An Active Learning-based Medical Diagnosis System*

Catarina Pinto[0000−0002−5791−1257], Juliana Faria[0000−0003−0060−9302], and Luis Macedo[0000−0002−3144−0362]

University of Coimbra, CISUC - Centre for Informatics and Systems of the University of Coimbra, Department of Informatics Engineering, Coimbra, Portugal
fmenino@student.dei.uc.pt jcfaria@student.dei.uc.pt
macedo@dei.uc.pt

Abstract. Every year thousands of people get their diagnoses wrongly, and several patients have their health conditions aggravated due to misdiagnosis. This problem is even more challenging when the list of possible diseases is long, as in a general medicine speciality. The development of Artificial Intelligence (AI) medical diagnosis systems could prevent misdiagnosis when clinicians are in doubt. We developed an AI system to help clinicians in their daily practice. They could consult the system to get an immediate opinion and diminish waiting times in triage services since this task could be carried out with minimal human interaction. Our method relies on Machine Learning techniques, more precisely on Active Learning and Neural Networks classifiers. To train this model, we used a data set that relates symptoms to several diseases. We compared our models with other models from the literature, and our results show that it is possible to achieve even better performance with much less data, mainly because of the contribution of the Active Learning component.

Keywords: Machine Learning · Active Learning · Deep Learning · Neural Networks · Medical Diagnosis · modAL

1 Introduction

Medicine and Artificial Intelligence (AI) have long crossed paths in different medical fields. These AI technologies have been applied with considerable success in the clinical diagnosis of acute and chronic diseases and breast cancer recurrence prediction, among others [10]. Regarding their performance in medical diagnosis, there is evidence that models in the literature are as good or better than clinicians at this task [10]. The use of AI may allow the optimization of the treatment of common complex diseases, such as cardiovascular diseases. Nevertheless, patients can benefit from a more precise treatment using AI algorithms based on big data. AI may also be an improvement at the financial level.

* This work is funded by the FCT - Foundation for Science and Technology, I.P./MCTES through national funds (PIDDAC), within the scope of CISUC R&D Unit - UIDB/00326/2020 or project code UIDP/00326/2020.

By being integrated into hospital management systems, it may reduce the costs associated with logistics and may also reduce time costs [10].

A correct medical diagnosis is crucial. The news of having an illness, the emotional distress and the costs of unnecessary treatment or all the consequences of a diagnosis that wrongly concludes that the patient is disease-free have a massive impact on the patient's life [6]. Unfortunately, there are still many cases of error or long waiting times. This is a problem for the patients since waiting may aggravate their health condition. The development of medical diagnostic models could prevent misdiagnosis when the clinicians are in doubt. They could consult the model to get an immediate opinion and diminish waiting times in triage services since this task could be carried out with minimal human interaction. Since symptoms can provide credible information for disease diagnosis, a symptom-based diagnostic model may be beneficial in achieving the aforementioned goals. Also, the diagnosis decision may become less subjective with the use of an algorithm.

Many of those AI systems rely on Machine Learning (ML) techniques. Such systems are trained with data sets that cross features with diseases to classify diseases correctly. However, these data sets are often difficult to obtain or too short to train ML algorithms effectively, resulting in imperfect models ([9]). Active Learning (AL) ([9, 8, 7]) is one of the most selected ML techniques to deal with the problem of scarcity of data. It allows the machine to choose for labelling the most informative instances among many unlabelled samples, reducing to a minimum the time spent by experts in the construction of the data set.

This study aims to help clinicians in their daily practice, especially those dealing with many diseases, as happens in general medicine specialities. We built a medical diagnosis model using AL techniques ([9]), more precisely the modAL framework ([2]), and combined it with Neural Networks (NN) as a classifier.

The remainder of this paper is structured as Materials, Methods, Results, Discussion, and Conclusion. In the next section, we describe the related work. In Section 3, we describe the materials used, while in Section 4 we present the strategy to build the model we propose. We show the results obtained with our model in Section 5, and, in Section 6, we discuss them. Finally, in Section 7, we summarize the work done and make conclusions.

