IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 27 July 2023, accepted 23 August 2023, date of publication 28 August 2023, date of current version 5 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3309157

RESEARCH ARTICLE

Threshold-Learned CNN for Multi-Label Text Classification of Electronic Health Records

ZHEN YANG AND FRANK EMMERT-STREIB
Predictive Society and Data Analytics Laboratory, Faculty of Information Technology and Communication Sciences, Tampere University, 33014 Tampere, Finland
Corresponding author: Frank Emmert-Streib (v@bio-complexity.com)

The work of Zhen Yang was supported by the Faculty of Information Technology and Communication Sciences, Tampere University.

ABSTRACT Text data in the form of natural language is a valuable resource that contains domain-specific information applicable to various applications. An example are electronic health records (eHR) offering comprehensive insights into patients' health histories, enabling knowledge extraction for clinical diagnosis and treatment. In this paper, we study multi-label text classification (MLTC) of eHR data by introducing two novel MLTC methods based on a threshold-learned convolutional neural network (CNN). We conduct comprehensive comparisons with other multi-label models and binary relevance (BR). Importantly, we do not only optimize the architecture of multi-label classifiers but also of the baseline BR model. As a result, our findings indicate that the adaptive-threshold CNN (AT-CNN) and implicit-threshold CNN (IT-CNN) provide a favorable approximation of a binary CNN (B-CNN) with the added benefit of improved runtime efficiency. The latter is crucial when the number of classes grows larger because the runtime of classifiers based on one-vs-rest mappings becomes increasingly prohibitive for such configurations.

INDEX TERMS Data science, multi-label classification, deep learning, natural language processing.

I. INTRODUCTION

Electronic health records (eHR) hold rich information about patients. Such records contain diagnostic and biomedical notes from clinicians and nurses providing indispensable information for identifying proper treatment actions of patients based on their health history [21], [48]. However, going manually through thousands of potentially long records is a time-consuming process requiring large amounts of resources. For this reason, there is an urgent need for an automatic procedure that can exploit the increasing number of eHR by turning them into a form of ready information that can inform subsequent clinical tasks. In order to approach this problem natural language processing (NLP) can be used and many remarkable results have been achieved on diverse tasks. For instance, NLP has been used for disease classification [2], [33], [43], [44], [50], disease events prediction [6], [57], and medical information extraction [17], [18], [46].

An important task for analyzing eHR data is text classification [3], [27], [32], [42], which is also the main objective of this paper. While the primary use of eHR in the clinic is for a particular patient, analyzing a large corpus of eHRs from thousands of patients can lead to the identification of population-specific properties that may inform our understanding of disorders beyond the individual patient. It is this latter purpose for which text classification of eHRs is of primary use. Conceptually, approaches behind text classification can be categorized into three different groups: binary classification, multiclass classification and multi-label classification. For multiclass classification, only one label can be assign to an instance within a set of available labels whereas each label is mutually exclusive. When the number of classes is two this becomes binary classification. In contrast, for multi-label classification, more than one label can be assigned to an instance [12]. Multi-label classification is also the focus of this paper.

While multi-label classification is an intriguing concept, it's practical realization is far from trivial and widely underexplored. A traditional approach for multi-label text classification is binary relevance (BR) [4]. This approach transforms a multi-label learning problem into multiple (corresponding to the same number as the classes) binary classifications. That

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

means for each class, a classifier is trained as ''one-versus the rest''. This type of approach is favoured by many and considered the most intuitive way for dealing with multi-label classification problems [52].

On a downside, the BR approach makes the assumption that each class is independent from the others. Hence, possible relations between classes are not explored but suppressed by neglecting dependencies which might carry exploitable information [55]. For example, it is known that a patient that has obesity is more likely to have heart disease, and people that have anxiety usually are accompanied by depression. Such health examples show that most of the time disorder classes appear in pairs or even high-order class-combinations. On the other hand, some diseases could be mutually exclusive. Either of those examples contradicts the assumption of the independence of classes underlying the BR models. Looking at correlations between class labels allows to divide approaches into three categories. First-order approaches deal only with one label individually, hence, they do not consider a label dependency. Binary relevance is an example of a first-order approach. In contrast, second-order approaches incorporate pairwise correlations between labels whereas higher-order approaches consider even more relations between labels.

In order to overcome the limitations of first-order approaches using binary classifiers, it has been suggested to form chains by stacking multiple binary classifiers in a specific chaining order [40]. Such algorithms organize the chain structure according to prior knowledge and information including label-dependencies. Commonly, each subsequent classifier is built upon the predictions of preceding ones, hence, forming an approach that is able to utilize high-order label-dependencies. Unfortunately, the complexity of classifier chains grows exponential with the number of classes. Also, the first few predictions play essential roles as the captured label-dependencies formed by the chain structure will largely depends upon the early predictions.

Another category is algorithm adaptation that modifies traditional binary classifiers to make them fit directly to multi-label problems. Examples therefor are, multi-label k-Nearest Neighbors [54], multi-label Decision Trees [7] and Ranking Support Vector Machines [11]. However, most of the algorithm adaption methods remain inferior, especially to modern deep learning based methods, as they are limited to modelling only first- or second-order label dependencies.

Importantly, in recent years deep learning [13] approaches have been widely applied to problems of text classification demonstrating impressive improvements over traditional methods. Examples for deep learning architectures are convolutional neural networks (CNN) [14], [23], [25], [37] and recurrent neural networks (RNN) [17], [29], [47]. Further novel architectures are provided by transformer models, e.g., BERT [9] and elmo [39]. Regarding multi-label classification, neural network models do not need to transform the multi-label problem into binary problems because their architectures should allow to learn labels and capture label-dependencies in higher layers [34].

In this paper, we study multi-label text classification and introduce two new deep learning models based on a multi-label CNN architecture called implicit-threshold CNN (IT-CNN) and adaptive-threshold CNN (AT-CNN). Specifically, we design dynamic learning thresholds to select the output labels for different samples regarding different labels. In addition, we compare our proposed architectures with a binary relevance CNN architecture to examine the capability as well as the scalability of our thresholds multi-label learning models. For our analysis, we use annotated data from the MIMIC-III database [15] containing 1610 free-text structure notes that are divided into 10 different phenotypes.

