produced, processed, and analyzed. The information encoded by clinical reports could be used to provide new healthcare services globally, addressing the problems related to people's social or economic status. As an example, clinical reports contain a variety of information in the form of numbers (e.g., laboratory results), images (e.g., x-ray), medical descriptions (e.g., treatment history), or transcripts (e.g., motivational interviewing therapy sessions), which can be used to create content-based services to assist patients and medical practitioners.

The analysis and human interpretation of healthcare data are challenging because of their dimension and unstructured and heterogeneous formats. Hence, Artificial Intelligence technologies are more massively applied to analyze healthcare big data [7]. For instance, they have been applied to textual clinical reports to perform tasks such as classification [8], clustering [9], and recommendation [10]. The state-of-the-art research in this direction has already yielded significant results, and many challenges [11] are further explored with the goal of assisting the healthcare personnel. They include dynamic forecasting [12], personalized monitoring [13], and individualized treatment recommendations [14] of patients, especially those presenting multimorbidities as considered more vulnerable. According to [5], given that 25% of the world population is already suffering from multimorbidity, its early identification is paramount for preventing the severe health issues which can happen in the future to the patients. Therefore, in our work, we aim at automatically identifying the multimorbidity factors indicated in the patient's clinical records. The morbidity identification is of great significance in assisting the healthcare personnel with several downstream tasks involving the handling of large volumes of electronic health records. For our experiments, we have used a dataset that contains the clinical records of patients, indicating the presence of one or more morbidity factors. In addition, deep learning (DL) models and advanced word embeddings representations have recently proven to be state-of-the-art for many natural language processing (NLP) tasks and are popularly used within many healthcare problems. Hence, in order to exploit their advantages, we have focused upon the representation of clinical records by methods such as word embeddings and bag-of-words in combination with feature selection techniques using classical machine learning (CML) and DL approaches. The work focuses on discovering whether the patients are suffering from single or multiple morbidity conditions by studying their past clinical records. In the following, we will list more in detail the contributions of our paper:

- We use CML and DL approaches for performing morbidity detection within clinical notes.
- We experimentally compare five pre-trained word embeddings and four bag-of-words representations coupled with different feature selection algorithms.
- We compare the proposed DL approaches against CML approaches with different bag-of-words feature representations.
- We compare the proposed DL approaches against CML approaches with word embeddings feature representations.
- Out of several CML and DL models we tested, we analysed their inclusion in an ensemble strategy to improve single models' performances.
- We prove that in the presence of small datasets, single classifiers obtain unstable performances, whereas ensemble approaches mitigate this instability and, at the same time, increases the accuracy of the overall classification. Note that our ensemble approach's computational cost affects only the training step, but not the prediction phase.
- We provide a comprehensive discussion over the performances of CML and DL approaches with each kind of feature representation and the advantages of using the ensemble strategy and under which constraints.

The remainder of the manuscript is organized as follows. Section II presents the literature survey and related work. Section III describes the motivations behind this work and defines the problem statement we are tackling. It also includes the details about the dataset description and the preprocessing we have performed. Section IV discusses the different types of feature representation methods we employed. Section V details the classification models used for this work. Section VI presents the experimental evaluation and the obtained results. Section VII includes the observations and trends of the classifiers' behaviour and the ensemble strategy we have come up with. Finally, Section VIII draws the conclusion for the conducted experiments and obtained results and shows the directions where we are headed.

II. RELATED WORK

This section briefly reviews the existing NLP and Artificial Intelligence methods within the healthcare domain and how the feature selection techniques and the word embedding models have been employed.

A. ARTIFICIAL INTELLIGENCE IN HEALTHCARE

Today Artificial Intelligence and its sub-fields such as DL, Text Mining, and in general, CML play an important role in clinical decision-making, comprehension, predictive disease detection, and therapy assistance [15]. DL healthcare applications made significant improvements in many areas, such as the analysis of blood samples, the identification of heart attacks, tumors, and so on [16]. DL models' high-quality performances for healthcare problems have brought to encouraging discussions and interest within the Artificial Intelligence community.

The use of DL techniques to identify multimorbidity in clinical reports have been extensively studied in recent years. For instance, DL models in [17] were fed by word and entity embeddings to the following two layers, Convolutional Neural Network (CNN) and second Max Pooling. The model improved the results that were obtained during the *i2b2*[1] obesity challenge in 2008. Another work [18] proposed DL based approaches for morbidity status identification. It was focused on automatic learning from the clinical records and feature discovery to disengage hand-crafted feature selection using single and multi-channel CNN models. The single-channel CNN model used an embedding layer to train the model, whereas the multi-channel model employed multiple CNN models in parallel, as an ensemble of CNN models, where each used different hyper-parameters. One more work [19] investigated the performances of long-short term memory (LSTM) networks for entity recognition based on character and word-level representations. The proposed LSTM model outperformed traditional state-of-the-art methods, such as the conditional random field for entity recognition. Authors in [20] uncovered the implementation of sentiment analysis techniques for patient discharge summaries classification. The proposed hybrid model used a semi-supervised technique based on the vector space model and statistical methods in conjunction with extreme learning machine auto-encoder. The goal was to examine and evaluate the treatment quality based on the discharge summaries. In [21], the authors tackled a multi-label binary text classification problem using the rule-based classifier and orthogonal machine learning strategies. The work evaluated the performances of long short-term memory against logistic regression employing pre-trained BioWordVec and domain-trained word embeddings representations. The work presented in [22] investigated the DL approaches, which used pre-trained language models on relation extraction from clinical records. Authors applied pre-trained and fine-tuned Bidirectional Encoder Representations from Transformers (BERT), showing that the fine-tune method (FT-BERT) performed better than the feature-based method (FC-BERT).

All the works mentioned above were focused on just DL or CML techniques. In fact, to the best of our knowledge, there are not many existing papers available in the literature within the healthcare domain where CML and DL techniques have been extensively compared. We address this by presenting a paper where we carried out an extensive set of experiments using DL and CML techniques with different combinations of feature representation models and word embeddings. Moreover, we employed ensemble strategies to further increase single models' accuracy and tested several combinations of CML and DL approaches with different feature representation techniques. The best heterogeneous ensembles we obtained at the end of the process exploited the pros of each constituent. Our target was a multi-classification task (i.e., identifying several morbidity factors) within the healthcare domain. We wanted to conduct one more analysis: how each CML and DL method behaved within the underlying domain using a small set of clinical notes.

B. WORD EMBEDDINGS MODELS

Clinical records are mostly in the form of free-text, which are unstructured, contain typographical errors, and are comprised of healthcare domain-specific terminologies [23]. The representation of these clinical records in a way that they can be used effectively by CML and DL approaches remains one of the top challenges within the healthcare domain. To exploit the hidden semantics within the clinical notes, using word embeddings is a must. The work in [24] provides a guide for training word embeddings on clinical text data. It discusses the different types of word representations, clinical text corpora, available pre-trained clinical word vector embeddings, intrinsic and extrinsic evaluation, applications, and limitations of these approaches. Authors in [25] leveraged the infused elementary distance matrix to update the topic distributions for calculating the corresponding optimal transports. This strategy provides the update of word embeddings with robust guidance, improving the algorithmic convergence. As an initial study, the paper [26] presented a comparative analysis of CML and DL approaches with different types of feature representations such as Term Frequency-Inverse Document Frequency (TF-IDF) and word embeddings.

Concerning previous works, in our approach, we have used five-word embeddings, namely pre-trained word2vec, domain-trained, GloVe, fastText, and USE, to model the input datasets with and without the stop words. One of the purposes was also to observe the impact of stopwords within the considered domain. The removal of stopwords can often lead to a different outcome, as it changes the context and the meaning of a sentence. For instance, using stopwords removal, the sentence *The patient is not stable* might turn into *The patient is stable*, thus changing the meaning of the initial sentence.

The literature suggests several interesting works that utilize state-of-the-art word embeddings and bag-of-words representations. However, not many have discussed the impact of different feature representation approaches for imbalanced datasets. To bridge this knowledge gap, we have used word embeddings and sentence embeddings generated by USE, along with different kinds of bag-of-words representations. We have also provided a detailed discussion about the impact of stopwords in word embeddings, observed from the performed experiments.

C. FEATURE SELECTION

Feature engineering in NLP involved creating specific numerical functions to represent salient aspects of the text, such as the nouns and pronouns ratio. This approach often required significant domain knowledge and effort to identify

[1] https://www.i2b2.org/NLP/Obesity/

meaningful features. Feature selection is extensively used to reduce data by eliminating irrelevant and superfluous attributes from the dataset [27], [28]. This technique enhances the data interpretation, improves data visualization, reduces training time of learning algorithms, and improves prediction performances [29]. The work in [30] mentions the effectiveness of feature selection algorithms in several applications and highlights the challenges faced due to the unique characteristics of data. In work performed in [31], the authors aimed to achieve an affordable, fast, and objective diagnosis of the genetic variant of oligodendroglioma by combining the feature selection with ensemble-based classification. In addition, the work in [32] presented a method called *FREGEX*, which is based on regular expressions to extract features from biomedical, clinical notes. It was used as a substitute for the *n*-grams based feature selection method and employed the algorithms Smith-Waterman and Needleman-Wunsch for sequence alignment. The three datasets used to evaluate the proposed method's performances were manually annotated and contained information on smoking habits, obesity, and obesity types. The features extracted by *FREGEX* based on regular expressions improved the performance of SVM and Naive Bayes based classifiers. The work in [33] used a modified differential evolution algorithm to perform feature selection for cardiovascular disease and optimization of selected features. It also evaluated several performance measures for the prediction of heart disease to combine the modified differential evolution algorithm with a feed-forward neural network and fuzzy analytical hierarchy process.

In our work, we have used three feature selection algorithms with both CML and DL approaches to exploit the advantages in identifying the features necessary for distinguishing the morbidity classes, as well as in substantially reducing the computation time for training the models. The majority of existing works with healthcare data are confined to using a few feature selection techniques, either with CML or DL algorithms. To contribute to the body of knowledge, we have applied multiple combinations of feature representation and selection. Moreover, we came up with a set of ensembles made of heterogeneous constituents that outperform the single classifiers. We provided a detailed discussion about the impact of the feature engineering process on the used dataset for the multi-classification task.

III. PROBLEM FORMULATION, DATASET, AND PREPROCESSING

This section provides the formulation of the problem we addressed, the used dataset, and the related preprocessing steps we have applied for the employed CML and DL models.

A. PROBLEM FORMULATION

As mentioned in the introduction, in this paper, we tackle a multi-label classification problem. For each patient, we have his/her clinical records and a list of morbidity conditions that he/she may suffer from. Thus, we aim at identifying the presence or absence of morbidity conditions in the patients by analyzing their clinical records using bag-of-words and word embeddings in conjunction with CML and DL approaches.

Several approaches exist to tackle the multi-label problem [34]. A straightforward and widely used one is to decompose the multi-label problem into multiple binary classification tasks. This technique is named *binary relevance method* in the literature [35]. Another approach is to transform the multi-label problem into a single-label multi-class classification problem in which the classes are all label combinations. However, since we address the recognition of 16 morbidities in our work, the number of possible classes (i.e., co-morbidities) would be $2^{16} = 65,536$. We discarded this approach since we believe that the number of classes would be too large with respect to the size of our training set. Other more complex solutions exist, including using a multi-label ensemble classifier built from a committee of (single-label) multi-class classifiers or the use of customized machine learning algorithms adapted to the multi-label problem.

However, since our study's primary goal is to provide a comprehensive comparison of different ML approaches and feature extraction techniques, we believe that using a widely adopted and simple classification strategy is the most appropriate. For this reason, in this work, we adopt the binary relevance method, and we transform the multi-label classification task into sixteen binary classification problems.

B. DATASET DESCRIPTION

We performed our research study on the *n2c2*[2] dataset released for the *i2b2* obesity and co-morbidity detection challenge in 2008. The dataset was completely anonymized by replacing personal and sensitive information of patients with surrogates. The dataset contains clinical records of patients, and these records indicate that patients may have one or more morbidity conditions from a range of sixteen morbidity conditions (diseases). The sixteen morbidity conditions are *Asthma*, *CAD*, *CHF*, *Depression*, *Diabetes*, *Gallstones*, *GERD*, *Gout*, *Hypercholesterolemia*, *Hypertension*, *Hypertriglyceridemia*, *OA*, *Obesity*, *OSA*, *PVD*, and *Venous Insufficiency*.

Originally the *n2c2* dataset contained six documents, out of which four were Training Textual Judgments, Training Intuitive Judgments, Test Textual judgments, and Test Intuitive Judgments. They all were annotated. The remaining two documents, namely Training Obesity Patients Records and Test Obesity Patients Records, contained the clinical records and a unique *id* associated with them. The textual judgment documents contain all the sixteen morbidity conditions, and within each morbidity condition, there is a specific number of ids and labels associated with them. The labels in textual judgment documents can obtain values in {Y, N, U, Q}, where "Y" means yes, the patient has the morbidity, "N" means no, the patient does not have the morbidity, "U" means the morbidity is not mentioned in the record,

[2] https://portal.dbmi.hms.harvard.edu/projects/n2c2-nlp/

and “Q” stands for questionable whether the patient has the morbidity. Besides, intuitive judgment documents represent clinical records where domain experts (doctors) were able to infer if those were indicative of having one or more morbidity conditions for the underlying patients. Hence, possible intuitive judgments are limited to labels “Y,” “N,” and “Q” because “U” is irrelevant as an intuitive judgment. The length of the clinical records is in the range of 500 to 1200 words. A sample of each of the six annotated documents of the morbidity condition *Asthma* is shown in Table 1.

TABLE 1. Sample data of class Asthma.