2 Related work

Shen et al. [10] reviewed papers from the medical context between 2000 and 2019 that compare human clinical performance with that of AI techniques, showing that the performance of AI algorithms is similar to that of clinicians, outperforming them when dealing with inexperienced clinicians.

Regarding the diagnosis problem, to the best of our knowledge, there are only three models that combine several diseases: [4], [6] and [3]. In this latter model, the authors used ML and different classifiers for experiments and obtained an accuracy of 84.9% utilising a combination of two modalities.

In [4], a preprocessing of medical texts is performed, and symptomatic entities are identified. For this processing, tools such as MIMIC-III (used to eliminate the portions of the text where there are no symptoms) and MetaMap (a Natural Language Processing (NPL) tool to identify symptoms extracted from the complete medical texts) are used.

After the preprocessing is done, the authors make a vector representation of the symptoms. The strength of association of each symptom with each disease is obtained (using TF-IDF) and used as a feature in the vector. The data obtained is then used to train a Bi-LSTM multi-label classification model. The preprocessing developed in this work involved assigning different weights to the various symptoms, culminating in a data set with cases given by weighted symptom vectors, ready to be used for model training.

With the 50 and 100 most common diseases being treated, this problem is viewed by the authors as multi-label classification, and the algorithm's performance is evaluated based on four metrics calculated using scikit-learn: precision (also known as positive predictive value), recall (also known as sensitivity), F1-score, and area under the curve (AUC) . Our model shares similarities with this one in that we evaluated it using some of these metrics (precision, recall and AUC) and specificity, negative predictive value, and accuracy.

When training the model, the authors use binary cross-entropy loss function and Adam Optimizer. The LSTM model has 100 hidden nodes and uses a dynamical mechanism with 50-time steps and drop-out strategies (where neurons are randomly chosen to be ignored during training).

Another work related to this one relies on automatic ICD-9 coding using Deep Learning [5], which in this case does not use symptoms but the interpretation of medical texts (DeepLabeler model). The model developed by the authors outperforms the DeepLabeler model, showing the importance of symptoms when inferring diseases. Although medical texts have most of the information available, extracting evidence of interest for diagnosis is difficult. Symptoms follow the process from illness to cure and are a significant source of knowledge regarding diagnosis.

Our work shares similarities with this study and with [4], as we use symptoms for diagnosis. In addition, we used a data set that matches a set of symptoms to a medical diagnosis. The preprocessing will be done using only Python, individualizing the symptoms as features.

3 Materials

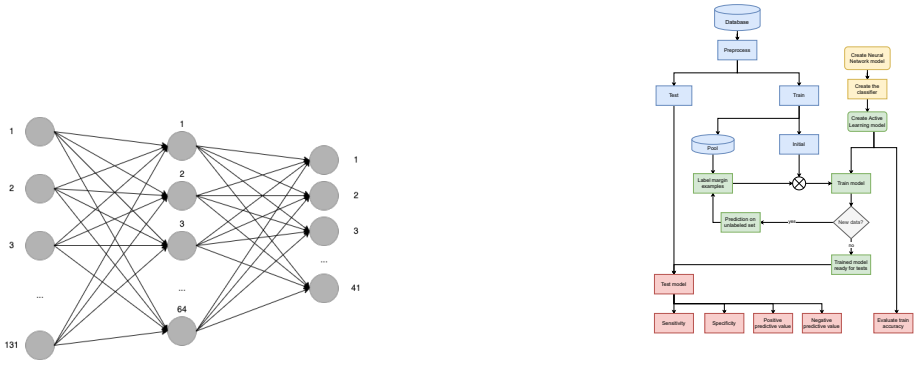
To build the model, we used the "Disease Symptom Prediction" [1] data set, which includes 41 different diseases and up to 17 symptoms. We have 4920 samples before preprocessing the data to train and test our model. There are 120 cases of each disease, and the data set is balanced. We also have a description of the

[1] `https://www.kaggle.com/itachi9604/disease-symptom-description-dataset?select=dataset.csv`

diseases, the precautions to be taken and the severity of the symptoms. The AL framework used in this work was modAL [2], which is geared towards Python3.