This paper is organized as follows: In the next section, we review related multi-label learning methods from the literature. In the Methods section, we discuss all methods we use for our analysis and introduce two novel methods. Furthermore, we discuss word embedding methods to transform raw text into numeric distributed representations. In the Results section, we presents our experimental discovers. Specifically, we study: (1) Optimization of CNNs, (2) Model comparison between eight models for various error measures. (2) Difficulty levels of multi-label classification by introducing noise levels. (3) Subclass classifications of the top performing multi-label CNNs and B-CNN trained on all sub-class combinations. (4) Time complexity of the runtime of the methods. In the Discussion section, we connect our results to findings from the literature and interpret our findings. Finally, this paper finishes with concluding remarks.

II. RELATED WORK

In this section, we review related work in more detail. This will later enhance our discussion when comparing such approaches with the results from our models.

Backpropagation for Multilabel Learning (BP-MLL) [53] is considered the first attempt to solve a multi-label classification task by a neural network architecture. The method utilizes a pair-wise loss function that incorporates label-occurrence information into the training. This learning framework was further improved by [34] where they compensated limitations of the pair-wise loss by using cross entropy which showed faster convergence. In addition, they used a Relu activation function and applied drop-out in their training. Both techniques are very common in training neural network making their architecture a popular neural network framework for multi-label classification for its simplicity and effectiveness [28]. Typically, neural network multi-label classification models include a learning module and a prediction module. The learning module transforms input sequences into vectors of features by a specific feature learner which can be any type of neural network architectures such as a CNN, RNN, or BERT. The prediction module will pass the learned features into an one-layer-perceptron where the number of nodes corresponds to the output labels. This gives

a confidence score for each class whereas high scores indicate relevant classes and low scores indicate irrelevant classes. For given scores, a threshold function is applied to select the best classes. The threshold function can either be learnt [10], [34] or set to a fixed value.

Despite the superior performance of neural networks over conventional machine learning methods for multi-label classification, vanilla neural network structures somehow still neglect label correlations during the training. For this reason, recently, more effort has been placed on this investigating how to better incorporate label-dependencies into neural networks. It is interesting to note that for image processing, the paper by [26] proposed a novel neural network structure embedded with a label-decision module. Following this work, in [10] a modified framework for classifying biomedical text data was introduced using also a label-decision module for predicting the number of true labels per sample. Hence, this module acts as a threshold function over the ranking scores produced by the prediction module to select the best combination of labels.

A novel way to address multi-label classification has been introduced by SGM (sequence generation model for multi-label classification) [49] by making it a sequence generation problem. Specifically, SGM adapts a Bidirectional LSTM (BiLSTM) sequential model along with attention to the text sequences. The learned features are used to predict each label individually using a specific chaining order and the next prediction of the label is conditioned on the previous one. Additionally, SGM uses a global-embedding as one memory mechanism to capture all the preceding label information at a current time-step to alleviate the punishment of wrong predictions from early time-steps. This is important because incorrect early predictions are most likely to result in a succession of wrong predictions in later time-steps, which is known to be a source of bias effecting the sequential chaining prediction structures. It has been shown that SGM is better in modeling label-dependencies compared to other network architectures including conventional BR, vanilla CNN, CNN-RNN [49].

In [51] a deep learning based method has been introduced that incorporates second-order label-occurrence information into the network. The label-occurrence information is mapped into vectors multiplied by the feature vectors learned by the feature extractors. In this way the network learns about the label-dependencies. They showed that their architecture can be embedded into many popular feature extractors such as CNN, RNN or BERT, and their results demonstrated that networks embedded with such a structure outperform vanilla versions. This indicates the large potential of utilizing label-dependencies which have not been fully utilized by most modern neural network architectures.

Another method introducing a novel idea is MAGNET (Multi-label text classification using attention based graph neural network) [36]. This method leverages a graph attention mechanism for label correlations. Using prior knowledge of the label-occurrences to build up a graph attention network allows their model to learn high-order dependencies. As feature extractors they use BERT and BiLSTM in combination with label features extracted by a graph attention network. They showed that their model is able to learn both higher-level contextual meaning of documents as well as correlations between labels. Also their model obtained competitive results compared with several state-of-the-art models and different multi-label classification benchmark datasets.

Interestingly, Ma et al. [30] argued that most current approaches for multi-label classification are not capable to distinguish between similar labels, and are thus failing to capture semantic label correlations. To improve upon this they proposed a label-specific dual graph neural network (LDGN) [30]. LDGN uses a BILSTM as feature learner and an attention mechanism to generate label-specific vectors. The network is capable of modeling label-dependencies from the input documents by combining label-occurrencies with label-specific vectors into a graph convolutional network, where the final output can capture correlations between labels. As a result they outperformed several state-of-the-art models which are designed also to capture label dependencies.

Lastly, Zhang et al. [56] introduced auxiliary prediction tasks for multi-label learning in addition to the classification of the labels to improve the performance of multi-label classification. Their method is based on a BERT architecture, combining text and label embedding representations to form an additional label co-occurrence prediction task. Furthermore, they use the outcome from a prediction as feedback to enhance the multi-label learning of the method. Their results showned that it is a top-performing method for the AAPD dataset, and also very competitive on RCV1-V2 dataset.

III. METHODS

In this section, we discuss all methods and data we use in this study.

A. CONVOLUTIONAL NEURAL NETWORK

The ideas behind different variants of neural networks is to learn unique types of representative features. For instance, a convolutional neural network (CNN) aims to learn regional features from adjacent inputs using different sizes of filters. By stack multiple filters on top of each other, more distant features are jointly learnt to extract more abstract features from larger regions. The most significant advantage of a CNN is to significantly reduce the number of parameters needed for building a network, as the neurons are not fully connected from its previous inputs, the inputs can be viewed as a plane, and the filter scans the input thoroughly with a specific window size and step size, where each step results in locally connected weights with the neurons within the region. As a result, the number of connected weights required are significantly reduced.

Let us consider an input of 2 dimensions $I \in R^{2\times 50}$ going through a fully connected layer of 100 neurons. Then the number of weights needed for a full connection layer is 10, 000, however, if we use a filter of window size 2×2, using a horizontal step of 1 to cross all the local regions of the input, this results in a total of 196 weights only. Usually one would assign multiple filters with different length sizes to tackle all the possible regional features, nevertheless a CNN is much more efficient than traditional neural network architectures.

The first well-known convolutional neural network was Imagenet [24] which was originally proposed for dealing with computer vision problem and showed leading performance by a considerably large margin over traditional methods, its special way of handling inputs agree with the property of images, where spatial features are very region specific, using filter to highlight these regional features and feed them into a chain of filters to extract more abstract representations greatly help the network to distinguish between different patterns in the image and learn to recognize different objects. Later researches found that CNN can also be used to handle textual input in a 2-D manner to consider combinations of different length of words to help the network to understand meanings [23]. In our study, we extend a CNN architecture by adding and learning thresholding mechanisms (described in the next section).