Training Documents	Test Documents
Training Data-Textual Judgments <diseases source="textual"> <disease name="Asthma"> <doc id="1" judgment="U"/> <doc id="2" judgment="Y"/> <doc id="10" judgment="U"/> ...	*Test Data-Textual Judgments* <diseases source="textual"> <disease name="Asthma"> <doc id="3" judgment="Y"/> <doc id="5" judgment="U"/> <doc id="8" judgment="U"/> ...
Training Data-Intuitive Judgments <diseases source="intuitive"> <disease name="Asthma"> <doc id="1" judgment="N"/> <doc id="4" judgment="N"/> <doc id="10" judgment="Q"/> ...	*Test Data- Intuitive Judgments* <diseases source="intuitive"> <disease name="Asthma"> <doc id="3" judgment="Y"/> <doc id="5" judgment="N"/> <doc id="9" judgment="Y"/> ...
Training-Obesity Patients Records <doc id="1"> <text> 490646815 | WMC | 31530471 | | 9629480 | 11/23/2006 12:00:00 AM | ANEMIA | Signed | DIS | Admission Date: 11/23/2006 Report Status: Signed Discharge Date: 6/20/2006 ATTENDING: TRUKA , DEON XAVIER M.D. SERVICE: BH .anList Medical Center. PRIMARY DIAGNOSIS: Congestive heart failure. ...	*Test-Obesity Patients Records* <doc id="3"> <text> 470971328 | AECH | 09071283 | | 6159055 | 5/26/2006 12:00:00 AM | PNUEMONIA | Signed | DIS | Admission Date: 4/22/2006 Report Status: Signed Discharge Date: 7/27/2006 ATTENDING: CARINE , WALTER MD SERVICE: PRINCIPAL DIAGNOSIS: Anemia and GI bleed. ...

As each clinical record may have multiple associated morbidities, as mentioned earlier, we chose to tackle the multi-class classification problem as several binary classification problems. To do this, we have extracted all the clinical records having labels “Y” or “N” from the textual and intuitive judgment documents.

1) DATA PREPROCESSING

The dataset used for our experiments contained abbreviations, some typos, and punctuation, and some preprocessing steps were thus necessary. In the scope of our work, we have used two types of feature representations, namely bag-of-words and word embeddings. On the one hand, for bag-of-words, we have employed TF-IDF, whose vector representation relies on the word’s occurrence frequency. On the other hand, the word embeddings’ working principle is based upon capturing the semantic relationships among words. The works in [36], [37] discuss the process and impact of document preprocessing in NLP tasks. Accordingly, the preprocessing steps we have performed for transforming our input dataset to be used with the bag-of-words models are reported below:

- Lower-casing the text to represent the same words of different cases such as *Asthma* and *asthma* as one, i.e., *asthma*.
- Tokenization of text to build a function f, where for each word w, the function f is associated with an integer index i.
- Punctuation and numeric values removal from the text.
- Lemmatization of the tokens.
- TF-IDF matrix generation from input data to transform each clinical note into a feature vector.

In order to study the impact of stopword removal for the experiments with word embeddings representation, we have preprocessed the input data to generate two sets of feature vectors. One set of feature vectors contains the stopwords, while the other set does not. In the second case, stopword removal has been performed by using the NLTK[3] library. Furthermore, these two feature vectors are separately used to train the CML models to observe the impact of stopwords on the classifier’s performance.

2) STOPWORDS AND THEIR IMPACT IN TEXT PREPOCESSING

Stopwords are those words that commonly occur in a text. There are both advantages and disadvantages in including or excluding stopwords while preprocessing the data. The use of stopwords is debatable, and it is difficult to define one standard protocol that can be applied to all datasets. Therefore, the impact of stopwords is very much dependent on the data type and nature of the task at hand. A general observation in this context is that removing stopwords reduces the data size, the model’s training time and can also improve the model’s performance because stopwords removal may leave meaningful tokens in the dataset. In addition, stopwords such as negations in sentences are important indications for inferring certain behavior types in the context of sentiment analysis [38], [39] and motivational interviewing, empathetic conversations, etc., in therapeutic scenarios. For example, the following two sentences: *“The patient is **not** stable.” “The patient is **not** happy.”* The removal of the stopword ***not*** changes these to sentences that convey precisely the opposite meaning and emotion as compared to the original. Therefore, from these observations, it can be inferred that for tasks such as spam filtering, auto-tag generation, text, genre, language, and caption classification, removing stopwords is likely not going to degrade the classification model’s performance. On the other hand, for tasks such as machine translation, text summarization, identification of change talk and sustain talk in motivational interviewing, patient’s

[3] https://www.nltk.org/

treatment recommendations, the removal of stopwords can lead to underperformances of the training model. In this context, the authors in [40] observed a decrease in performances of SVM classification models from 70.76% to 55.26% for the task of automatic annotation of clinical text fragments based on codebooks having a large number of categories. Similarly, authors in [41], [42] also reported the underperformance of the employed classification models for text classification as a consequence of removing the stopwords. In the case of the DL models, we have used BiLSTM layers, which handle the long-term dependencies and have the capability to store information for a long duration. Therefore, given the presence of several tokens in our dataset behaving as the negation mentioned above and the ability of DL approaches in conjunction with word embeddings representations to tackle the contextual relationship of the words, we performed the experiments with the DL approaches only with the input dataset containing the stopwords.

3) TRANSFORMING INPUT DATA FOR TRAINING OF DL MODELS

We have used the DL models with bag-of-words and word embeddings representations described in Section IV-B. The DL models require the input data to be in integer encoded format, where each word is represented by a unique integer. Thus, each word can be mapped to the corresponding word vector using the embeddings layer. In addition to integer encoding, we have also padded the data to have symmetrical length throughout. The steps of encoding and padding the input data are mentioned below:

- Encoding the input texts into numeric integer representations using vocabulary-index relation. For instance, consider the sentence s: *the patient is asthmatic*, and a function f that maps *the* to "5", *patient* to "34", *is* to "10" and *asthmatic* to "87". Then, the resulting integer-encoded sentence $s_{encoded}$ will be [5, 34, 10, 87].
- Padding each of the input text (integer encoded) to a length equivalent to (average + standard deviation) number of tokens. Most clinical texts are around the average length for our dataset, and the very few remaining clinical texts are too long. Therefore, we have limited the number of tokens for each text sequence in order to reduce computational cost as well as keeping the dimension of input text reasonable. For our work, we have computed the padding length equal to the sum between the average and the standard deviation of the number of tokens each input text had. This formula has been found empirically on our data and turned out to be a good trade-off between the size of the padding and the length of the document. For example, for four clinical records with 25, 39, 44, and 80 tokens, respectively, the average length is $avg = 47$, and the standard deviation is $std = 20.29$. Hence, the length that we consider for padding is 67. Also, in the presence of very long clinical notes, which comprise 3.12% of the total dataset, we have broken them down into more notes with the same annotations. Although this last preprocessing step slightly augmented the dataset, it did not change the sixteen classes' unbalanceness distribution. In Table 2, we show the number of clinical notes and percentage of occurrences of each class before and after the preprocessing step.

TABLE 2. Percentage of occurrences of each class and number of clinical notes before and after the preprocessing step.

Morbidity	Before Preprocessing		After Preprocessing	
	#Clinical Notes	%Occ.	#Clinical Notes	%Occ.
Ashtma	952	15.13	984	15.24
CAD	942	60.72	972	60.08
CHF	904	48.89	936	49.15
Depression	968	23.14	1000	24.2
Diabetes	980	70.2	1012	70.55
Gallstones	996	17.47	1028	18.29
Gerd	858	22.84	880	22.95
Gout	1004	12.55	1034	12.38
Hypercholesterolemia	876	56.62	904	56.42
Hypertension	942	18.68	970	20.41
Hypertriglyceridemia	976	5.53	1006	5.37
OA	934	21.63	960	22.08
Obesity	930	44.52	960	44.58
OSA	994	14.08	1026	14.62
PVD	938	14.93	968	15.5
Venos Insufficiency	858	7.23	882	7.26

IV. FEATURES REPRESENTATIONS

We have used bag-of-words TF-IDF and word embeddings representations to generate feature vectors. On the one hand, TF-IDF has served as a baseline for many NLP tasks [43] for decades and has proven to be very useful. On the other hand, word embeddings are the current state-of-the-art due to their innate capability of capturing the semantics and contextual information for textual features representation of words and text sequences [44], [45].

A. TF-IDF

TF-IDF is a feature extraction technique that calculates the weight for each word based on its frequency within a document. In document d, TF defines the occurrence of a word w. In the entire document, IDF measures the rarity of a word w. Equation (1) shows the TF-IDF formula of w in a document d_i where c_i^w is the frequency of the word w in the i-th document d_i, $|d_i|$ is the size of the document expressed as the number of words, n is the number of documents in the collection, and n^w is the number of documents where the word w occurs at least once. TF-IDF values are usually normalized in the range [0,1].

$$TF - IDF(w, d_i) = \frac{c_i^w}{|d_i|} \cdot log\frac{n}{n^w} \tag{1}$$

To generate the feature vectors using bag-of-words TF-IDF representation, we have used the TF-IDF Vectorizer[4] from the scikit-learn library. We have performed the experiments

[4] https://tinyurl.com/y8jqmscd

with four types of feature vectors using the TF-IDF representations: **All Features** (where feature selection is not applied) and the ones obtained by applying three feature selection algorithms: **ExtraTreesClassifier**, **InfoGainAttributeEval**, and **SelectKBest**. The reason for limiting the number of features is to reduce the computational time for training the models by keeping only those features that contribute most in distinguishing the instances of the different classes. Feature selection also has the effect of disregarding those terms that are irrelevant and may confuse the classifier or determine overfitting.

- **ExtraTreesClassifier** is essentially an ensemble learning method that conceptually shares a similar working principle as that of Random Forest. The only difference is the method for constructing decision trees. For a given set of m features, which are selected randomly from the features set of the input data, ExtraTreesClassifier[5] selects the top features based on their importance (it can be typically calculated by the Gini Index [46]). These random samples of features are further used to create decision trees which are mutually correlated. This process helps to minimize the chances of overfitting and ranks the features in descending order.
- **InfoGainAttributeEval** is used for feature selection based upon measuring how each feature contributes to decreasing the overall entropy [47]. Entropy is basically a measure of the impurity degree in the dataset. The data is characterized as less impure when the entropy is closer to zero. Hence, the usefulness of an attribute is identified by its contribution to reduce the overall entropy. It can be represented by:

$$\begin{aligned} &InfoGain(Class, Attribute) \\ &\qquad = H(Class) - H(Class|Attribute) \end{aligned} \tag{2}$$

  where H is the information entropy.
- **SelectKBest** takes the score function as a parameter, which is applied to a pair (m, y) where m corresponds to the features of the input data and y to the corresponding labels. The score function returns an array of scores, one for each feature $m[:, i]$ of m. SelectKBest[6] then simply retains the first k features of m with the highest scores.

The parameter *vocabulary* of the TF-IDF vectorizer should be provided with a custom list of words (vocabulary) to use the feature selection algorithms from the python library. This custom vocabulary contains the words (features) in ranked order provided by feature selection algorithms based upon the features' information gain. We have set the configuration to $max\_features = 600$ and $vocabulary = custom\_vocab$, where $custom\_vocab$ is the vocabulary of ranked features selected by applying the feature selection algorithms. This setting generates the feature vectors matrix of $\{n \times 600\}$ dimension, where n is the number of text documents (clinical notes).

B. WORD EMBEDDINGS

This section describes the general working principle of the word embeddings followed by the details of all the word embeddings used for our experiments: pre-trained word2vec, domain-trained, GloVe, fastText, and USE embeddings. They are reported below. Word embeddings are distributed representations that model words' properties into vectors of real numbers in a predefined vector space, capturing features and preserving their semantic relationships. As an outcome of this representation, the words having similar meanings have a similar representation. In Figure 1, we have presented the visualization of 300-dimensional word embeddings of 18586 words generated from our dataset using the word2vec model in high dimensional space using Tensorboard.[7] From the visualization, one can note how the words are mapped near to those whose word embeddings have a similar meaning. For instance, in the case of the word *diabetes*, the words *diabetic* and *insulinotherapy* are represented in the close semantic space, notable by their scores 0.772 and 0.777.