4 Methods

An intelligent model capable of diagnosing several diseases with high performance can turn into a revolution in the medical field. Based on [10], Neural Networks (NN) algorithms are the most used in this type of problem, but to the best of our knowledge it has not been used in conjunction with AL techniques. To unite AL and NN, we followed the following steps. First, we preprocessed the data set, turning the several symptoms into features and attributing weights for each symptom. Next, we built our neural classifier and used it on our AL model. A representation of the developed NN can be found in Figure 1a. An input layer with 131 neurons (which concerns the total number of symptoms), a hidden layer with 64 neurons and an output layer with 41 neurons (corresponding to the 41 diseases) were used. The layers are all fully connected.



(a) Neural network representation. (b) Block diagram.

Fig. 1: Methodology: Neural network and block diagram.

Figure 1b presents the flow diagram of the proposed methodology. This study was conducted using Python (4.2.5) and the modAL library (0.4.1) precisely for the AL component.

4.1 Preprocessing

The data set we used comprises a list of symptoms and the correspondent disease. In each case (sample), the number of symptoms varies. We also had a file filled with the information about the severity of the symptoms on a scale of 1 up to 7, being seven the most serious symptom. Considering that each sample had a different number of symptoms and that symptoms were strings, it would be hard

to add weight to each one of them, and it would also be more difficult for the classifier to be trained. To overcome these difficulties, we turned the 17 unique symptoms into features and filled the columns with 0's (if that sample does not have the symptom) and 1's (if that sample has the symptom). After that, we attributed the respective weight (severity) for each column (symptom).

4.2 Active Learning Component

To develop our AL model, we created a classifier and defined some parameters. As a result of the capability of NN to achieve excellent performance and analyze a vast amount of data, we decide to use it as our classifier. The Pytorch library was used to build a simple two fully connected layers classifier, i.e., two linear layers which apply a linear transformation to the incoming data. The loss function chosen for the classifier was the CrossEntropyLoss which computes the cross-entropy loss between input and target. The optimizer used was Adam, a stochastic optimization method.

Beyond the classifier, we also had to define some parameters to our model, such as the query strategy, the number of epochs of the model at its creation (Epochs Learner), and at the Active Learning loop (Epochs AL Loop), the number of instances and queries. Some experiments were done for all those parameters until we got the best values for them, as described in 4.4. The number of learner epochs defines the number of complete cycles over the training data set at the time the model is created (an object of the ActiveLearner class). The number of AL Loop epochs refers to the number of complete cycles over the training instances passed to the model at the training time.

4.3 Metrics

As stated in previous sections, the model we developed is compared with others already developed and used in the literature. To this end, the model's performance is evaluated based on several metrics: sensitivity, specificity, positive and negative predictive value, and accuracy. The model is also compared and evaluated using the AUC. Sensitivity, specificity, positive and negative predictive value, and accuracy correspond to micro-averaging measurements since it is a multi-class problem where the classes are not equally represented in the data set.

The rationale for choosing these metrics is directly related to the medical domain in which this work lies. Sensitivity is crucial in a medical context since it corresponds to the ability to identify an individual's disease (a late diagnosis or the absence of a diagnosis could lead to the patient's death). On the other hand, specificity in a medical context translates to the ability to exclude disease hypotheses in healthy individuals. While not as striking as the previous metric, it may correspond not to subjecting a healthy individual to unnecessarily invasive or extremely aggressive treatments.