For the word embedding, we use word2vec [31] a widely used word embedding method [19] to transform all words to unique vector representations. The method is quite fast allowing to obtain the embeddings efficiently.

B. THRESHOLD LEARNED CNN

In this section, we discuss the structure of our novel models, called AT-CNN and IT-CNN. Both models are based on a basic feature learner CNN extended by multi-label threshold functions added to the basic feature learner.

The base CNN we use for all models has a structure similar as in [23] and [50]. This structure is illustrated in Fig. 1(a). Let x denote the input, raw texts will be divided into different tokens with represents an unique symbol, first layer of the network will accept these tokens and convert them into vectors, each token will be $x_i \in R^{50}$, $i = [1, 2, 3, \ldots n]$ and n is the maximum number of tokens of the input sample and we use the embedding size of 50 in our experiments, after which multiple convolutional filters of varying window sizes will be applied to the input with step size of 1 to go across the entire input to extract abstract features, convolutional filters of window size $N \times 50$ will extract features from the corresponding N-gram from word embedding of size 50, a collection of features resulted one filter is called feature map, hence filter of $N \times 50$ with a step 1 will result in feature maps of size $n - N + 1$, additionally, one can assign multiple filters with the same window sizes to the input, though with the same size the parameters within each filter can be different, thus enabling different filters with the same window size to extract different types of features from same region of the input, after convolutional layer we wil have a collection of feature maps $C = [C_1, C_2, C_3 \ldots\ldots C_m]$, $C_m \in R^{n-N+1}$ and m is the total number of different filters, as a rule of thumb, a convolutional layer is commonly followed by a pooling layer, as we know a feature map containing a number of features, pooling operation usually select more essential features from each feature map by taking either the maximum, average, or minimal values, in our experiment, we use a maximum pooling to extract only 1 feature with the maximum value from each feature map, hence the final feature representation $f \in R^m$ is learned through CNN.

The learned features can be used to perform a classification task using either a binary or multi-label framework. The label-decision structure is based on [34] where cross-entropy, dropout and an advanced optimizer were added to the learning process, see Fig. 1(b). Label-decision modules can be categorized as binary and multi-label modules. Both of them utilize a fully-connected layer that takes the learned features as input and output the number of labels. The binary module has only two neurons at the output layer whereas the multi-label module has as many neurons as classes whereas each of the neurons indicates a confidence scores for the corresponding class.

For the multi-label module, one needs to select the threshold for making the final prediction of the class, which is usually a scalar value between 0 and 1. A lazy way would be to set a constant value of 0.5 as the global threshold for all the classes, however, this is generally a poor choice especially when the number of classes is high [35]. Usually, it is better to search for an optimal threshold for obtaining a better performance, even though this is usually non-trivial. In general, a dynamic threshold that can adjust their values according to different classes has been studied and shown to be more accurate than a constant value [10], [20].

In this paper, we propose and study two threshold functions as extension of a CNN. We call the resulting models AT-CNN and IT-CNN. Specifically, the first model uses an adaptive-threshold (AT), which utilizes adjusted binary confidence scores as the threshold for the multi-label classes. It operates by predicting each label individually based on the learned features. These predicted scores are then concatenated to form a threshold vector, which is used to select the positive labels from the predictions of the multi-label module. The second models uses as threshold function an implicit-threshold (IT) that predicts the number of positive classes and selects the top-k best scores from the multi-label module. It functions as an individual network, trained and optimized concurrently with the basic feature learner during the training process. The structure of both threshold functions can be seen in the Fig. 1(c).

In this study, we will use the following notation for the models:

1) Binary-CNN (B-CNN): This model utilizes binary classification.
2) Multi-label CNN (M-CNN): This network incorporates a multi-label module with a fixed threshold.