- **Pre-trained Word2Vec** *Word2Vec* is an algorithm invented by Google for training word embeddings that relies on the distributional hypothesis [48]. The distributional hypothesis uses skip-gram or Continuous Bag of Words (CBOW) algorithms. In the CBOW model, for a given context, the objective is to predict the focal word. The CBOW model with a softmax loss function is essentially a log-linear classification model. The aim is to determine the most likely parameters of the embedding vectors, which can be represented by Equation 3:

$$P(w_f|w_c) = \frac{exp(w_f^T w_c)}{\sum_{i=1}^{V} exp(w_f^T w_c)} \tag{3}$$

  where w_c is the context (one or more words), w_f is the focal word, and V is the vocabulary size. On the other hand, the skip-gram model can be considered as a complementary model to the CBOW model in terms that its objective involves predicting a context word given a single focal word [24]. The skip-gram model is represented by Equation 4:

$$P(w_f|w_c) = \sum_{c=1}^{C} \frac{exp(w_f^T w_c)}{\sum_{i=1}^{V} exp(w_f^T w_c)} \tag{4}$$

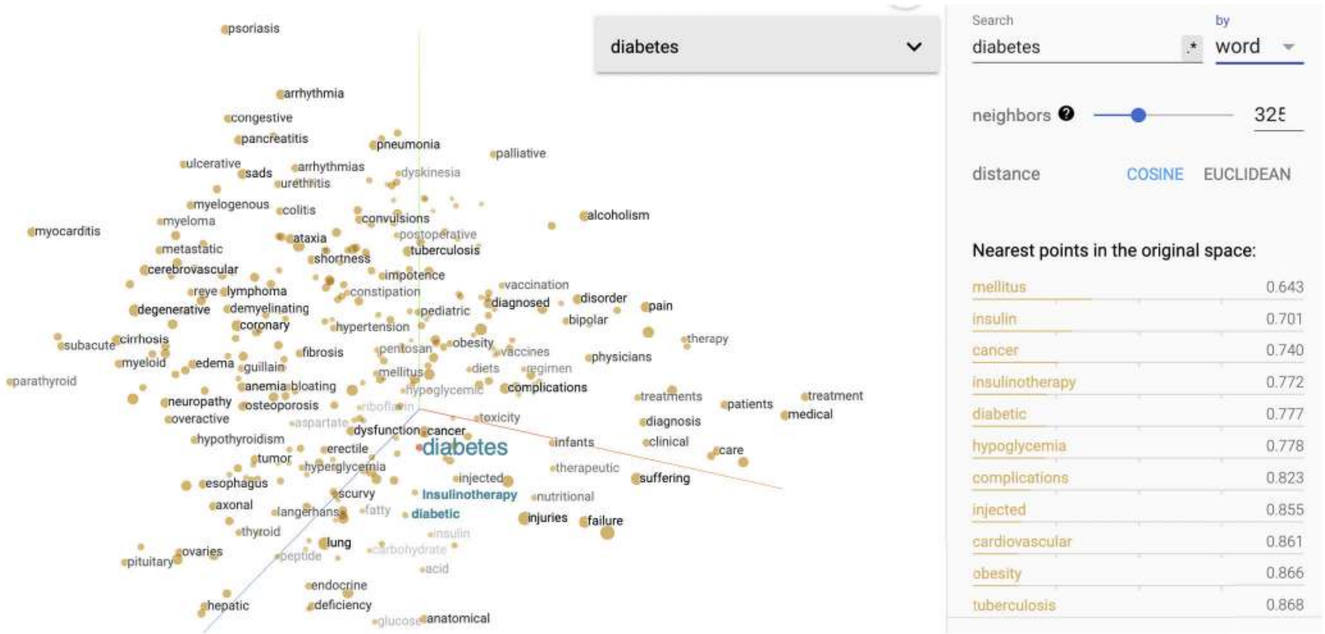
  The *Word2Vec* algorithm aims to detect the meaning and semantic relations by studying the co-occurrences among words in a given corpus. We have used the pre-trained Word2Vec[8] model, which is trained on the part of the Google News dataset (about 100 billion words). This pre-trained model contains vectors of three million words and phrases, which are represented in 300-dimensional space.
- **Domain-trained Word2Vec** The domain-trained word embeddings are generated by using the *Word2Vec* algorithm on the *n*2*c*2 dataset. The rationale of using these

[5] https://tinyurl.com/ybnzo8rh

[6] https://tinyurl.com/y5c7w6bo

[7] https://projector.tensorflow.org/

[8] https://code.google.com/archive/p/word2vec/



diabetes

Search: diabetes .* by word

neighbors 325

distance COSINE EUCLIDEAN

Nearest points in the original space:

mellitus	0.643
insulin	0.701
cancer	0.740
insulinotherapy	0.772
diabetic	0.777
hypoglycemia	0.778
complications	0.823
injected	0.855
cardiovascular	0.861
obesity	0.866
tuberculosis	0.868

FIGURE 1. Visualizing the semantic relationships between words by Word2Vec word embeddings representation.

embeddings is their advantage in representing the out-of-vocabulary words due to training on the target domain (in our case, healthcare). We have generated the word embeddings of 300 dimensions with 10 epochs and a window size of 5 by using the Gensim[9] library.

- **GloVe** generator algorithm was developed as an open-source project at Stanford in 2014 [49]. For a given context, to identify how frequently the words appear, *GloVe* utilizes a statistics-based matrix to compute the vectors' scores based on the co-existence of words within the context. Unlike the *Word2Vec* algorithm, *GloVe* uses both the skip-gram model, which is a local context window, and the latent semantic analysis method, which belongs to the global matrix factorization methods. For our work, we have used the pre-trained *GloVe6B*[10] embeddings model, trained by the Stanford NLP Group on 600 billion tokens of Wikipedia[11] and Gigaword[12] with dimension 300.
- **fastText** One drawback of *Word2Vec* and *GloVe* algorithms is the fact that they are not able to handle out-of-vocabulary words. To overcome this limitation, Facebook proposed *fastText*,[13] which is essentially an extension of the *Word2Vec* algorithm [50]–[52]. *FastText* extends the *Word2Vec* skip-gram model by considering internal sub-word information. Basically, words are represented as n-gram of characters instead of learning vectors for words directly. For instance, for $n = 3$, the word *apple* consists of *app*, *ppl*, and *ple*. *FastText* does not consider the internal structure of the word and represents a bag-of-words model with a sliding window over a word. Also, as long as the characters are contained in the window, it is unaffected by the order of the *in*-grams. This approach helps the model to compute word representations of out-of-vocabulary words and allows the model to understand suffixes and prefixes because it is very likely that some of the n-grams also appears in other words.
- **Universal Sentence Encoder** (USE). While the common practice with the word embeddings focuses on representing the word, the technique to represent the sentence through a single vector is unclear. To address this, Google introduced pre-trained embeddings models known as *USE*, which are optimized to train with a longer text sequence than a single word such as phrases, sentences, and short paragraphs [53], [54]. The pre-trained USE[14] model is trained on several domains with a variety of data sources to dynamically accommodate a wide variety of natural language understanding tasks. It transforms the text into high-dimensional vectors by performing an encoding. It comes with two variations, i.e., one trained with transformer encoder and the other trained with the deep averaging network. For our work, we have used the deep averaging network pre-trained USE, which takes variable-length English texts as input and outputs 512-dimensional vectors.

V. CLASSIFICATION MODELS

We have used two types of classification models based on CML and DL approaches with each type of feature

[9] https://radimrehurek.com/gensim/

[10] https://nlp.stanford.edu/projects/GloVe/

[11] https://dumps.wikimedia.org/enwiki/

[12] https://catalog.ldc.upenn.edu/LDC2011T07

[13] https://fastText.cc/docs/en/english-vectors.html

[14] https://tfhub.dev/google/universal-sentence-encoder/4

representation mentioned in Section IV. Figure 2 shows the generalized architecture of the pipeline used for the classification of clinical records using TF-IDF representations with CML and DL approaches.

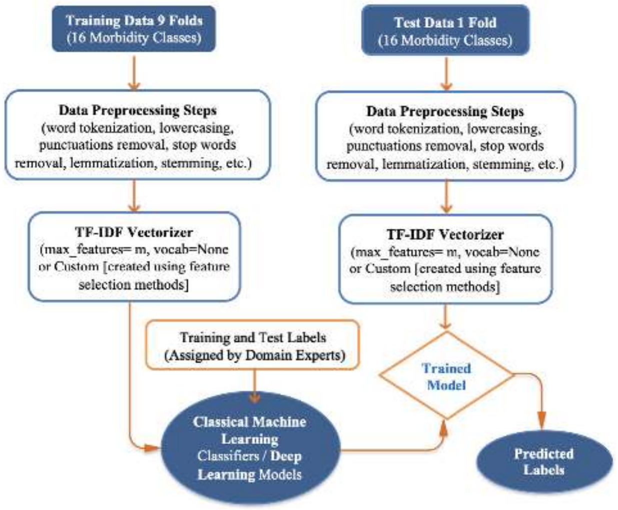

Training Data 9 Folds (16 Morbidity Classes)
Test Data 1 Fold (16 Morbidity Classes)
Data Preprocessing Steps (word tokenization, lowercasing, punctuations removal, stop words removal, lemmatization, stemming, etc.)
Data Preprocessing Steps (word tokenization, lowercasing, punctuations removal, stop words removal, lemmatization, stemming, etc.)
TF-IDF Vectorizer (max_features= m, vocab=None or Custom [created using feature selection methods]
TF-IDF Vectorizer (max_features= m, vocab=None or Custom [created using feature selection methods]
Training and Test Labels (Assigned by Domain Experts)
Trained Model
Classical Machine Learning Classifiers / Deep Learning Models
Predicted Labels

FIGURE 2. The architecture of the pipeline for morbidity detection in clinical records using TF-IDF representations with CML and DL approaches.

The pipeline consists of training and testing phases. Prior to the training stage, we preprocess the clinical records, as mentioned in Section III-B1. After that, classifiers are trained on the feature vectors derived from the training samples. After creating feature vectors, the previously trained classifiers predict each clinical record label in the testing sample. Finally, the performances of different classifiers are evaluated by calculating standard metrics such as precision, recall, and F-1 score. In the following sections, we will list the CML algorithms used for our experiments, followed by the DL models and their architectures.

A. CLASSICAL MACHINE LEARNING MODELS

Experimental results reported in this paper were obtained using standard implementations of CML algorithms provided by the Weka toolkit using Python Weka-Wrapper[15] interface with Java Virtual Machine[16] environment. We have employed Support Vector Machine (SVM) [55], k-Nearest Neighbours (kNN) [56], Naive Bayes [57], Random Forest [58], Random Tree [59], J-48 [60] and J-Rip [61].

B. DEEP LEARNING MODELS

We have used DL models with two types of representations, one with word embeddings and the other with bag-of-words.

1) DEEP LEARNING MODELS USED WITH WORD EMBEDDINGS

The DL model we used for word embeddings representations is the network with an embeddings layer, two BiLSTM layers, a dense layer followed by an output layer for the binary classification task. Figure 3 presents the related architecture. The embeddings layer is initialized by the following four inputs:

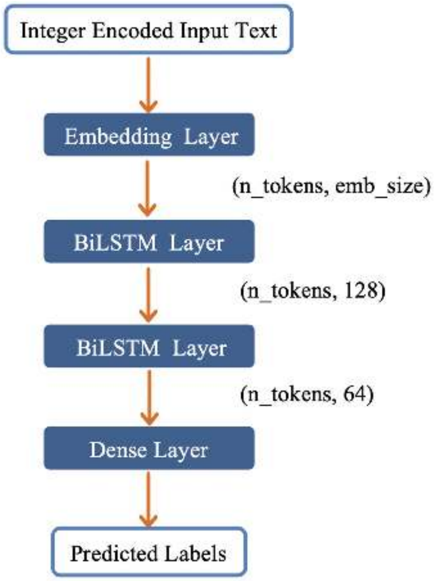


FIGURE 3. The architecture of DL models to use word embeddings representation.

- *input_dim* (size of the vocabulary);
- *output_dim*: (dimension of the dense embeddings);
- *weights (embeddings_matrix),* and
- *input_length* (length of input sequences).

The *input_dim* represents the length (V) of the unique vocabulary made from our input data (clinical records). The input matrix (integer encoded vectors) has dimension $\{n \times m\}$, with n equal to the number of clinical records and *input_length* corresponding to m, which is the maximum number of tokens considered for each text. The *embeddings_matrix* is the vector representation of the corresponding words of the vocabulary and has dimension $\{V \times x\}$, where x represents the *output_dim*. Specifically, *output_dim* for all the embeddings is 300 except USE, which has a value of 512. The output of the embeddings layer is passed to two hidden layers that implement BiLSTM neural networks [62]. LSTM is a particular kind of recurrent neural network that can store the history of the input data and has already proven to be able to find patterns in data where the sequence of the information matters [63]. By using the bidirectional version, the models can learn from the input data both backward and forward. Finally, the output of the BiLSTM layer is fed to a fully connected dense layer to predict the labels.

2) DEEP LEARNING MODELS USED WITH BAG-OF-WORDS REPRESENTATION

For the bag-of-words model, in conjunction with the employed feature selection algorithms, we used TF-IDF representation. The differences between the neural network

[15] https://pypi.org/project/python-weka-wrapper/
[16] https://pypi.org/project/javabridge/

model used here with that described in the previous paragraph are the following:

- Firstly, the one described here does not have an embeddings layer, and the input is directly fed to the BiLSTM layer.
- Secondly, the input data do not undergo the preprocessing steps such as integer encoding and padding when used with TF-IDF representation.

The input to the BiLSTM layer, in this case, is the TF-IDF matrix, which is generated by the TF-IDF vectorizer and has dimension $\{n \times 600\}$, with n the number of text documents (clinical records). Figure 4 presents the architecture of the DL network used with TF-IDF representation.

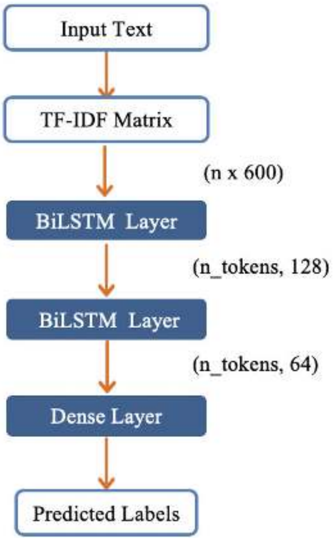


FIGURE 4. The architecture of DL models to use TF-IDF representation.

VI. EXPERIMENTS AND RESULTS

The server specifications we have used to develop our methods and run the experiments are summarized in Table 3.

TABLE 3. Server specifications.

Item	Specification
CPU	Intel Core i3-7100 (-HT-MCP-) CPU @ 3.90 GHz
GPU	NVIDIA GP102 [TITAN X], 12 GB memory
Graphic driver	NVIDIA graphic driver version 440.33.01
CUDA	Version 10.2
OS	Ubuntu (17.10)
Python	Version 3.6.6

We have conducted our experiments with CML and DL approaches using the bag-of-words applied to feature selection algorithms and word embeddings representations. We have also employed ensemble learning over a large number of combinations of classifiers to improve the single model performances and obtain stable results.