4.4 Experimental Design

The implementation phase began by processing the data set used, as reported in Section 4.1. The development of the classifier model using AL and (Pytorch) followed. Once the model was developed, we made tests to find the best combination of parameters for it. We tried different values for the epoch, instances, and also three different sampling methods: least confident (LC), entropy and margin sampling. The parameters' values chosen were the number of epochs of the model at its creation (30), the number of epochs of the model in the AL loop (40), the sampling strategy (margin sampling), the number of instances that are passed to the model when it is learning in the AL loop (10), the number of queries made in the same loop (10), and the number of layers of the NN in use (two as described at 4.2). The model was trained and tested ten times for each parameter set (to ensure that the results were statistically significant). Then we evaluated the best parameter set based on the average accuracy for both training and testing.

5 Results

We tested a combination of all parameters described in Section 4.4. The best results for a network of 1, 2, and 3 layers correspond to those on Table 1. These results consider the model's performance from the time it is created until it completes the AL loop. In this loop, the model is updated according to the new data it receives, and its performance is successively improved. In many tests, we verified that this performance reached 100% quickly, reflecting that the model was overfitting.

In the tests, it was impossible to have the individual perception since they were done all at once. Although, for these combinations of parameters and when the remaining metrics were assessed (namely sensitivity and specificity), the accuracy was good, leading us to realize that the model should be overfitting. Due to this fact, these were not the parameters adopted for the final version, as seen in Section 4.4. Even so, these results showed that the network with the best results was the two layers network (id two and id three). Therefore, from this point on, we focused only on this network to find the best combination of parameters.

The strategy adopted to find the best set of parameters that did not overfit was to go through the combinations from the best to the successively worse ones and evaluate those with good sensitivity and specificity values, not all of which were equal to 1.

We found the best results to arise almost entirely for several instances passed to the model in the AL loop. However, we also found that overfitting almost always (if not always) occurred when that number of instances is 15. Thus, we considered only the best cases for which the number of instances differed from 15. The sets of parameters assessed individually and their respective performances are shown, in order, in Table 2. For all the sets in Table 2, the sampling strategy

was margin sampling, the number of instances was 10, and the number of queries was 10.

Table 1: Best combinations of parameters.

id	Layers	Epochs (Learner)	Epochs (AL loop)	Sampling strategy	Instances to train	Queries	Train accuracy	Test accuracy
1	1	40	20	Margin	10	10	56.02%	54.70%
2	2	10	40	Margin	15	10	61.15%	57.86%
3	2	50	40	LC	15	10	59.31%	60.03%
4	3	40	50	LC	15	10	58.39%	58.37%
5	3	20	50	LC	10	10	57.51%	59.65%
6	3	30	10	LC	10	10	56.77%	58.92%
7	3	40	50	Margin	10	10	58.39%	58.37%

Table 2: More parameters combinations and its respective accuracies.

id	Epochs (Learner)	Epochs (AL loop)	Train accuracy	Test accuracy
1	10	10	55.82%	54.57%
2	50	40	55.55%	54.04%
3	50	30	54.99%	54.10%
4	30	20	54.90%	53.43%
5	30	20	53.05%	52.32%
6	10	40	52.72%	50.91%
7	50	10	52.25%	49.46%
8	50	50	51.65%	50.46%
9	50	20	51.58%	50.92%
10	30	30	51.54%	50.26%
11	30	40	51.49%	49.27%

When evaluating the performance of the first ten models in the previous tables, we realized that the results, although good, could be better without falling into overfitting. We then evaluated them until the 11^{th} model, which appeared to have the best results without overfitting. This was then the model adopted as the final. It should be noted that, in the medical field, the most relevant metric should be the sensitivity, which, for the model in question, reaches values around 90%. Table 4 shows an example of the results generated by the model defined with the previous parameters. We will discuss these results in the following section.

Table 3: Comparison between different models. Guo et al.'s model [4]

	Our model (41 diseases)	Guo et al.'s Model (50 diseases)	Guo et al.'s (100 diseases)
Micro-Precision	92.07%	50.80%	47.20%
Micro-Recall	92.07%	63.20%	52.90%

6 Discussion

The accuracy results at Table 1 are the mean of the models' accuracy during the entire loop. It starts with low values until reaching a great accuracy. Nevertheless, this metric only evaluates the mean, not the final result, considering that the values obtained in Table 4 are consistent with what was expected and closer to the final values of the accuracy of the model.