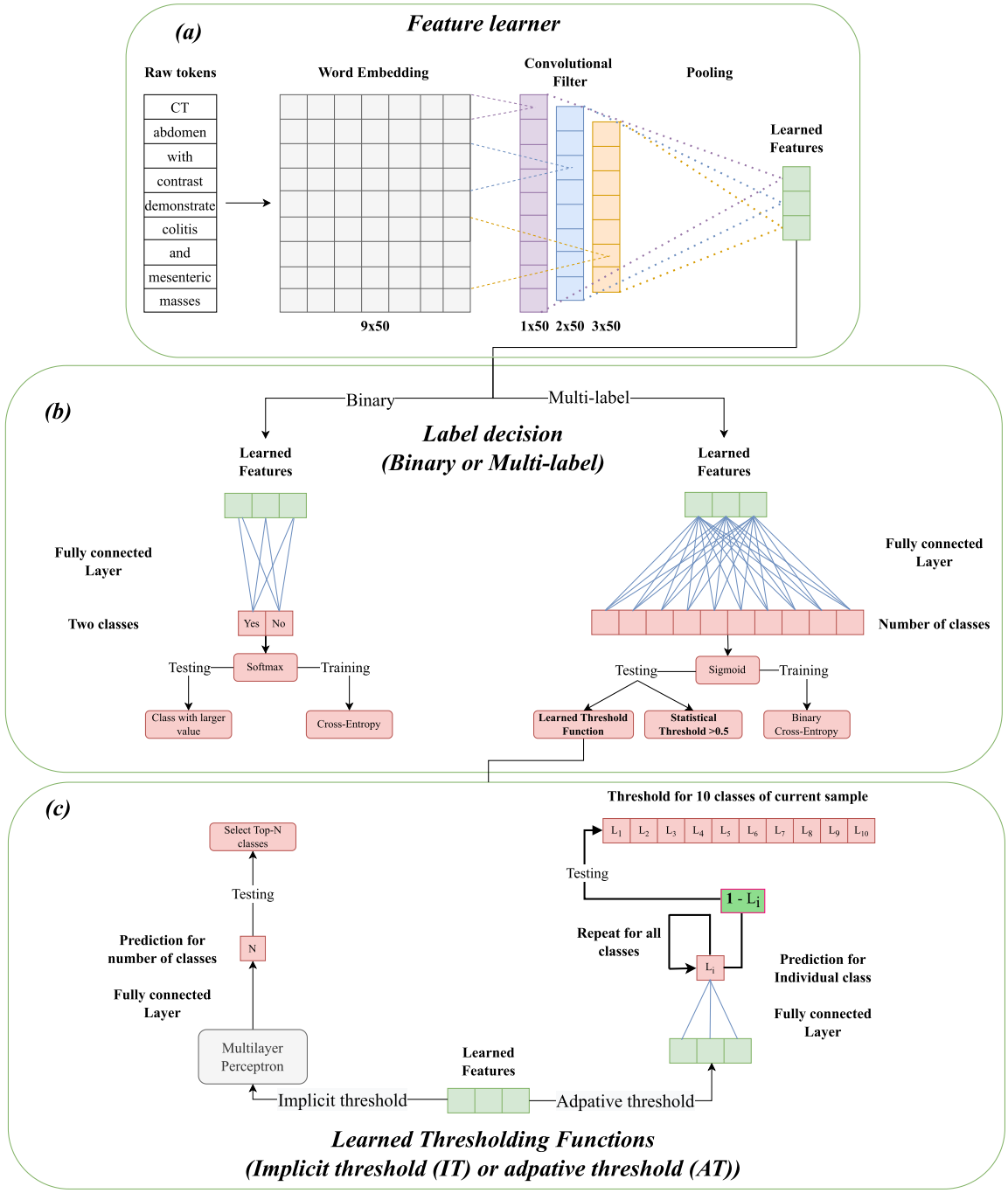

FIGURE 1. The base network architecture of the feature learner that utilizes a CNN [23], [50] is illustrated in part (a). The label-decision module for binary and multi-label learning is shown in part (b). Part (c) shows two thresholding functions we use to extend the base CNN (in (a)) for learning a multi-label classifier. Left: Implicit-threshold (IT) function leading to an IT-CNN. Right: Adaptive-threshold (AT) function leading to an AT-CNN.

3) Implicit-threshold CNN (IT-CNN): This refers to the multi-label CNN with Implicit-threshold.
4) Adaptive-threshold CNN (AT-CNN): This model employs Adaptive-threshold in the multi-label CNN.

C. DATA

For our study, we use the discharge summaries extracted from the MIMIC-III database. MIMIC-III stands for Medical Information Mart for Intensive Care [22], which is a freely accessible database that contains de-identified clinical data collected from more than 53, 000 hospital admissions for adult patients gathered from year 2001 to 2012. The data were collected at the Beth Israel Deaconess Medical Center in Boston, Massachusetts (USA). The MIMIC-III database contains a variety of patient records including structured, controlled vocabulary data such as laboratory notes, ICD

codes, and free-form texts such as progress notes, discharge summaries, and reports of electrocardiogram/imaging studies. In our study we focus on the free-form texts of discharge summaries containing the most informative clues for patient phenotyping [41].

The annotated discharge summaries are from [15]. They annotated 1610 discharge summaries from MIMIC-III dataset, where 415 discharge summaries from patients being a frequent flyer in the ICU (>=3 ICU visits within 365 days), and 313 random selected discharge summaries from the later visits of above frequent flyers. Additionally, 882 random discharges summaries were selected from those patients who are not ICU frequent flyers, yielding 1610 summaries from 1297 unique patients.

All the 1610 discharge summaries were annotated into 10 different phenotypes. For this several annotators of domain experts were used. It was ensured that each phenotype was labelled at least twice by different annotators to guarantee the most reliable label quality. In case of an uncertain phenotype, a senior clinician annotator decided on the final label. The frequencies from all 10 phenotypes ranges from 126 to 460 cases.

In order to measure the degree of agreement on the labels between different annotators, Cohen's Kappa [8] was provided. Let A denotes the number of samples both annotators agree on phenotype 1, B denotes the number of samples annotator 1 labeled phenotype 1 while annotator 2 labeled phenotype 2, C denotes samples annotator 1 labeled phenotype 2 while annotator 2 labeled phenotype 1, and D denotes samples both annotators agree on phenotype 2, then one can define:

$$P_0 = \frac{A+B}{A+B+C+D} \tag{1}$$

$$P_1 = \frac{A+B}{A+B+C+D} \cdot \frac{A+C}{A+B+C+D} \tag{2}$$

$$P_2 = \frac{C+D}{A+B+C+D} \cdot \frac{B+D}{A+B+C+D} \tag{3}$$

Based on these, Cohen's Kappa measure K can be defined as:

$$K = \frac{P_0 - P_1 + P_2}{1 - P_1 + P_2} \tag{4}$$

Cohen's Kappa is a scalar measure between 0 and 1 that indicates the degree of agreement between two annotators who classified N samples into C different categories. The higher the value of K is the more agreement in the annotations while lower values indicate conflicting classification requiring attention by by a senior annotator. The frequency of labels and the Cohen's Kappa measures for all 10 phenotypes used in this study are shown in Fig. 2.

D. EVALUATION MEASURES

Measures for evaluating multi-label classification problems can be grouped into two categories. (1) Sample-based measures: Such measures compare a predicted vector $y_s[y_1, y_2, y_3 \dots y_c]$, with the true vector $\hat{y}_i[\hat{y_1}, \hat{y_2}, \hat{y_3} \dots \hat{y_c}]$, where s represents an individual sample and c is the total number of classes. An example for such a measure is F-sample (defined below). (2) Label-based measures: Rather than considering each sample individually, label-based measures compare a predicted sub-class vector from all samples against true sub-class vector. In this case, we compare a predicted vector $y_c[y_1, y_2, y_3 \dots y_n]$ with the true vector $\hat{y_c}[\hat{y_1}, \hat{y_2}, \hat{y_3} \dots \hat{y_s}]$, where c represents an individual label and s is the total number of samples. For this type of measure, we use F-micro and F-macro.

In general, a F-score takes the TP (true positives), TN (true negatives), FP (false positives) and FN (false negatives) into account and gives a scalar value which presents the evaluation of the model. In a multi-label scenario, one needs to use the aggregated F-score which includes F-micro, F-macro and F-sample according to different types of aggregations used. The definition for F-score, F-micro, F-macro and F-samples are provided as follows:

$$F\text{-score} = \frac{(1+\beta^2)tp_s}{(1+\beta^2)tp_s + fp_s + \beta^2 fn_s} \tag{5}$$

$$F\text{-micro} = \frac{\sum_{c=1}^{C}(1+\beta^2)tp_c}{\sum_{c=1}^{C}(1+\beta^2)tp_c + fp_c + \beta^2 fn_c} \tag{6}$$

$$F\text{-macro} = \frac{1}{C}\sum_{c=1}^{C}\frac{(1+\beta^2)tp_c}{(1+\beta^2)tp_c + fp_c + \beta^2 fn_c} \tag{7}$$

$$F\text{-samples} = \frac{1}{S}\sum_{s=1}^{S}\frac{(1+\beta^2)tp_s}{(1+\beta^2)tp_s + fp_s + \beta^2 fn_s} \tag{8}$$

here C is the total number of different classes and S is the total number of samples. For our analysis we are setting $\beta = 1$ corresponding to the F1-score.