We used 10-fold cross-validation as an experimental design [64] to ensure the robustness of performance estimation and avoid the bias of our single models. The performances of different classifiers and feature representations were measured in terms of F-1 score (F-1) using micro and macro averaging over 10 folds provided by the scikit-learn[17] library. The formulas to calculate precision, recall, and F-1 score are given by:

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{7}$$

where TP, FP, and FN represent true positive, false positive, and false negative of each label, respectively. The experiments that have been carried out can be divided into three groups for ease of understanding, which are mentioned below:

- In the first set of experiments, CML and DL approaches were used with bag-of-words representations coupled with feature selection algorithms using TF-IDF representation, as mentioned in Section IV-A.
- In the second set of experiments, we used CML and DL approaches with word embeddings generated by pre-trained models of word2vec, domain-trained with word2vec, GloVe, fastText, and USE embeddings. The feature vectors generated by these pre-trained word embeddings to train CML classifiers were generated from the same input data by either keeping the stopwords or removing them. The purpose of generating two sets of feature vectors was to study the relatedness of stopwords with the context of the text and their impact on the classifier's performance. The DL models were trained only with the feature vectors of the input data with stopwords as the standard experiment.
- As the last set of experiments, we implemented ensemble learning techniques on a large number of combinations of classifiers to improve the single model performances.

The following subsections describe the three sets of experiments.

A. EXPERIMENTAL RESULTS WITH BAG-OF-WORDS COUPLED WITH FEATURE SELECTION ALGORITHMS

This section provides the details of the experiments performed with CML and DL approaches with bag-of-words coupled with feature selection algorithms using TF-IDF representations. TF-IDF evaluates the importance of a feature based on its frequency. Identifying features that contribute the most to distinguish the classes is useful for improving the models' performances. Thus we have adopted three feature selection algorithms, namely ExtraTreesClassifier, InfoGainAttributeEval, and SelectKBest, along with the All Features. Table 4 depicts the results of CML classifiers with

[17] https://tinyurl.com/y4mt646z

TABLE 4. Performances of CML classifiers with all features using TF-IDF representations.

Morbidity Class	J-48		J-Rip		Naive Bayes		Random Forest		Random Tree		SVM		KNN (k=1)		KNN (k=5)	
	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1
Asthma	**99.4**	**98.75**	98.4	97	91.1	81.6	**99.4**	**98.75**	98.9	97.95	98.7	97.55	98.1	96.45	89.6	76.65
CAD	93.6	93.4	94.3	94.05	88.4	87.95	97.7	97.55	96.4	96.2	**99.2**	**99.1**	96	95.8	70.8	67.45
CHF	96	96.05	94	94.05	86.9	86.95	97.8	97.75	96.2	96.2	**98.9**	**98.9**	94.7	94.7	68	64.45
Depression	95.4	93.3	95.1	93.15	84.7	76.85	96.3	94.45	95.2	93.2	**97**	**95.75**	96.7	95.2	79.5	59.85
Diabetes	95.7	94.85	96	95.25	89.3	86.9	96.3	95.45	96.5	95.8	**96.9**	**96.25**	93.5	92.6	73.2	71.75
Gallstones	99.1	98.45	99.4	98.95	89.4	82.05	98.8	97.85	98.4	97.15	**99**	**98.2**	98.8	97.85	83.4	51.75
Gerd	97.3	96.15	96.4	94.95	88.1	83.25	98.1	97.25	97.4	96.35	**97.9**	**96.95**	97.2	96.05	82.5	68.7
Gout	**99.7**	**99.3**	**99.7**	**99.3**	95.2	89.2	99.2	98.1	98.4	96.45	99.6	99.1	98.7	97.15	90.2	68.2
Hypercholesterolemia	97.4	97.35	91.7	91.55	82.9	82.55	97.5	97.4	95.7	95.6	**97.7**	**97.7**	95.7	95.6	78.9	78.2
Hypertension	**97.8**	**96.25**	95.5	92.75	86.4	76.9	97.7	95.95	97.5	95.7	97.6	96.1	95.2	92.5	85.1	72.1
Hypertriglyceridemia	98.2	89.95	98.8	94.1	96.7	86.6	**99.4**	**96.9**	**99.4**	**96.9**	**99.4**	**96.9**	99.2	95.95	94.5	48.6
OA	97.8	96.8	97.1	95.75	88.7	83.2	97.6	96.35	97	95.55	**98.5**	**97.75**	96.1	94.4	75.1	65.2
Obesity	**99**	**99**	97.6	97.6	80.9	80.75	96.8	96.75	96.1	96.1	97.2	97.15	96.3	96.25	76	74.85
OSA	99.5	98.95	**99.8**	**99.6**	89.2	76.55	98.6	96.95	97.8	95.35	98.8	97.4	98.5	96.8	89.4	72.65
PVD	98.1	96.25	98.7	97.45	92.1	82.05	98.5	96.9	98.3	96.65	98.9	97.85	**99.4**	**98.7**	90.1	73.65
Venous Insufficiency	97.3	90.1	97.4	90.85	99	96.3	**100**	**100**	99.5	98.3	**100**	**100**	98.8	95.95	94.5	68.05
Average	97.58	95.93	96.86	95.39	89.31	83.72	98.1	97.14	97.41	96.21	**98.45**	**97.66**	97.05	95.74	82.55	67.63

TABLE 5. Performances of CML classifiers with feature selection algorithm ExtraTreesClassifier using TF-IDF representations.

Morbidity Class	J-48		J-Rip		Naive Bayes		Random Forest		Random Tree		SVM		KNN (k=1)		KNN (k=5)	
	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1
Asthma	99.1	98.15	99.1	98.15	73.7	67	99.2	98.35	98.3	96.75	99.2	98.35	**99.4**	**98.75**	86.2	54.55
CAD	95.5	95.35	94.7	94.45	93	92.7	98.5	98.45	97	96.9	**99.6**	**99.55**	97	96.9	83.4	81.05
CHF	96	96	97.2	97.25	86	85.95	98.9	98.9	97.3	97.35	**99.1**	**99.1**	98.2	98.25	86	85.8
Depression	94.7	92.45	96.5	95.1	75.1	72.75	96.9	95.4	96.5	94.95	**99.2**	**98.85**	96.3	94.45	78.1	48.85
Diabetes	96.7	96.1	95.6	94.75	93.8	92.75	**98**	**97.55**	96.9	96.35	97.8	97.3	96.7	96	79.2	67.5
Gallstones	97.6	95.75	99.1	98.45	70.7	66.4	99	98.2	99.4	98.95	**99.8**	**99.65**	99	98.2	83.4	50.35
Gerd	97.3	96.1	95.8	94.15	76	73.6	98.8	98.35	97.4	96.4	**99.8**	**99.65**	98.4	97.65	78.2	48.2
Gout	99.4	98.65	99.8	99.55	85.3	76.9	99.8	99.55	99.2	98.2	**100**	**100**	99.2	98.1	87.6	48.25
Hypercholesterolemia	95.3	95.25	91.7	91.6	87.8	87.45	**98.4**	**98.35**	97.5	97.45	98.2	98.15	97.9	97.95	82.3	81.95
Hypertension	97.5	95.7	96	93.4	71.4	67.1	97.9	96.35	97	95.15	**99.2**	**98.6**	97.2	95.25	82.6	52.45
Hypertriglyceridemia	99.2	96.1	98.7	94	91.4	75.25	**99.6**	**98**	99.6	98	99.4	96.9	99.4	96.9	94.5	48.6
OA	97.5	96.35	96.7	95.2	75.4	72.45	98.9	98.4	96.8	95.25	**99.1**	**98.75**	98.5	97.75	80.3	52.6
Obesity	98.2	98.15	97.6	97.6	87.7	87.75	98.9	98.9	96.1	96.1	**99.4**	**99.35**	95.9	95.85	79.6	78.3
OSA	99.3	98.55	**99.8**	**99.6**	79.1	71.25	99.4	98.7	98.6	97.1	99.4	98.7	98.6	96.95	86.8	52.5
PVD	98.1	96.2	97	94.2	77.1	70.1	**98.9**	**97.85**	98.5	97	**98.9**	**97.85**	98.5	96.9	85.7	50.25
Venous Insufficiency	98.1	92.7	97.8	92.15	86.5	71.5	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	92.8	48.15
Average	97.47	96.10	97.07	95.60	81.88	76.93	98.82	98.21	97.88	96.99	**99.26**	**98.80**	98.14	97.24	84.17	59.33

TABLE 6. Performances of CML classifiers with feature selection algorithm SelectKBest using TF-IDF representations.

Morbidity Class	J-48		J-Rip		Naive Bayes		Random Forest		Random Tree		SVM		KNN (k=1)		KNN (k=5)	
	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1
Asthma	**99.4**	**98.75**	98.8	97.8	91.5	82.55	**99.4**	**98.75**	98.9	97.95	98.9	97.95	97.9	96	89	75.05
CAD	94.8	94.6	91.2	90.9	88.7	88.25	97.2	97.1	94.9	94.65	**99.2**	**99.1**	95.8	95.55	69.2	65.35
CHF	96.2	96.25	95.7	95.7	86.9	86.95	98.2	98.25	96	96.05	**99.1**	**99.1**	95.4	95.35	69.2	65.9
Depression	95.9	94.05	96.1	94.6	84.7	77.2	96.3	94.45	95.5	93.5	**97.1**	**95.75**	95.5	93.55	79.8	60.7
Diabetes	96.1	95.35	95.2	94.35	89.7	87.4	96.5	95.7	94.3	93.2	**97.1**	**96.55**	92.4	91.4	70.1	69.2
Gallstones	**99**	**98.25**	**99**	**98.25**	89.3	81.9	98.8	97.85	97.6	95.85	**99**	98.2	98.8	97.85	83.3	53.05
Gerd	97	95.65	96.6	95.3	88.2	81.25	**98.1**	**97.25**	96.7	95.4	97.7	96.6	97.4	96.35	83.4	70.4
Gout	99.6	99.1	**99.7**	**99.3**	94.9	88.45	99.2	98.1	98.8	97.25	99.6	99.1	98.8	97.35	89.8	70.05
Hypercholesterolemia	97	97	90.6	90.5	83	82.65	97.5	97.45	95	94.85	**97.7**	**97.7**	94.7	94.7	78.9	78.55
Hypertension	97.5	95.75	93.9	90.4	86.5	77.05	**97.7**	95.95	95.8	93	**97.7**	**96.1**	96.6	94.5	86.8	72.85
Hypertriglyceridemia	98.4	91.4	98.8	94.4	96.6	86.3	99.4	96.9	99.4	96.9	99.4	96.9	**99.6**	**98**	94.5	48.6
OA	97.2	95.95	97	95.7	88.2	82.55	97.6	96.35	97	95.55	**98.3**	**97.4**	96.4	94.75	77.8	67.95
Obesity	**98.8**	**98.8**	97.7	97.75	80	79.85	96.3	96.25	95.3	95.2	97	96.95	96.1	96.05	76.7	76.35
OSA	99.5	98.95	**99.8**	**99.6**	89.3	76.7	98.6	96.95	97.8	95.5	98.8	97.4	98.2	96.3	90.7	78.65
PVD	98	96	97.4	94.95	91.8	81.55	98.5	96.9	98.5	97.05	98.9	97.85	**99**	**98.05**	90	73.2
Venous Insufficiency	97.6	90.65	97.3	90.4	98.5	94.85	**100**	**100**	99.8	99.15	**100**	**100**	98.6	95.2	95.2	74.05
Average	97.63	96.03	96.55	94.99	89.24	83.47	98.08	97.14	96.96	95.69	**98.47**	**97.67**	96.95	95.68	82.78	68.74

All Features using TF-IDF representations. Tables 5, 6, and 7 illustrate the results of CML Classifiers with feature selection algorithms ExtraTreesClassifier, SelectKBest, and InfoGain, respectively. Finally, Table 8 includes the results of the DL models with the four bag-of-words applied to feature selection algorithms using TF-IDF representations. The key observations from the performed experiments are listed below:

TABLE 7. Performances of CML classifiers with feature selection algorithm InfoGain using TF-IDF representations.

	J-48		J-Rip		Naive Bayes		Random Forest		Random Tree		SVM		KNN (k=1)		KNN (k=5)	
Morbidity Class	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1
Asthma	**99.4**	**98.75**	98.8	97.8	91.5	82.55	**99.4**	**98.75**	98.9	97.95	98.9	97.95	97.9	96	89	75.05
CAD	94.8	94.6	91.2	90.9	88.7	88.25	97.2	97.1	94.9	94.65	**99.2**	**99.1**	95.8	95.55	69.2	65.35
CHF	96.2	96.25	95.7	95.7	86.9	86.95	98.2	98.25	96	96.05	**99.1**	**99.1**	95.4	95.35	69.2	65.9
Depression	95.5	93.45	96.1	94.6	84.7	77.2	96.3	94.45	95.5	93.5	**97.1**	**95.75**	95.5	93.55	79.8	60.7
Diabetes	95.9	95.15	95.2	94.35	89.7	87.4	96.5	95.7	94.3	93.2	**97.1**	**96.55**	92.4	91.4	70.1	69.2
Gallstones	97.4	95.25	99	98.25	89.3	81.9	98.8	97.85	97.6	95.85	**99**	**98.2**	98.8	97.85	83.3	53.05
Gerd	96.3	94.6	96.6	95.3	88.2	83.35	**98.1**	**97.25**	96.7	95.4	97.7	96.6	97.4	96.35	83.4	70.4
Gout	99.6	99.1	**99.7**	**99.3**	94.9	88.45	99.2	98.1	98.8	97.25	99.6	99.1	98.8	97.35	89.8	70.05
Hypercholesterolemia	97	97	90.6	90.5	83	82.65	97.9	97.9	95	94.85	**99.6**	**99.1**	94.7	94.7	78.9	78.55
Hypertension	97.2	95.35	93.9	90.4	86.5	77.05	97.5	95.65	95.8	93	**97.7**	**96.1**	96.6	94.5	86.8	72.85
Hypertriglyceridemia	97.7	87.45	98.8	94.4	86.5	77.05	99.4	96.9	99.4	96.9	99.4	96.9	**99.6**	**98**	94.5	48.6
OA	97.2	95.95	97	95.7	88.2	82.55	97.6	96.35	97	95.55	**98.3**	**97.4**	96.4	94.75	77.8	67.95
Obesity	**98**	**97.95**	97.7	97.75	80	79.85	96.3	96.25	95.3	95.2	97	96.95	96.1	96.05	76.7	76.35
OSA	99.1	98.1	**99.8**	**99.6**	89.3	76.7	98.6	96.95	97.8	95.5	98.8	97.4	98.2	96.3	90.7	78.65
PVD	97.5	95.1	97.4	94.95	91.8	81.55	98.5	96.9	98.5	97.05	98.9	97.85	**99**	**98.05**	90	73.2
Venous Insufficiency	97.6	90.65	97.3	90.4	98.5	94.85	**100**	**100**	99.8	99.15	**100**	**100**	98.6	95.2	95.2	74.05
Average	97.28	95.29	96.55	94.99	88.61	83.02	98.10	97.15	96.96	95.69	**98.59**	**97.75**	96.95	95.68	82.78	68.74

TABLE 8. Performances of DL models with bag-of-words coupled with feature selection algorithms using TF-IDF representations.