By analysing the results obtained with the trained model, the variation in classification difficulty between the various diseases is evident. While in diseases such as Tuberculosis, Pneumonia or Heart Attack the model is infallible (that is, it correctly classifies all cases), in cases such as Hepatitis A the model has extreme difficulty in identifying the pathology (the model cannot identify a single case of this pathology).

A small test set can be the reason for the values obtained for some of those metrics, i.e., if we had just one sample of that disease in the test set, a wrong/right classification would lead us to zero/one, and it is not enough to evaluate the model correctly.

Since sensitivity is the most relevant metric in the medical field (it dictates how good the model is at detecting the disease in people who are actually sick), an overall value of 92.07% is quite favourable in general practice since 41 distinct diseases are included here. Even so, the poor results for the illnesses mentioned above indicate that the present model is unsuitable for their classification. Still, the diseases for which the model presents more difficulties can be selected, and we can build one or several new, more specific models for these pathologies. Also, for further study, using a more extensive data set could improve the model's performance.

An average specificity of 99.80% tells us that our model is quite good at identifying negative cases as negative, so it is possible to use it in a medical context which could involve not the detection of the disease but the screening of various illnesses in the diagnosis.

Regarding state of the art in the area (depicted in Section 2) and starting with the work of [3], it can be seen that using ML and testing several types of classifiers, the authors obtained a model with 84.9% accuracy, combining these two modalities. As shown in Table 3, our model got an average accuracy of 99.61%, and the added complexity of the work should be highlighted, given the need to test several classifiers (in this work, we used only NN).

Regarding the work developed by [4], one can see in Section 2 the increased work in data preprocessing compared to our model. The authors do the process-

Table 4: Metrics for each disease obtained by testing the model created. SS - sensibility, SP - specificity, PPV - positive predicted value, NPV - negative predicted value, ACC - accuracy

Disease	**SS**	**SP**	**PPV**	**NPV**	**ACC**
Allergy	1	1	1	1	1
GERD	1	1	1	1	1
Chronic cholestasis	0	1	0	0.979675	0.979675
Drug Reaction	1	1	1	1	1
Peptic ulcer disease	1	1	1	1	1
AIDS	1	1	1	1	1
Diabetes	1	1	1	1	1
Gastroenteritis	1	1	1	1	1
Bronchial Asthma	1	1	1	1	1
Hypertension	1	1	1	1	1
Migraine	1	1	1	1	1
Cervical spondylosis	1	1	1	1	1
Paralysis (brain hemorrhage)	1	1	1	1	1
Jaundice	1	1	1	1	1
Malaria	1	1	1	1	1
Chicken pox	1	1	1	1	1
Dengue	1	1	1	1	1
Typhoid	1	1	1	1	1
Hepatitis A	0	1	0	0.973577	0.973577
Hepatitis B	1	1	1	1	1
Hepatitis C	1	1	1	1	1
Hepatitis D	1	0.952233	0.313433	1	0.953252
Hepatitis E	1	1	1	1	1
Alcoholic hepatitis	1	1	1	1	1
Tuberculosis	1	1	1	1	1
Common Cold	1	1	1	1	1
Pneumonia	1	1	1	1	1
Dimorphic hemmorhoids	1	1	1	1	1
Heart attack	1	1	1	1	1
Varicose veins	1	1	1	1	1
Hypothyroidism	1	1	1	1	1
Hyperthyroidism	1	1	1	1	1
Hypoglycemia	1	1	1	1	1
Osteoarthristis	0	1	0	0.96748	0.96748
Arthritis	1	0.966597	0.448276	1	0.96748
Paroymsal Positional Vertigo	1	1	1	1	1
Acne	1	1	1	1	1
Urinary tract infection	1	1	1	1	1
Psoriasis	1	1	1	1	1
Impetigo	1	1	1	1	1
Micro-averaging measure:	92.07%	99.80%	92.07%	99.80%	99.61%