In this study, we use F-micro, F-macro, F-sample for evaluating multi-class classification, and the F-score for binary classes. F-scores presents a harmonic mean of precision and recall and β is a trade-off parameter (we use 1 in all our analyses) for false-negatives and false-positives. F-scores are commonly used in evaluating multi-label classification tasks.

In addition to the F-scores discussed above, we also use the Hamming loss. In general accuracy gives the percentage between correctly predicted labels and all the labels for binary problem, however under multi-label scenario each sample receive multiple labels, in this case the accuracy can be calculated either by using the exact match where all the labels from one sample have to match the true labels, or using fractional match where each label is compared to the true label individually, the later style is refereed as hamming accuracy, hamming loss is defined by $1-hammingaccuracy$. Hence, the Hamming loss can be viewed as an alternative accuracy measure for the multi-label scenario. In Eqn. 9, we show how to obtain the Hamming loss. Here N is the number of total samples and C is the number of classes, $\oplus$ denotes the XOR operation and $\hat{y}_j^{(i)}$, $y_j^{(i)}$ correspond to the predicted label j and true label j from sample i,

$$\text{Hamming Loss} = \frac{1}{NC}\sum_{i=1}^{N}\sum_{j=1}^{C}\hat{y}_j^{(i)} \oplus y_j^{(i)}. \tag{9}$$

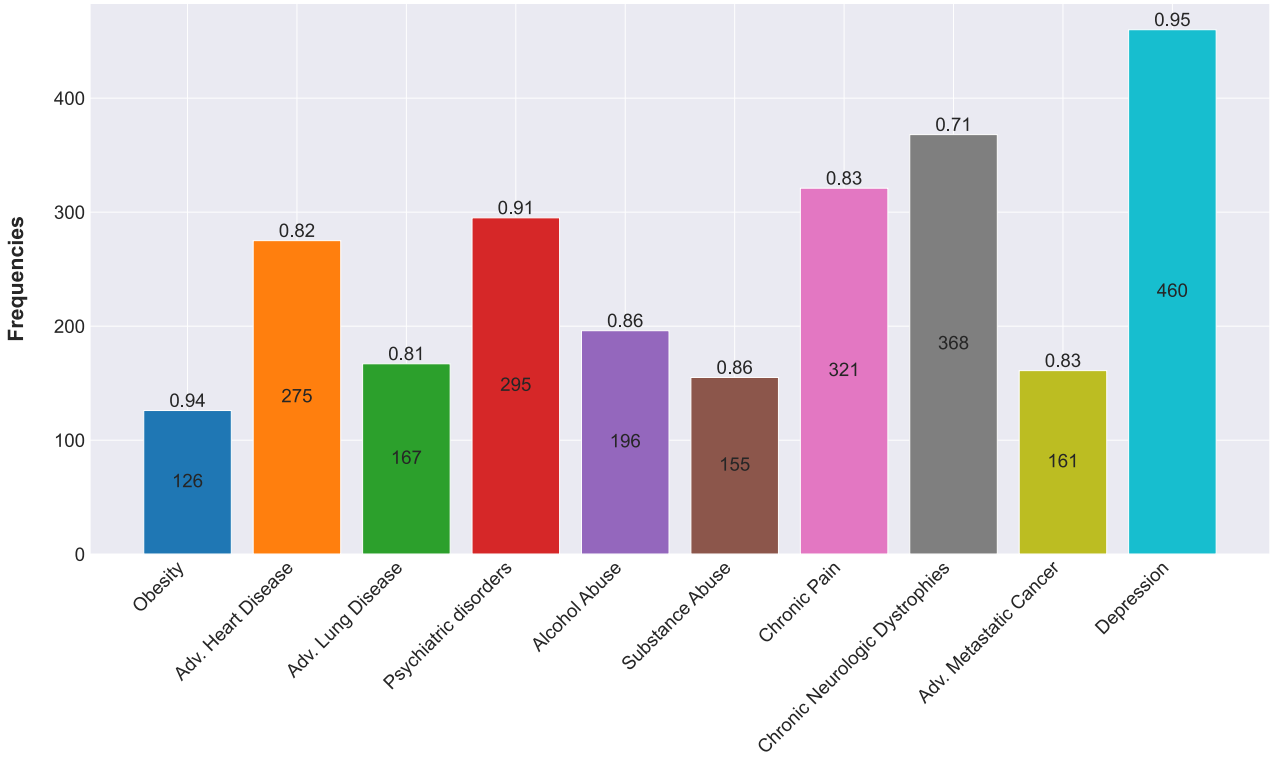

FIGURE 2. Overview of the used data from the MIMIC-III database. The numbers correspond to frequencies and Cohen's Kappa scores for the 10 phenotypes used in this study. Cohen's Kappa scores are shown on top of the bars while the y-axis corresponds to the frequencies (middle of the bars) of the phenotypes.

E. EXPERIMENTAL SETTING FOR THE MODELS

In this section, we describe practical aspects of our analysis. For our study, we use python 3.9 and the PyTorch [38] package to build our networks. For C-BERT we used the pre-trained network provided by [1]. For the numerical analyses, the data is split into three parts: 70% for training, 10% for validating and 20% for testing.

1) PREPROCESSING

We adopt the same processing procedure as in [50]. Raw texts were cleaned to remove stop words and symbols by using predefined rules. The cleaned input is converted into individual tokens, paddings are added to the tails of the input texts to ensure equal lengths for all the inputs. Overall, this results in 1610 samples with a total length of 5572 and a total vocabulary size of 48848. The embedding dimension is 50 learned with word2vec [31] using the all discharge summaries of MIMIC-III as corpus.

2) Binary-CNN

For training the binary CNN (B-CNN), we use convolutional filters of window of $[1\times1, 1\times2, 1\times3, 1\times4, 1\times5]$, each group of filter window has 100 different filters, forming 500 feature maps. We use softmax as the last activation function to a logit output of the network and cross-entropy as the loss function. For this, we train 10 different classifiers for 10 different classes, using one-vs-rest learning framework.

3) MULTI-LABEL CNN

For training the multi-label CNN (M-CNN), we use convolutional filters of window length equal to $[1\times1, 1\times3, 1\times4, 1\times5]$. For each filter window group we use 500 filters adding up to a total of 3000 feature maps. A sigmoid function is used as the last activation function and binary cross-entropy is used as the loss function. In the testing phase, a thresholding function with a constant value of 0.5 is applied to make a decision about the classes.