	Extra Tress Classifier		InfoGain		SelectKBest		All Features	
Morbidity Class	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1
Asthma	90.86	80.25	**92.22**	**82.14**	84.86	45.89	84.87	45.90
CAD	**83.02**	**82.33**	67.85	54.05	60.51	38.35	60.09	43.16
CHF	**86.3**	**85.91**	74.01	69.64	52.98	40.37	54.20	53.38
Depression	**86.15**	**76.46**	81.41	66.7	76.86	43.43	76.86	43.45
Diabetes	**86.93**	**83.54**	80.2	72.35	70.2	41.21	70.20	41.24
Gallstones	**96.08**	**90.66**	85.24	58.12	82.52	45.2	82.52	45.21
Gerd	**90.9**	**86.26**	80.18	58.74	77.15	43.5	77.15	43.55
Gout	95.61	**86.4**	**95.91**	86.08	87.45	46.62	87.45	46.65
Hypercholesterolemia	**78.64**	**76.12**	67.07	60	56.61	35.96	56.84	51.45
Hypertension	**81.31**	**49.54**	79.2	47.86	**81.31**	44.53	**81.31**	44.84
Hypertriglyceridemia	**97.13**	**83.71**	94.56	50.02	94.46	48.56	94.46	48.57
OA	**90.35**	**81.51**	85.22	75.04	78.37	43.91	78.37	43.93
Obesity	**94.83**	**94.69**	72.36	66.57	55.48	35.53	55.48	35.68
OSA	**97.88**	**95.08**	94.66	88.32	85.91	46.2	85.91	46.21
PVD	**85.72**	**67.9**	85.08	64.62	85.07	45.96	85.07	45.96
Venous Insufficiency	92.3	**53.39**	**92.77**	48.08	**92.77**	48.08	**92.77**	48.12
Average	**89.63**	**79.61**	83.00	65.52	76.41	43.33	76.47	45.45

- In general, the feature selection algorithms have improved the performance of CML classifiers (typically by 1%). The two best-performing classifiers with All Features are SVM and Random Forest with 98.45 and 98.1 micro F-1 scores, respectively (as shown in Table 4). The use of the ExtraTreesClassifier as the feature selection algorithm has improved the micro F-1 score of Random Forest to 98.82 and SVM to 99.26 (shown in Table 5), which is the best performance of CML classifiers among all the experiments.
- In contrast, the Naive Bayes classifier used with All Features has performed the best with a Micro F-1 score of 89.31 (as shown in Table 4) than with any feature selection algorithms.
- In the case of DL approaches, All Features using TF-IDF has been outperformed by the feature selection algorithms achieving up to 13% of F-1 score (shown in Table 8).
- The reason for the low performance of DL models with All Features using TF-IDF is because that TF-IDF selects the features based on the frequency of the words, not useful to distinguish the morbidity classes. Feature selection algorithms identify the most important features that allow the DL models to learn the context of clinical records, and this results in further improvement of the classification performances.
- From our experimental results, it turned out that the adoption of feature selection algorithms has shown more benefit on DL models than on CML algorithms. In fact, with All Features, the micro F-1 score of DL models was 76.47, whereas, with the usage of ExtraTreesClassifier, it has improved to 89.63 (as shown in Table 8).
- As far as the computational time and resource requirements are concerned, the CML models have proven to be computationally faster and less demanding for what resources are concerned. The training time of the CML models seen so far was up to 600 seconds; that of the DL models was much higher (a couple of hours) using the same machine mentioned in Table 3 (DL approaches employed both the CPU and the GPU whereas the CML models just the CPU).

B. EXPERIMENTAL RESULTS WITH WORD EMBEDDINGS

In this other group of experiments, we have trained the CML and DL approaches with the embeddings generated by the pre-trained word2vec, domain-trained with word2vec,

TABLE 9. Performances (averaged over all the morbidity classes) of CML classifiers with word Embeddings when input data contain stopwords.

	Domain-Train		**fastText**		**GloVe**		**Word2Vec**		**USE**	
Classifier	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1
J-48	92.21	87.79	93.34	89.65	92.95	88.84	92.62	86.39	**93.42**	**89.8**
J-Rip	86.12	77.87	**90.16**	**85.49**	89.16	83.7	88.76	80.92	89.56	83.83
Naive Bayes	58.14	51.5	65.89	**63.83**	63.51	60.84	61.96	60.47	**68.13**	60.33
Random Forest	97.92	96.95	98.03	96	97.98	95.9	96.63	92.85	**98.06**	**97.1**
Random Tree	97.19	95.69	**97.5**	**96.33**	97.1	95.68	95.87	92.88	96.98	95.52
SVM	79	51.2	89.08	78.07	86.13	70.78	87.31	71.85	**90.06**	**85.31**
KNN (k=1)	97.18	95.74	**97.51**	**96.07**	97.18	95.82	95.95	93.04	97.31	95.95
KNN (k=5)	81.79	65.52	**84.12**	68.14	83.38	66.76	83.13	65.71	83.16	**68.4**

TABLE 10. Performances (averaged over all the morbidity classes) of CML classifiers with Word Embeddings when input data do not contain stopwords.

	Domain-Train		**fastText**		**GloVe**		**Word2Vec**		**USE**	
Classifier	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1
J-48	91.79	87.09	93.44	89.59	**93.65**	89.85	93.36	**89.91**	92.53	87.91
J-Rip	85.09	75.21	89.78	84.43	89.86	**85.53**	**89.91**	84.62	88.34	81.77
Naive Bayes	57.5	52.25	**69.25**	**66.96**	66.5	64.48	65.71	63.93	51.55	47.02
Random Forest	97.8	96.83	**98.13**	**97.24**	98.05	95.93	98.12	96.08	97.9	96.09
Random Tree	97.25	95.84	97.2	95.74	96.99	95.64	**97.41**	96.07	97.01	**96.61**
SVM	79.43	51.65	**90.27**	81.32	89.49	79.17	89.83	79.91	89.9	**84.42**
KNN (k=1)	97.13	95.54	**97.54**	**96.3**	97.22	96.26	97.43	96.02	97.43	96.19
KNN (k=5)	81.67	64.27	**84.57**	69.12	84.3	68.44	84	68	84.05	**69.69**

TABLE 11. Performances of DL models with Word Embeddings.

	Domain-Train		**fastText**		**GloVe**		**Word2Vec**		**USE**	
Morbidity Class	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1
Asthma	86.75	57.51	95.68	90.56	**96.63**	**93.45**	92.85	84.57	87.81	58.77
CAD	60.5	54.4	87.68	87.32	**91.07**	**90.5**	74.08	73.05	80.88	82.73
CHF	61.17	56.74	87.6	87.52	**93.91**	**93.85**	89.36	89.25	81.51	78.35
Depression	80.57	60.54	94.62	91.75	**96.58**	**94.83**	91.73	87.6	88.13	71.26
Diabetes	73.26	56.9	92.15	91.41	**94.8**	**91.73**	91.32	89.72	88.93	85.1
Gallstones	83.53	54	88.32	83.56	**92.46**	**85.96**	89.74	75.65	89.42	69.86
Gerd	78.43	54.49	83.1	74.77	**89.04**	**82.17**	75.74	63.59	86.95	68.82
Gout	88.25	53.96	96.21	90.67	**96.81**	**91.43**	87.96	69.63	91.33	64.02
Hypercholesterolemia	67.56	66.05	88.83	88.12	**91.08**	**90.56**	88.45	88.02	82.42	83.84
Hypertension	79.94	57.29	**97.24**	95.16	97.23	**95.92**	89.91	82.8	89.28	93.54
Hypertriglyceridemia	94.35	61.77	93.36	86.47	98.87	93.92	97.13	72.57	**98.56**	**87.13**
OA	76.97	57.75	88.85	83	**93.26**	**89.33**	82.22	70.02	86.4	67.04
Obesity	55.05	48.72	84.3	83.31	**85.8**	**85.49**	67.52	64.19	64.08	62.36
OSA	86.92	55.38	93.25	84.97	**97.58**	**94.77**	91.85	83.92	92.44	71.69
PVD	86.03	55.86	95.41	90.04	**99.14**	**98.25**	92.53	81.55	91.23	67.87
Venous Insufficiency	92.3	47.95	**97.9**	**86.43**	97.66	84.23	94.04	66.16	87.89	32.8
Average	78.22	56.2	91.53	87.19	**94.3**	**91.21**	87.28	77.64	86.46	71.81

fastText, GloVe, and USE models. The results of the experiments are summarized in Tables 9, 10, 11. In particular, Tables 9 and 10 present the results of CML classifiers using word embeddings representation with the input data without the removal of stopwords (raw) and with the input data not containing the stopwords (pre-processed), respectively. The best performances of CML classifiers with word embeddings representations extracted from Tables 9 and 10 are shown in Figure 7. Moreover, for ease of understanding, Figure 8 represents the performance of the CML and DL classifiers with bag-of-words coupled with feature selection algorithms. Figure 9 shows the CML and DL classifiers' plot with word embedding representation. The winning configurations are highlighted for each kind of used representation.

The key observations from the performed experiments are listed below:

- The CML classifiers have performed only slightly better (less than 1% of difference) with embeddings when the input data do not contain the stopwords. The case when the input data contain the stop words has lower performances, where the domain-trained and USE embeddings are the exceptions.
- Given the small size of the used dataset and the minimal difference between the two kinds of CML models (with and without stopwords), we cannot draw any conclusions related to improvements or not derived from the presence of stopwords in the dataset. However, we believe that, given the technical terminology used within the clinical notes, stopwords should not play an important role when preprocessing the dataset. A more detailed analysis of them is out of the scope of the paper and will be investigated in a future direction.

- In the case of DL models, the use of word embeddings has further improved their performance with respect to the bag-of-words representation coupled with feature selection algorithms. The best performance of the DL model is observed when GloVe word embeddings are employed with 94.3 average micro F-1 scores (Table 11) against the average micro F-1 score of 89.63 when used with bag-of-words representation (Table 8). Besides, the former corresponds to the best performance of DL models for all sets of experiments.
- Generally, it is expected that the domain-specific word embeddings will perform better (due to the absence of out-of-vocabulary words) than pre-trained word embeddings, but it does not happen if the training data is small. The small amount of data in fact, jeopardizes the chances of learning the subtle peculiarities of the domain and will lead to the high variance estimation of the model's performance. For such a reason, the performances of the DL models with domain-trained embeddings are worse than those of the other four pre-trained embeddings. In contrast to the DL models, the performance of CML models using domain-specific word embeddings is only slightly affected by the small size of the dataset.
- Regarding the computational time, the CML models have again turned out to be computationally fast and less resource exhaustive as compared to the DL models. The training time of the CML models ranges between 80 and 600 seconds. The reason for the reduced training time with respect to the CML models employing bag-of-words is the lower dimension of the embedding vectors (typically 300-dimensional for all types of word-embeddings except USE, which has 512).
- Different from the CML classifiers, the training time of the DL classifiers has increased up to 40 hours. The reason for this higher computational cost lies within the employment of new layers of deep neural networks.

The comparison of the training time between the CML and DL models are presented in Figures 5 and 6. Finally, Table 11 presents the results of DL models with the word embeddings representation.

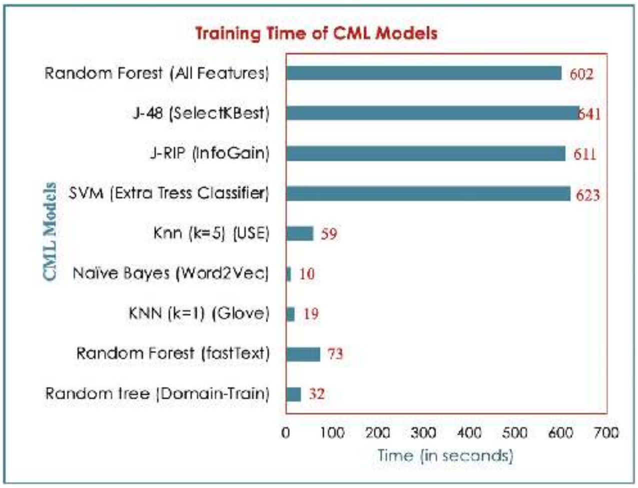


FIGURE 5. The training time of CML models with different representations.