ing of medical reports, from which they extract symptom entities and, based on this, generate symptom vectors that will then be used in the construction of the model. For this purpose, Guo et al. have to use three different tools. The present work uses a data set that relates several diseases with their respective symptoms, making the preprocessing task much more accessible. Although the most common in clinical settings is still the writing of medical reports, developing an interface that allows the insertion of symptoms in a practical and fast way would allow the direct use of our model and its direct response. Something in common between the work developed by those authors in the literature and the work developed by us is the attribution of weights to symptoms which we consider to be an added value in the detection of the correct pathology. Comparing the models in terms of complexity, one can see that [4] developed a NN as well, but, in this case, a Bi-LSTM with 100 hidden neurons, with a dynamic mechanism with 50-time steps and dropout strategies. The present work developed a NN with a sequential container with two layers, both with a linear activation function.

The metrics obtained for each model are presented in Table 3. As can be seen, the model we developed shows better values both for sensitivity and positive predictive value. The AUC also shows significant differences between our model and Guo et al. models (0.99 vs 0.853 and 0.854). However, we should not directly compare these values since we could not calculate the micro-AUC while Guo et al. did it. It should also be noted that the results mentioned refer to different databases. So, to have a more accurate comparison, it would be interesting to apply the model developed to the database used by the [4].

In Figure 2, it can be observed the variation of the accuracy obtained in the test based on the number of instances that the model has already used for training. We did this study for the best model found (mentioned before) being presented its performance for each of the three sampling strategies studied.

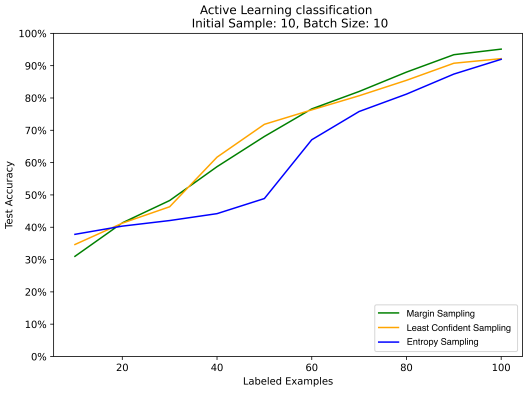


Fig. 2: Evolution of the test accuracy with the number of classified examples.

The margin sampling is the one that presents the best results at the end of the ten queries. This strategy assumes that the most informative data samples are those which fall within this margin, so accordingly to the results is a good option for our problem, which has several classes and, consequently, several support vectors to separate them. Still, we can see that excellent accuracy values are reached with few instances (2.5% of the data set), and, when exceeding this value, the model already tends to enter in overfitting.

In [1] a model was developed using a Support Vector Classifier to predict the same diseases based on the same data set. To train this classifier, the authors use 85% of the data set (which corresponds to 4182 training samples) and obtain a test accuracy of 94.72%. In the present work, only 100 cases are used, and accuracy values of the same order are obtained, which can be considered a remarkable positive consequence of applying AL.

7 Conclusion

The model proved to be quite good and, compared with the models developed so far, gave evidence of surpassing them. These facts make it clear that using AL in this area is an asset since it allows the creation of models with better performance while maintaining simplicity. As can be observed in Figure 2, with a few instances (100), our model is capable of achieving a high value of accuracy (>90%). This approach allows us to construct models that demand a lower computational power than usual, as we need to train only with 2.5% of the data set. This achievement becomes even more significant if we also consider the cost for medical experts to build data sets.

As this is a symptom-based diagnostic model, diagnosing the pathology in question in the early stages of the disease or asymptomatic cases may not be easy or even possible. Nevertheless, a model similar to this could be beneficial to assist the physicians in their daily practice.