4) IMPLICIT-THRESHOLD CNN

For the Implicit-Threshold CNN (IT-CNN), we use the same setup as for the Multi-label CNN but we replace the thresholding function with a network that is capable of learning the number of positive classes for each sample. The structure is similar to [10], however, we employ a network with a two-layer MLP with 512 and 256 neurons respectively. The network takes the learned features as the inputs and outputs the number of positive classes for each sample. Losses from the two networks are aggregated and the two networks are trained simultaneously using one forward and backward pass. In the testing phase, the prediction of the additional network is used to select the top-k scores from the output of the original network to select the final classes.

5) ADAPTIVE-THRESHOLD CNN

Also the Adaptive-Threshold CNN (AT-CNN) inherits the same setup from the Multi-label CNN while the thresholding

function is replaced with learned adaptive thresholds. In addition to the prediction for all classes at the final layer from M-CNN, binary predictions for each class is also performed, and the prediction scores from the binary predictors are concatenated to form a vector that represents the numeric thresholds for each class. Each sample will learn its own threshold vector during training, and in the testing phase these thresholds are compared against the output scores from the last layer of the model to select the positive classes.

6) CLINICAL-BERT

For comparison with the above models we use Clinical-BERT (C-BERT) [1]. We use the variant "Bio-Discharge-Summary-BERT" which was pre-trained on all discharge summaries from MIMIC-III. The model was further fine-tuned on our dataset using a multi-label framework with a threshold of 0.5.

7) MAGNET

Another baseline model is MAGNET [36], which is LSTM-based, designed for the classification of multi-label tasks. We use the same model as in [36] using our setup and pre-trained word embeddings.

8) SGM

SGM [49] is another LSTM-based multi-label learning method we use for a baseline comparison. We use the model from [49] provided on github using our setup and the same pre-trained word embeddings. The results are obtained without global embeddings.

9) LACO

LACO [56] is a recently introduced multi-label learning method based on a BERT architecture. LACO also implicitly utilizes label correlation information to enhance the performance on multi-label learning tasks and outperforms many strong baseline methods by a large margin on several benchmark datasets. We use Bio-Clinical-Bert as the pre-trained BERT model for LACO. Furthermore, we apply a similar preprocessing procedure on the data as for all other methods.

IV. RESULTS

In the following, we present the results of our analysis. First, we study the optimization of the CNN models. Then we compare the performance of different methods with each other for the 10 class multi-label classification problem. Thereafter we study the influence of the difficulty level and the number of classes on the performance of multi-label classification. Finally, we investigate the time complexity of the best performing classifiers.

A. OPTIMIZATION OF CNNs

In this section, we discuss the process of optimizing our CNN-based methods. Specifically we investigate how different parameters affect the performances of all the CNN-based methods.

In Figure 3, we show the performances of B-CNN, AT-CNN, IT-CNN and M-CNN in dependence on the number of filters used per window size (see Figure 3 column (a)) and in dependence on the filter window sizes (see Figure 3 column (b)). For the number of filters, we start at 100 and increase the size at each step by 100 up to a size of 600. For the filter window sizes, we study six setting given by [1×1, 1×2, 1×3, 1×4, 1×5].

As one can see in the Fig. 3, B-CNN is a very stable regardless of the parameter settings and shows only a decreased performance for too small filter window sizes. In contrast, all of the multi-label learning models are strongly influenced by a low number of filter maps and small filter window sizes. Furthermore, when the number of filter maps and filter window sizes are too large, the performance starts to decay indicating an overfitting effect.

Based on this and other analyses (not shown), we select the optimal parameters for each model for the following analysis. That means all models used in the following, have been optimized with respect to the parameters of the models. We want to highlight that this includes the baseline model B-CNN which is the only model that transforms the multi-label problem into multiple binary classifications.

B. MODEL COMPARISON

In this section, we study a 10-class multi-label classification problem and compare the performance of 8 models: Binary CNN (denoted B-CNN), Multi-label CNN (denoted M-CNN), Implicit-Threshold Multi-label CNN (denoted IT-CNN), Adaptive-Threshold Multi-label CNN (denoted AT-CNN), Clinical-BERT (denoted C-BERT)) MAGNET, SGM and LACO. The data for this analysis are from the MIMIC-III database where the 10 classes correspond to 10 disease phenotypes of patients. Among all models, the B-CNN is the only classifier utilizing a binary learning framework by learning individual classifiers as one-vs-rest, whereas the other 7 architectures are based on a genuine multi-label learning framework. As performance measures, we use F-micro, F-macro, F-sample and the Hamming Loss. For estimating these scores and the standard errors we use a 10-fold cross validation (CV).

The results of this analysis are shown in Table 1. From the table one can see that the B-CNN has the top scores for all three F-score types, but for the Hamming loss the AT-CNN is best. Interestingly, C-BERT, MAGNET and SGM have the worst F-score performances. A reason why the two LSTM-based models (MAGNET and SGM) do not perform well could be the small sample size of our data because it is known that LSTM models have difficulties fitting such data. In contrast, CNN-based methods can efficiently learn the most important combinations of phrases that contribute to a certain phenotype. For the threshold multi-label CNNs, the AT-CNN outperforms IT-CNN on the F-micro and F-macro score by 3.3% and 2.2% respectively. For the Hamming loss the difference is even larger corresponding to 18%.