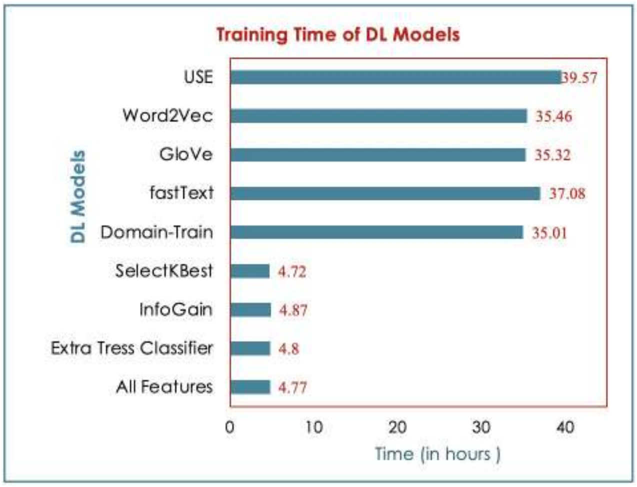
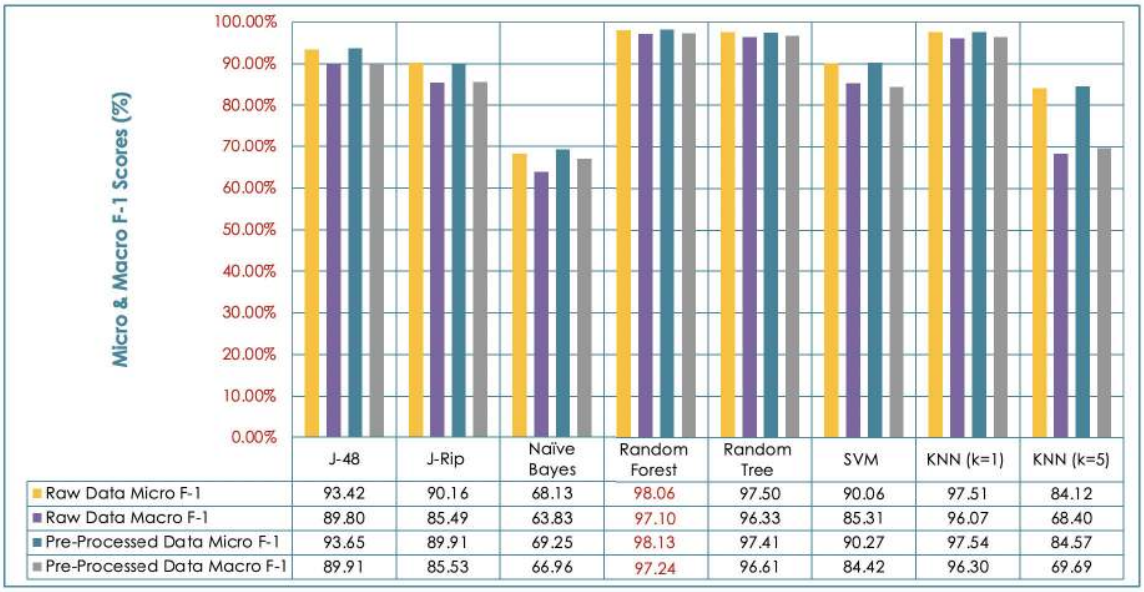


FIGURE 6. The training time of DL models with different representations.

C. EXPERIMENTAL RESULTS WITH ENSEMBLE APPROACH

In this final group of experiments, we will discuss the ensemble approach we have employed. Ensemble learning works by first training each single machine learning model and then combining the predictions of them. The rationale behind ensemble learning is to take the best from a given set of algorithms by combining their outputs. Given the large number of classifiers employed in our study, it was not feasible to experiment with all possible combinations of machine learning algorithms. For this reason, we selected the most effective DL and CML algorithms for experimenting with the ensemble approach. We performed our experiments with the BiLSTM based DL model and 8 CML algorithms. We used the four bag-of-words models with feature selection and the five types of word embeddings for each of them. Hence, we considered $9 \times 9 = 81$ classification models totally. Considering the formula $2^a - (a+1)$, with $a \geq 2$ equal to the number of models, for calculating the total number of possible ensembles constituted, in our case we have $a = 81$. This would account for a total of $(2^{81} - 82)$ possible combinations. It would be unfeasible to compute all the possible ensembles resulting from the formula above. Therefore, we have limited the number of models for generating the configurations of ensembles. We hypothesized that combining CML and DL classifiers in the same ensemble configuration would increase the model's stability without decreasing accuracy. Hence, we included the 6 top-performing CML models and the 5 top-performing DL models in our pool of classifiers to be included in the ensemble configurations. We used r out of 11 classifiers for each configuration, with r being an odd integer number between 3 and 11. Using an odd number of classifiers, we could straightforwardly apply the majority voting technique. The choice of using 11 classifiers corresponded to 1013 different



	J-48	J-Rip	Naïve Bayes	Random Forest	Random Tree	SVM	KNN (k=1)	KNN (k=5)
■ Raw Data Micro F-1	93.42	90.16	68.13	98.06	97.50	90.06	97.51	84.12
■ Raw Data Macro F-1	89.80	85.49	63.83	97.10	96.33	85.31	96.07	68.40
■ Pre-Processed Data Micro F-1	93.65	89.91	69.25	98.13	97.41	90.27	97.54	84.57
■ Pre-Processed Data Macro F-1	89.91	85.53	66.96	97.24	96.61	84.42	96.30	69.69

FIGURE 7. Best performances of CML classifiers using embeddings with and without stopwords taken from Tables 9 and 10.

ensemble configurations, which we believe is a reasonable number for our experiment. The classifiers selected for the ensemble configurations are listed below:

- Random Forest classifier used with SelectKBest feature selection algorithm.
- SVM classifier used with ExtraTreesClassifier feature selection algorithm.
- kNN classifier (where $k = 1$) used with ExtraTreesClassifier feature selection algorithm.
- kNN classifier (where $k = 1$) used with fastText word embeddings representation.
- Random Forest classifier used with USE word embeddings representation.
- Random Forest classifier used with fastText word embeddings representation.
- DL model used with USE word embeddings representation.
- DL model used with GloVe word embeddings representation.
- DL model used with fastText word embeddings representation.
- DL model used with InfoGain feature selection algorithm.
- DL model used with ExtraTreesClassifier feature selection algorithm.

We computed the performances of all the above-mentioned 1013 ensemble combinations, and the results of the six best performing combinations among them are discussed in the following. Out of the top six ensemble models, ensembles 1, 3, and 5 consist of five classification models, while 3 classification models constitute ensembles 2, 4, and 6. The results of the ensembles are summarized in Table 12.

The structure of the top six ensemble combinations is listed below:

- **Ensemble-1**. The number of constituting classifiers for Ensemble-1 is 5, which are: DL models with (fastText and GloVe) word embeddings, SVM with ExtraTreesClassifier algorithm, Random Forest with SelectKBest algorithm, kNN($k = 1$) with fastText word embeddings.
- **Ensemble-2**. The number of constituting classifiers for Ensemble-2 is 3, which are: DL model with GloVe word embeddings, SVM with ExtraTreesClassifier, kNN($k = 1$) with fastText word embeddings.
- **Ensemble-3**. The number of constituting classifiers for Ensemble-3 is 5, which are: DL models with (fastText and GloVe) word embeddings, SVM with ExtraTreesClassifier algorithm, kNN($k = 1$)ich are: DL model with GloV with ExtraTreesClassifier algorithm, kNN($k = 1$) with fastText word embeddings.
- **Ensemble-4**. The number of constituting classifiers for Ensemble-4 is 3, which are: DL model with fastText word embeddings, SVM with ExtraTreesClassifier, kNN($k = 1$) with fastText word embeddings.
- **Ensemble-5**. The number of constituting classifiers for Ensemble-5 is 5, which are: DL models with (fastText and GloVe) word embeddings, SVM with ExtraTreesClassifier algorithm, Random Forest with fastText word embeddings, kNN($k = 1$) with fastText word embeddings.
- **Ensemble-6**. The number of constituting classifiers for Ensemble-6 is 3, which are: DL model with GloVe word embeddings, Random Forest with SelectKBest algorithm, kNN($k = 1$) with fastText word embeddings.

To get the final predictions of the ensembles, we have used the majority voting technique, generally used for these kinds of tasks [65]. In this technique, multiple models are used to make predictions for each clinical record, and predictions by each

TABLE 12. Performances of ensemble approaches.

Morbidity	Ensemble-1		Ensemble-2		Ensemble-3		Ensemble-4		Ensemble-5		Ensemble-6	
Class	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1
Asthma	**99.37**	**98.75**	99.05	98.14	**99.37**	**98.75**	99.16	98.34	**99.37**	**98.75**	**99.37**	**98.75**
CAD	99.58	99.55	99.58	99.56	99.47	99.44	**99.68**	**99.67**	99.58	99.56	99.58	99.55
CHF	**99.78**	**99.78**	99.67	99.67	**99.78**	**99.78**	99.67	99.67	**99.78**	**99.78**	99.34	99.34
Depression	98.86	98.37	98.97	98.52	98.86	98.37	**99.07**	**98.67**	98.86	98.37	98.76	98.22
Diabetes	98.27	97.90	98.27	97.91	98.16	97.77	98.16	97.78	98.16	97.77	**98.37**	**98.02**
Gallstones	99.70	99.47	**99.80**	**99.65**	**99.80**	**99.65**	99.70	99.48	99.70	99.47	99.60	99.30
Gerd	98.95	98.49	**99.18**	**98.83**	99.07	98.66	99.07	98.66	98.95	98.49	98.95	98.49
Gout	99.70	99.31	99.70	99.31	99.70	99.31	**99.80**	**99.54**	99.70	99.31	99.70	99.31
Hypercholesterolemia	**98.52**	**98.49**	98.06	98.02	98.40	98.37	98.29	98.26	98.29	98.25	**98.52**	**98.49**
Hypertension	**99.68**	**99.47**	99.58	99.30	99.47	99.12	99.47	99.12	99.47	99.12	99.58	99.30
Hypertriglyceridemia	99.49	97.44	99.49	97.44	99.49	97.44	**99.59**	**97.97**	99.49	97.44	99.39	96.90
OA	98.93	98.39	98.93	98.40	**99.04**	**98.56**	**99.04**	**98.56**	98.93	98.39	98.72	98.06
Obesity	99.03	99.02	99.25	99.24	99.03	99.02	98.71	98.69	98.92	98.91	**99.46**	**99.46**
OSA	99.50	98.94	**99.60**	**99.16**	99.50	98.94	99.40	98.74	99.50	98.94	99.40	98.73
PVD	99.04	98.06	**99.15**	**98.28**	99.04	98.06	99.04	98.06	99.04	98.06	98.93	97.84
Venous Insufficiency	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
Average	**99.27**	**98.84**	**99.27**	**98.84**	99.26	98.83	99.24	98.83	99.23	98.79	99.23	98.73

TABLE 13. Average, best micro F-1 score and standard deviation of CML and DL classifiers and ensembles. results are averaged over all the morbidity classes.

CML Classifiers Used With	Average Micro F-1 Score	Best Micro F-1 Score	Standard Deviation
All Features	**94.66**	98.45	5.71
Extratrees Algorithm	94.33	**99.26**	7.04
InfoGain Algorithm	94.48	98.59	5.68
SelectKBest Algorithm	94.58	98.47	**5.60**
Domain-Train Embeddings	85.95	97.25	14.49
Fasttext Embeddings	90.02	97.54	10.27
Glove Embeddings	89.53	98.05	11.15
Word2vec Embeddings	89.47	98.12	11.50
USE Embeddings	87.33	97.43	16.34
DL Approach Used with	**Average Micro F-1 Score**	**Best Micro F-1 Score**	**Standard Deviation**
Bag-of-Words Representations	81.37	89.63	6.31
Word Embeddings Representations	**87.58**	**94.3**	**6.11**
Ensemble Approach Using	**Average Micro F-1 Score**	**Best Micro F-1 Score**	**Standard Deviation**
3-Models	96.96	**99.27**	2.35
5-Models	97.97	**99.27**	0.96
7-Models	98.34	99.12	0.47
9-Models	**98.51**	98.93	**0.27**
11-Model (just 1)	97.79	97.79	n.a.

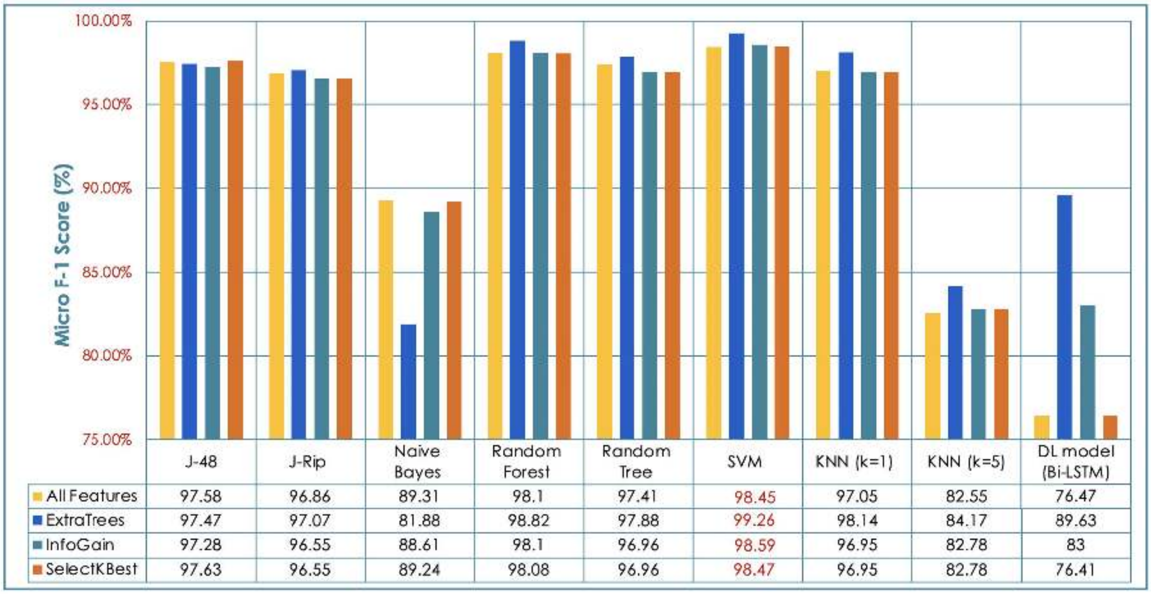
model are considered as a "vote." For instance, for a document (a clinical record), if three classifiers have predicted the class of a sample as 1, 0, and 1, then the final predicted label will be 1, as it secures more than half the votes. For ease of understanding, we have summarized our experimental results in Table 13. The first section of Table 13 presents the average performances of the eight CML algorithms with each of the four bag-of-words models coupled with feature selection algorithms and the five-word embeddings. The second section of Table 13 presents the average performances of the nine DL models used with each of the two representations, i.e., the four bag-of-words models representations and the five-word embeddings representations. Lastly, the third section shows the average performances of all the ensemble models we have tested with 3, 5, 7, and 9 constituents.