Since the data set in use has, for each disease, only symptoms that can be associated with the disease (and not extra symptoms that, although not associated with the disease, are often reported by patients), it is thought that this data set already corresponds to the treatment of data in which there should already be some noise component, noise being understood as symptoms not associated with the diagnosed disease. Noise data would be more real than the data set used. Therefore, it may be interesting to evaluate the performance of this classifier with this type of data, or even to re-train the architecture using data with noise.

Each disease is associated with a limited number of symptoms, and the various cases of each disease correspond to combinations of those symptoms. There are then cases of each disease consisting of the same symptoms. On the one hand, it is possible to have quite distinct sets of symptoms among several diseases, making it simple to determine the disease in question. On the other hand, and due to the existence of equal cases, it may happen that there are equal cases in the training and test sets, making them similar (in a very extreme case, it would

be the equivalent to evaluate the classifier with the training set). Therefore, it would be of interest, in future work, to explore data sets with more diversified cases for each disease, with the inclusion of symptoms corresponding to noise, and to evaluate the performance of the classifier developed under these conditions. The addition of noise to cases should add difficulty to classification and a better representation of the real world, in which patients very often report more symptoms than those associated with the disease which they are diagnosed.

Although the data currently used corresponds to real data, we consider that it will be relevant to test the classifier in real-world situations, as well as with different data sets and methodologies, namely, different distributed classes, possibly larger data sets, and different query strategies.

References

1. Disease type prediction using symptoms, https://www.kaggle.com/naga26/disease-type-prediction-using-symptoms (2020)
2. Danka, T.: modal: A modular active learning framework for python3, https://modal-python.readthedocs.io/en/latest/ (accessed: 29.10.2021) (2018)
3. Faris, H., Habib, M., Faris, M., Elayan, H., Alomari, A.: An intelligent multimodal medical diagnosis system based on patients' medical questions and structured symptoms for telemedicine. Informatics in Medicine Unlocked **23** (2021). https://doi.org/10.1016/j.imu.2021.100513
4. Guo, D., Li, M., Yu, Y., Li, Y., Duan, G., Wu, F.X., Wang, J.: Disease Inference with Symptom Extraction and Bidirectional Recurrent Neural Network. Proceedings - 2018 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2018 pp. 864–868 (2019). https://doi.org/10.1109/BIBM.2018.8621182
5. Li, M., Fei, Z., Zeng, M., Wu, F.X., Li, Y., Pan, Y., Wang, J.: Automated ICD-9 coding via a deep learning approach. IEEE/ACM Transactions on Computational Biology and Bioinformatics **16**(4), 1193–1202 (2019). https://doi.org/10.1109/TCBB.2018.2817488
6. Mangiameli, P., West, D., Rampal, R.: Model selection for medical diagnosis decision support systems. Decision Support Systems **36**(3), 247–259 (2004). https://doi.org/10.1016/S0167-9236(02)00143-4
7. Ren, P., Xiao, Y., Chang, X., Huang, P., Li, Z., Chen, X., Wang, X.: A survey of deep active learning. CoRR **abs/2009.00236** (2020), https://arxiv.org/abs/2009.00236
8. Settles, B.: From theories to queries. In: Guyon, I., Cawley, G.C., Dror, G., Lemaire, V., Statnikov, A.R. (eds.) Active Learning and Experimental Design workshop, In conjunction with AISTATS 2010, Sardinia, Italy, May 16, 2010. JMLR Proceedings, vol. 16, pp. 1–18. JMLR.org (2011), http://proceedings.mlr.press/v16/settles11a/settles11a.pdf
9. Settles, B.: Active Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers (2012). https://doi.org/10.2200/S00429ED1V01Y201207AIM018
10. Shen, J., Zhang, C.J., Jiang, B., Chen, J., Song, J., Liu, Z., He, Z., Wong, S.Y., Fang, P.H., Ming, W.K.: Artificial intelligence versus clinicians in disease diagnosis: Systematic review. JMIR Medical Informatics **7**(3), 1–15 (2019). https://doi.org/10.2196/10010