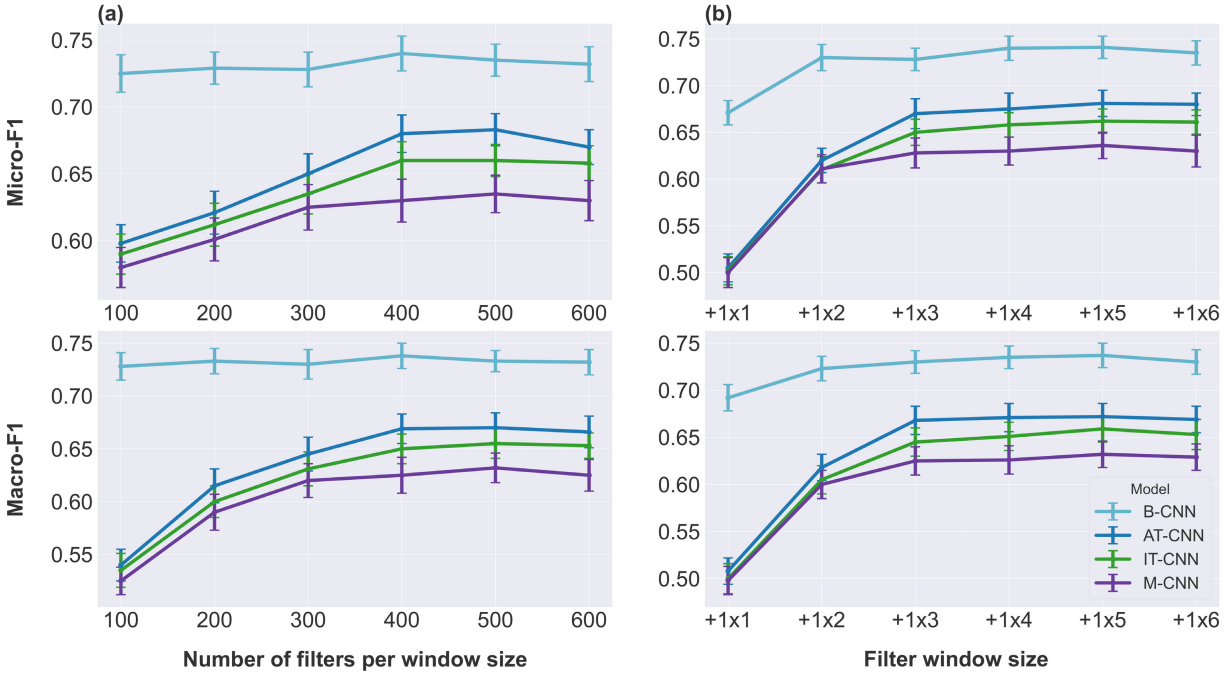

FIGURE 3. Optimization of the parameters for B-CNN, AT-CNN, IT-CNN and M-CNN. Column (a) shows the impact of the number of filters used per window size, starting from 100 with an increment of 100, to 600. Column (b) shows the impact of filter window size on the performance, starting from window size of 1 × 1 then adding larger window sizes.

Interestingly, F-sample of the IT-CNN is almost as good as for the B-CNN.

Due to the fact that C-BERT, MAGNET and SGM are the worst performing models, we do not consider these in the following because they add nothing to our analysis. Interestingly, despite the fact that LACO uses contextualized word embeddings and an enhanced multi-label learning framework, it shows a very similar performance to M-CNN which means it under-performs. For this reason, we also do not consider LACO in the following analysis.

In Fig. 4, we show more detailed results by providing the classification scores for the 10 underlying phenotypes corresponding to the 10 classes of the classification task. From this figure it is interesting to note that while the B-CNN (thick line) has the best overall score (see Table 1), for the class ''Cancer'' and ''Depression'' this is not the case. Instead, AT-CNN performs better. Furthermore, it is interesting to remark that the most difficult class (having the lowest F-score) for B-CNN and AT-CNN is ''Pain'' while for ''Lung'' we observe the largest difference between the multi-label learning classifiers (AT-CNN, IT-CNN and M-CNN) and B-CNN.

In order to highlight the differences between the B-CNN and the best other classifier, we added bar plots to Fig. 4. These bars correspond to ΔF between the best performing multi-label classifier (AT-CNN, IT-CNN and M-CNN) and B-CNN for the corresponding phenotype. Importantly, the color of the bar indicates the best performing classifier. While a positive score indicates a better performance of B-CNN a negative score indicates a better performance of the corresponding multi-label learning classifier. As one can see, AT-CNN is in 7 out of the 10 cases the best multi-label classifier and in the remaining three cases IT-CNN performs best. In summary, this demonstrated that both AT-CNN and IT-CNN outperform M-CNN.

C. DIFFICULTY LEVELS OF CLASSIFICATION

In order to further explore the capabilities of the multi-label classifiers against the B-CNN, we study these models using simulated data by varying the difficulty level of the classification. In order to do this, we generate simulated datasets by randomly flipping a certain portion of labels. In this way we change the percentage of positive and negative labels, which means we are essentially adding noise into the data. This allows us to control the difficulty level of the classification problem by using simulated data based on the original data. Using this procedure, we generate 5 different simulated datasets corresponding to 5 different percentages of label flips. The percentages range from 90% to 50% with a 10% interval corresponding to 5 different difficulty levels.

The results of this analysis can be seen in Fig. 5. It is interesting to note that for all randomizations, AT-CNN and IT-CNN outperform again M-CNN. Furthermore, the distance between AT-CNN, IT-CNN and the B-CNN is in general small but does not change much with increasing noise levels. Interestingly, there are cases where IT-CNN has slightly higher F-sample values than B-CNN; (figure on the right-hand-side in Fig. 5). Furthermore, we notice that the

TABLE 1. Results for B-CNN, AT-CNN, IT-CNN, M-CNN, C-BERT, MAGNET, SGM and LACO for the Hamming loss, F-micro, F-macro and F-sample (standard error in brackets). The results in bold highlight the best performance for each error measure. A "+" indicates that the higher the score the better whereas a "−" indicates the lower the score the better.

Model	Hamming loss (-)	F-micro (+)	F-macro (+)	F-sample (+)
B-CNN	0.092 ±0.006	**0.725** ±0.014	**0.728** ±0.013	**0.535** ±0.012
AT-CNN	**0.090** ±0.005	0.681 ±0.014	0.672 ±0.014	0.486 ±0.013
IT-CNN	0.109 ±0.006	0.662 ±0.013	0.659 ±0.014	0.530 ±0.013
M-CNN	0.114 ±0.008	0.636 ±0.014	0.632 ±0.014	0.460 ±0.015
C-BERT	0.112 ±0.007	0.618 ±0.010	0.615 ±0.010	0.454 ±0.013
MAGNET	0.130 ±0.010	0.602 ±0.011	0.605 ±0.014	0.445 ±0.010
SGM	0.119 ±0.009	0.629 ±0.012	0.631 ±0.013	0.480 ±0.012
LACO	0.111 ±0.008	0.635 ±0.010	0.645 ±0.012	0.438 ±0.015