The comparison of the aforementioned performances has been done in terms of the average micro F-1 score, best micro F-1 score, and standard deviation. Note that values in each row of the table are averaged over all the morbidity classes and settings within the underlying model. The best performer among them gives an F-1 score of 99.27, with an average of 97.97 and a standard deviation of 0.96. From the results, we can observe that the CML and DL classifiers' performances are lower than the presented ensembles.

VII. DISCUSSION

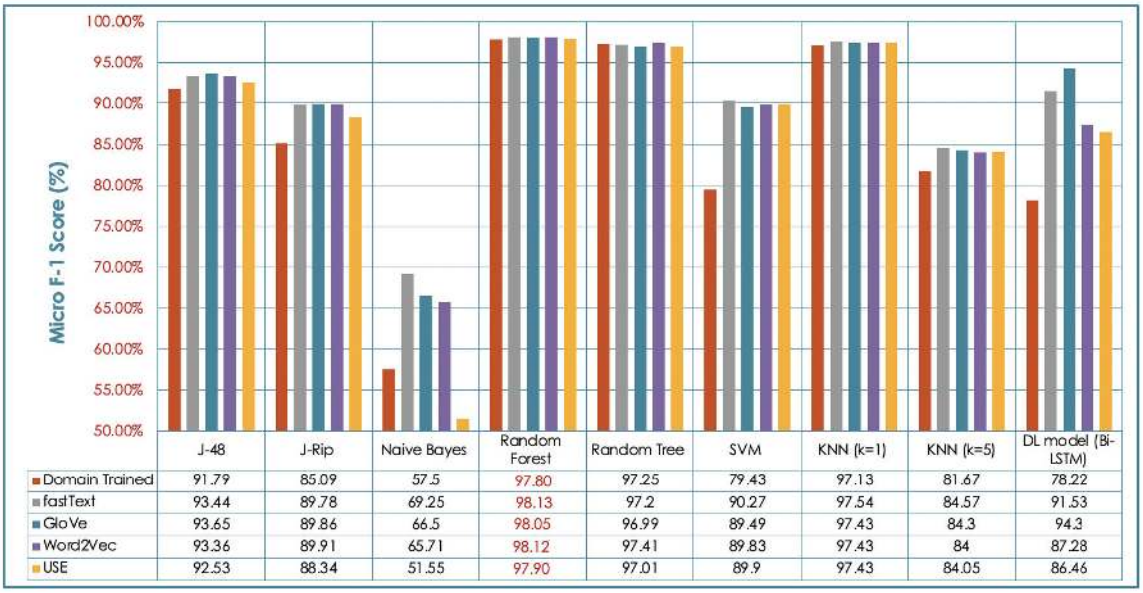
We have analyzed the performance variations of CML and DL classifiers with the different feature vector representations.

Firstly, we will discuss the performances of CML classifiers with word embeddings with or without stopwords. The results indicate that the performances of CML classifiers are slightly better when data do not include the stopwords in general when used with different embeddings, with domain-trained and USE embeddings being the exception. Unlike the other word embeddings approaches that take a word as an input to generate the feature vectors, the input to USE is a sentence. Therefore, the embeddings produced by USE for the sentence capture the context of the sentence



	J-48	J-Rip	Naive Bayes	Random Forest	Random Tree	SVM	KNN (k=1)	KNN (k=5)	DL model (Bi-LSTM)
■ All Features	97.58	96.86	89.31	98.1	97.41	98.45	97.05	82.55	76.47
■ ExtraTrees	97.47	97.07	81.88	98.82	97.88	99.26	98.14	84.17	89.63
■ InfoGain	97.28	96.55	88.61	98.1	96.96	98.59	96.95	82.78	83
■ SelectKBest	97.63	96.55	89.24	98.08	96.96	98.47	96.95	82.78	76.41

FIGURE 8. Experimental results of CML and DL models with and without the employment of feature selection algorithms.



	J-48	J-Rip	Naive Bayes	Random Forest	Random Tree	SVM	KNN (k=1)	KNN (k=5)	DL model (Bi-LSTM)
■ Domain Trained	91.79	85.09	57.5	97.80	97.25	79.43	97.13	81.67	78.22
■ fastText	93.44	89.78	69.25	98.13	97.2	90.27	97.54	84.57	91.53
■ GloVe	93.65	89.86	66.5	98.05	96.99	89.49	97.43	84.3	94.3
■ Word2Vec	93.36	89.91	65.71	98.12	97.41	89.83	97.43	84	87.28
■ USE	92.53	88.34	51.55	97.90	97.01	89.9	97.43	84.05	86.46

FIGURE 9. Experimental results of CML and DL models with word embeddings.

and the mutual relatedness of words within it. The removal of stopwords can change the meaning of the sentence, negatively impacting the predictions.

In the case of CML classifiers used with bag-of-words representation, the performances of CML classifiers have improved with the ExtraTreesClassifier feature selection algorithm, i.e., SVM with the F-1 score of 99.26, which is the best performance for all the performed experiments. Overall, the CML classifiers have performed better with the feature selection algorithms.

Furthermore, in the case of DL approaches, the used feature selection algorithms have substantially improved the model's performance. The F-1 score of 76.25 with All Features has increased to 89.63 when ExtraTreesClassifier is adopted. In the case of the DL approaches used with different word embeddings, GloVe has achieved the best results. In the context of training time, the CML models have proven to be computationally lighter and faster to train. Conversely, the DL models have a long training time, which increases while switching from experiments with bag-of-words to word embedding representations.

Finally, the integration of CML and DL approaches by employing the ensemble technique to produce ensemble models has improved the single best classification model's performances. While the best performances of the DL models were achieved with GloVe word embeddings obtaining a micro F-1 score of 94.3, the top 989 out of 1013 ensembles got a higher score than it. Although the best ensemble score of 99.27 is only slightly better than the best performance of a single CML model, 99.26, the efficacy of ensemble models can be appreciated by their high average and low standard deviation values. The average micro F1 scores of ensembles made of 3, 5, 7, and 9 classification models are greater than the average of each single representation technique used for experiments. In addition, while the CML classifiers suffer from a high standard deviation value, the ensembles are much more stable with a standard deviation, which decreases from 2.35 when using 3 classifiers to 0.27 when using 9 classifiers. Despite being computationally intensive, the ensemble method proved to be a viable technique. Indeed, for a highly imbalanced and small dataset like the one we used, the prediction stability of the model is quintessential. In general, for the minority class, the classification models tend to achieve lower precision or recall scores. Using the ensemble approach, we can not only deal better with the prediction of the minority class but also reduce the variance of predictions and thus the generalization error.

VIII. CONCLUSION

We have used CML, and DL approaches to tackle the multi-classification of clinical records by employing bag-of-words using TF-IDF and word embeddings feature representation methods. We have conducted experiments to observe how each method can contribute to morbidity identification, leveraging various feature selection and pre-processing techniques. The results show that the size of the dataset is critical for DL models' performances when the training data is unbalanced. For our dataset, CML classifiers have performed better than the DL models when used with the word embeddings representations. For the DL approaches, word embeddings representations have performed significantly better than the TF-IDF representation of All Features and feature selection algorithms. Finally, we have generated ensemble models by coupling DL models and CML classifiers used with different representations and adopting the majority voting strategy. The ensembles have proven to be useful for the small dataset in mitigating the biased behavior of a single classifier model as well as in improving the single best model's performance prediction stability. Although word embeddings are powerful vector representation techniques, the performances of DL models greatly depend upon the size of the data. A small dataset prevents the BiLSTM layers of the models from learning the fine peculiarities of input data, which are important elements in "handling long-term dependencies." A large dataset can be a game-changer in enhancing the DL model's performance. In the context of future work, techniques like data augmentation and state-of-the-art word embeddings representations exploiting transformer architecture such as BERT, ELMO, XLNet, etc., could be employed to deal with the constraints of small datasets in order to improve the performances of DL models and the overall ensemble. Moreover, a detailed analysis of the benefits of removing or not the stopwords from the clinical notes will be carried out to understand when they are useful or not in the underlying domain. Last but not least, we would like to apply the proposed approach to solve other multi-label classification problems present in domains different from health and analyze and compare the results against those obtained by the study we have done in this paper.

ABBREVIATIONS

Classical Machine Learning-(CML), Deep Learning-(DL), Natural Language Processing-(NLP), Term Frequency-Inverse Document Frequency-(TF-IDF), Long-Short Term Memory-(LSTM), Convolutional Neural Network-(CNN), Universal Sentence Encoder-(USE), Continuous Bag of Words-(CBOW), Support Vector Machine-(SVM), k-Nearest Neighbours-(kNN), Bidirectional Encoder Representations from Transformers-(BERT).

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X GPU used for this research. Rim Helaoui was not involved in the decisions related to the processing of the raw data, nor in the processing activities of the raw data. Her role in the project is not a data controller, nor a data processor, but a knowledge contributor. Philips was not involved in the decisions related to the processing of the raw data, nor in the processing activities of the raw data. Philips' role was merely knowledge contributor.

REFERENCES

[1] A. A. Uijen and E. H. van de Lisdonk, "Multimorbidity in primary care: Prevalence and trend over the last 20 years," *Eur. J. Gen. Pract.*, vol. 14, no. sup1, pp. 28–32, Jan. 2008.

[2] *World Mortality Report*. Department of Economic and Social Affairs, United Nations Publications, New York, NY, USA, 2013.

[3] R. W. Blum, F. I. P. M. Bastos, C. Kabiru, and L. C. Le, "Adolescent health in the 21st century," *Lancet*, vol. 379, no. 9826, pp. 1567–1568, Apr. 2012.

[4] S. M. Sawyer, S. Drew, M. S. Yeo, and M. T. Britto, "Adolescents with a chronic condition: Challenges living, challenges treating," *Lancet*, vol. 369, no. 9571, pp. 1481–1489, Apr. 2007.

[5] S. Mercer, J. Furler, K. Moffat, D. Fischbacher-Smith, and L. Sanci, *Multimorbidity: Technical Series on Safer Primary Care*. Geneva, Switzerland: World Health Organization, 2016.

[6] K. Barnett, S. W. Mercer, M. Norbury, G. Watt, S. Wyke, and B. Guthrie, "Epidemiology of multimorbidity and implications for health care, research, and medical education: A cross-sectional study," *Lancet*, vol. 380, no. 9836, pp. 37–43, Jul. 2012.

[7] R. Wyber, S. Vaillancourt, W. Perry, P. Mannava, T. Folaranmi, and L. A. Celi, "Big data in global health: Improving health in low-and middle-income countries," *Bulletin World Health Org.*, vol. 93, pp. 203–208, Jan. 2015.

[8] S. Bromuri, D. Zufferey, J. Hennebert, and M. Schumacher, "Multi-label classification of chronically ill patients with bag of words and supervised dimensionality reduction algorithms," *J. Biomed. Informat.*, vol. 51, pp. 165–175, Oct. 2014.

[9] J. Gu, W. Feng, J. Zeng, H. Mamitsuka, and S. Zhu, "Efficient semisupervised MEDLINE document clustering with MeSH-semantic and global-content constraints," *IEEE Trans. Cybern.*, vol. 43, no. 4, pp. 1265–1276, Aug. 2013.

[10] D. Dessì, D. R. Recupero, G. Fenu, and S. Consoli, "A recommender system of medical reports leveraging cognitive computing and frame semantics," in *Machine Learning Paradigms*. Cham, Switzerland: Springer, 2019, pp. 7–30.

[11] S. Consoli, D. R. Recupero, and M. Petkovic, Eds., *Data Science for Healthcare—Methodologies and Applications*. Cham, Switzerland: Springer, 2019.

[12] H.-C. Thorsen-Meyer, A. B. Nielsen, A. P. Nielsen, B. S. Kaas-Hansen, P. Toft, J. Schierbeck, T. Strøm, P. J. Chmura, M. Heimann, L. Dybdahl, L. Spangsege, P. Hulsen, K. Belling, S. Brunak, and A. Perner, ''Dynamic and explainable machine learning prediction of mortality in patients in the intensive care unit: A retrospective study of high-frequency data in electronic patient records,'' *Lancet Digit. Health*, vol. 2, no. 4, pp. e179–e191, Apr. 2020.

[13] E. Gultepe, J. P. Green, H. Nguyen, J. Adams, T. Albertson, and I. Tagkopoulos, ''From vital signs to clinical outcomes for patients with sepsis: A machine learning basis for a clinical decision support system,'' *J. Amer. Med. Inform. Assoc.*, vol. 21, no. 2, pp. 315–325, Mar. 2014.

[14] C.-S. Wu, A. R. Luedtke, E. Sadikova, H.-J. Tsai, S.-C. Liao, C.-C. Liu, S. S.-F. Gau, T. J. VanderWeele, and R. C. Kessler, ''Development and validation of a machine learning individualized treatment rule in first-episode schizophrenia,'' *JAMA Netw. Open*, vol. 3, no. 2, Feb. 2020, Art. no. e1921660.

[15] V. Kumar, B. K. Mishra, M. Mazzara, D. N. Thanh, and A. Verma, ''Prediction of malignant and benign breast cancer: A data mining approach in healthcare applications,'' in *Advances in Data Science and Management*. Singapore: Springer, 2020, pp. 435–442.

[16] T. Panch, P. Szolovits, and R. Atun, ''Artificial intelligence, machine learning and health systems,'' *J. Global Health*, vol. 8, no. 2, Dec. 2018, Art. no. 020303.

[17] L. Yao, C. Mao, and Y. Luo, ''Clinical text classification with rule-based features and knowledge-guided convolutional neural networks,'' *BMC Med. Informat. Decis. Making*, vol. 19, no. S3, p. 71, Apr. 2019.

[18] K. Rajput, G. Chetty, and R. Davey, ''Obesity and co-morbidity detection in clinical text using deep learning and machine learning techniques,'' in *Proc. 5th Asia–Pacific World Congr. Comput. Sci. Eng. (APWC CSE)*, Dec. 2018, pp. 51–56.

[19] Z. Liu, M. Yang, X. Wang, Q. Chen, B. Tang, Z. Wang, and H. Xu, ''Entity recognition from clinical texts via recurrent neural network,'' *BMC Med. Informat. Decis. Making*, vol. 17, no. S2, p. 67, Jul. 2017.

[20] S. A. Waheeb, N. Ahmed Khan, B. Chen, and X. Shang, ''Machine learning based sentiment text classification for evaluating treatment quality of discharge summary,'' *Information*, vol. 11, no. 5, p. 281, May 2020.

[21] M. Oleynik, A. Kugic, Z. Kasá , and M. Kreuzthaler, ''Evaluating shallow and deep learning strategies for the 2018 n2c2 shared task on clinical text classification,'' *J. Amer. Med. Inform. Assoc.*, vol. 26, no. 11, pp. 1247–1254, Nov. 2019.

[22] Q. Wei, Z. Ji, Y. Si, J. Du, J. Wang, F. Tiryaki, S. Wu, C. Tao, K. Roberts, and H. Xu, ''Relation extraction from clinical narratives using pre-trained language models,'' in *Proc. AMIA Annu. Symp.*, Bethesda, MD, USA: American Medical Informatics Association, 2019, p. 1236.

[23] R. Leaman, R. Khare, and Z. Lu, ''Challenges in clinical natural language processing for automated disorder normalization,'' *J. Biomed. Informat.*, vol. 57, pp. 28–37, Oct. 2015.

[24] F. K. Khattak, S. Jeblee, C. Pou-Prom, M. Abdalla, C. Meaney, and F. Rudzicz, ''A survey of word embeddings for clinical text,'' *J. Biomed. Inform., X*, vol. 4, Dec. 2019, Art. no. 100057.

[25] H. Xu, W. Wang, W. Liu, and L. Carin, ''Distilled wasserstein learning for word embedding and topic modeling,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1716–1725.

[26] D. Dessi, R. Helaoui, V. Kumar, D. R. Recupero, and D. Riboni, ''Tf-IDF vs word embeddings for morbidity identification in clinical notes: An initial study,'' *CEUR Proceddings*, vol. 2796, pp. 1–12, Mar. 2020.

[27] J. Tang, S. Alelyani, and H. Liu, ''Feature selection for classification: A review,'' in *Data Classification, Algorithms Applications*. Boca Raton, FL, USA: CRC Press, 2014, p. 37.

[28] I. Guyon and A. Elisseeff, ''An introduction to variable and feature selection,'' *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.

[29] D. Jain and V. Singh, ''Feature selection and classification systems for chronic disease prediction: A review,'' *Egyptian Informat. J.*, vol. 19, no. 3, pp. 179–189, Nov. 2018.

[30] J. Li and H. Liu, ''Challenges of feature selection for big data analytics,'' *IEEE Intell. Syst.*, vol. 32, no. 2, pp. 9–15, Mar. 2017.

[31] S. Huda, J. Yearwood, H. F. Jelinek, M. M. Hassan, G. Fortino, and M. Buckland, ''A hybrid feature selection with ensemble classification for imbalanced healthcare data: A case study for brain tumor diagnosis,'' *IEEE Access*, vol. 4, pp. 9145–9154, 2016.

[32] C. A. Flores, R. L. Figueroa, and J. E. Pezoa, ''FREGEX: A feature extraction method for biomedical text classification using regular expressions,'' in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2019, pp. 6085–6088.

[33] T. Vivekanandan and N. C. S. N. Iyengar, ''Optimal feature selection using a modified differential evolution algorithm and its effectiveness for prediction of heart disease,'' *Comput. Biol. Med.*, vol. 90, pp. 125–136, Nov. 2017.

[34] O. Dekel and O. Shamir, ''Multiclass-multilabel classification with more classes than examples,'' in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 137–144.

[35] J. Read, B. Pfahringer, G. Holmes, and E. Frank, ''Classifier chains for multi-label classification,'' *Mach. Learn.*, vol. 85, no. 3, p. 333, Dec. 2011.

[36] V. Kumar, A. Verma, N. Mittal, and S. V. Gromov, ''Anatomy of preprocessing of big data for monolingual corpora paraphrase extraction: Source language sentence,'' in *Emerging Technologies in Data Mining and Information Security*, vol. 3. Singapore: Springer, 2019, p. 495.

[37] D. Dessi, D. R. Recupero, G. Fenu, and S. Consoli, ''Exploiting cognitive computing and frame semantic features for biomedical document clustering,'' in *Proc. SeWeBMeDA@ ESWC*, 2017, pp. 20–34.

[38] M. Atzeni and D. R. Recupero, ''Multi-domain sentiment analysis with mimicked and polarized word embeddings for human-robot interaction,'' *Future Gener. Comput. Syst.*, vol. 110, pp. 984–999, Sep. 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X19309719

[39] A. Dridi and D. R. Recupero, ''Leveraging semantics for sentiment polarity detection in social media,'' *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 8, pp. 2045–2055, Aug. 2019, doi: 10.1007/s13042-017-0727-z.

[40] M. Hasan, A. Kotov, A. I. Carcone, M. Dong, S. Naar, and K. B. Hartlieb, ''A study of the effectiveness of machine learning methods for classification of clinical interview fragments into a large number of categories,'' *J. Biomed. Informat.*, vol. 62, pp. 21–31, Aug. 2016.

[41] Y. HaCohen-Kerner, D. Miller, and Y. Yigal, ''The influence of preprocessing on text classification using a bag-of-words representation,'' *PLoS ONE*, vol. 15, no. 5, May 2020, Art. no. e0232525.

[42] Y. HaCohen-Kerner, A. Rosenfeld, A. Sabag, and M. Tzidkani, ''Topic-based classification through unigram unmasking,'' *Procedia Comput. Sci.*, vol. 126, pp. 69–76, Jan. 2018.

[43] W. Zhang, T. Yoshida, and X. Tang, ''TFIDF, LSI and multi-word in information retrieval and text categorization,'' in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2008, pp. 108–113.

[44] M. Naili, A. H. Chaibi, and H. H. Ben Ghezala, ''Comparative study of word embedding methods in topic segmentation,'' *Procedia Comput. Sci.*, vol. 112, pp. 340–349, Jan. 2017.

[45] M. Kholghi, L. De Vine, L. Sitbon, G. Zuccon, and A. Nguyen, ''The benefits of word embeddings features for active learning in clinical information extraction,'' in *Proc. Australas. Lang. Technol. Assoc. Workshop*, Melbourne, VIC, Australia, Dec. 2016, pp. 25–34. [Online]. Available: https://www.aclweb.org/anthology/U16-1003

[46] B. Chandra and P. Paul Varghese, ''Fuzzifying gini index based decision trees,'' *Expert Syst. Appl.*, vol. 36, no. 4, pp. 8549–8559, May 2009.

[47] S. Gnanambal, M. Thangaraj, V. Meenatchi, and V. Gayathri, ''Classification algorithms with attribute selection: An evaluation study using weka,'' *Int. J. Adv. Netw. Appl.*, vol. 9, no. 6, pp. 3640–3644, 2018.

[48] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, ''Distributed representations of words and phrases and their compositionality,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[49] J. Pennington, R. Socher, and C. Manning, ''Glove: Global vectors for word representation,'' in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[50] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, ''Bag of tricks for efficient text classification,'' 2016, *arXiv:1607.01759*. [Online]. Available: http://arxiv.org/abs/1607.01759

[51] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, ''Enriching word vectors with subword information,'' *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2017.

[52] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, ''Advances in pre-training distributed word representations,'' in *Proc. Int. Conf. Lang. Resour. Eval.*, 2018, pp. 1–4.

[53] D. Cer, Y. Yang, S.-Y. Kong, N. Hua, N. Limtiaco, R. St. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, B. Strope, and R. Kurzweil, ''Universal sentence encoder for english,'' in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*, 2018, pp. 169–174.

[54] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, ''Supervised learning of universal sentence representations from natural language inference data,'' 2017, *arXiv:1705.02364*. [Online]. Available: http://arxiv.org/abs/1705.02364

[55] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
[56] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Mach. Learn.*, vol. 6, no. 1, pp. 37–66, Jan. 1991.
[57] G. H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," in *Proc. 11th Conf. Uncertainty Artif. Intell.*, 1995, pp. 338–345.
[58] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[59] B. Pittel, "Note on the heights of random recursive trees and random m-ary search trees," *Random Struct. Algorithms*, vol. 5, no. 2, pp. 337–347, Apr. 1994.
[60] S. L. Salzberg, *C4. 5: Programs for Machine Learning by J. Ross Quinlan*. Burlington, MA, USA: Morgan Kaufmann, 1994.
[61] W. W. Cohen, "Fast effective rule induction," in *Machine Learning Proceedings*. Amsterdam, The Netherlands: Elsevier, 1995, pp. 115–123.
[62] M. Liwicki, A. Graves, S. Fernàndez, H. Bunke, and J. Schmidhuber, "A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks," in *Proc. 9th Int. Conf. Document Anal. Recognit., ICDAR*, 2007, pp. 1–5.
[63] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Austin, TX, USA: Association for Computational Linguistics, Nov. 2016, pp. 551–561. [Online]. Available: https://www.aclweb.org/anthology/D16-1053
[64] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. Int. Joint Conf. AI*, vol. 14. no. 2, Aug. 1995, pp. 1137–1145.
[65] G. Brown, "Ensemble learning," in *Encyclopedia of Machine Learning*, vol. 312. Boston, MA, USA: Springer, 2010.

VIVEK KUMAR received the M.S. degree from NUST-MiSiS, Moscow, Russia. He is currently pursuing the Ph.D. degree with the University of Cagliari, Italy. He is also a Marie Skłodowska-Curie Researcher with the University of Cagliari, Italy, and also with Philips Research, The Netherlands. He has authored several publications and is serving as a reviewer for journals and conferences of IEEE, ACM, Springer, Elsevier, MDPI, Taylor & Francis, and IGI-Global. His research interests include machine learning, deep learning, natural language processing, and sentiment analysis applied to the healthcare domain.

DIEGO REFORGIATO RECUPERO received the Ph.D. degree in computer science from the University of Naples Federico II, Italy, in 2004. From 2005 to 2008, he has been a Postdoctoral Researcher with the University of Maryland, College Park, USA. He has been an Associate Professor with the Department of Mathematics and Computer Science, University of Cagliari, Italy, since December 2016. He won different awards in his career, such as Marie Curie International Reintegration Grant, Marie Curie Innovative Training Network, Best Research Award from the University of Catania, Computer World Horizon Award, Telecom Working Capital, and Startup Weekend. He co-founded six companies within the ICT sector and is actively involved in European projects and research (with one of his companies, he won more than 30 FP7 and H2020 projects). His current research interests include sentiment analysis, semantic web, natural language processing, human–robot interaction, financial technology, and smart grid. Among others, the fields of machine learning, deep learning, and big data are key technologies employed to effectively solve several tasks. He is the author of more than 100 conference and journal papers in these research fields, with more than 1000 citations.

DANIELE RIBONI (Member, IEEE) received the Ph.D. degree in computer science from the University of Milano, in 2007. He was a Postdoctoral Fellow and an Assistant Professor with the University of Milano. He is currently an Associate Professor of Computer Science with the University of Cagliari, since 2015. His research interests include activity recognition, pervasive healthcare, knowledge management, and privacy issues in pervasive and mobile computing. He served as the TPC chair and TPC vice-chair for different conferences and workshops in the field, including IEEE PerCom and the International Conference on Intelligent Environments. His contributions appear in major conferences and journals.

RIM HELAOUI received the Diploma degree in computer science from the Karlsruhe Institute of Technology (KIT), Germany, with a focus on cognitive systems, and the Ph.D. degree in the subfield of A.I., knowledge representation and reasoning from the University of Mannheim, Germany. As a Team Member of the Data and Web Science Group, she focused on statistical relational approaches in the context of complex human activity recognition. During her Ph.D., she had the opportunity to conduct extensive teaching and supervision activities at both bachelor and master programs. She currently holds a scientist position with the Personal Health Department, Philips Research, working on computational intelligence for automated personal health services and further supervising interns there.