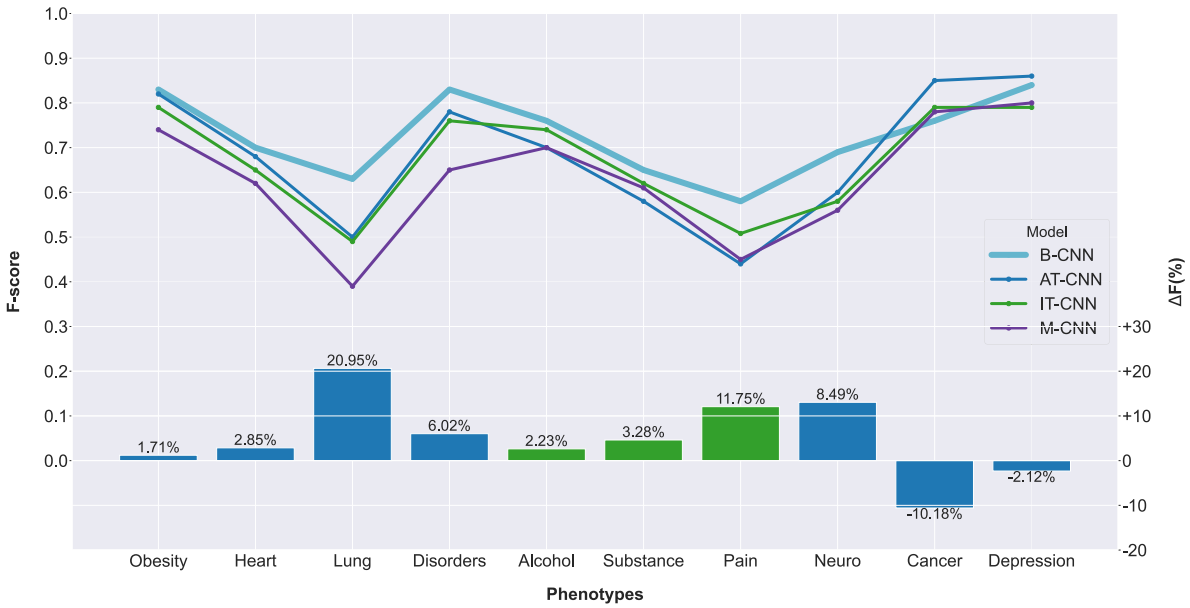

FIGURE 4. Results for B-CNN (thick line), AT-CNN, IT-CNN, and M-CNN for individual classes corresponding to ten phenotypes. The bar plots indicate ΔF between B-CNN and the best performing other classifier (see color for the classifier and y-axis on the right).

standard error increases with increasing percentages of the label flips (left to right). This is reasonable, since the noise level increases form left to right.

Due to the fact that the AT-CNN and IT-CNN outperform always the M-CNN, we study in the following only the AT-CNN, IT-CNN and the B-CNN.

D. SUBCLASS CLASSIFICATIONS

Next, we study the influence of the number of classes on the classification performance. Specifically, for our dataset we select all combinations of classes for the 10 phenotypes and perform a classification analysis for each. In total this allows to generate

$$\sum_{n=2}^{10} \binom{10}{n} = 1013 \tag{10}$$

different datasets whereas each binomial coefficient $\binom{10}{n}$ gives the number of subclasses that can be formed drawing n classes from 10. For each of these 1013 combinations we

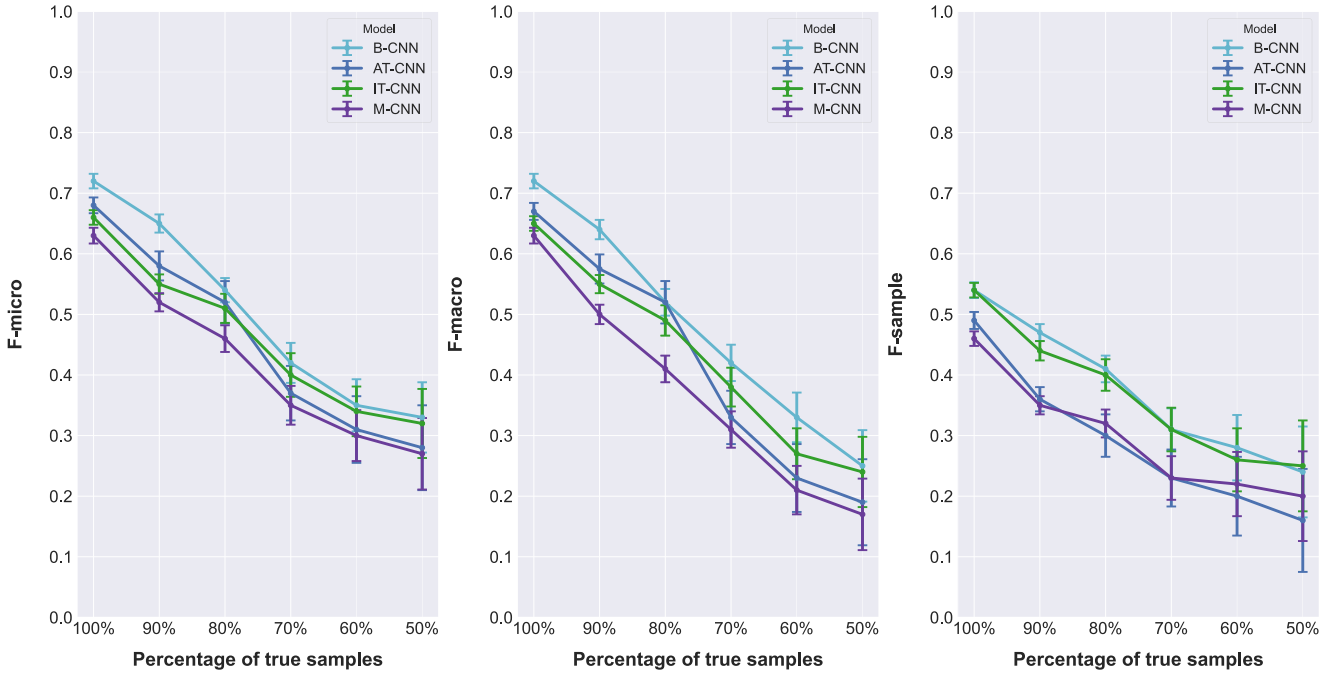

FIGURE 5. Performance of the classification models trained with simulated data using label-randomizations of 90%, 80%, 70%, 60%, and 50%. The performance is measured using F-micro (left), F-macro (middle) and F-sample (right).

perform an analysis similar to the previous sections, i.e., using 10-fold CV for estimating various error measures.

In Fig. 6 (a), we show the results for B-CNN, AT-CNN and IT-CNN for 1013 different datasets. These models correspond to the different subclasses one can form ranging from 10 to 2 classes (from left to right). Each of these corresponds to a n-class classification problem. The y-axis in Fig. 6 (a) indicates the F-micro scores, while the x-axis shows the different models trained on different combinations of the classes. The results are ordered by the scores of the AT-CNN per number of subclasses from highest to lowest values and the results for the B-CNN and IT-CNN are plotted using the same order. This explains the smoothly decaying lines for AT-CNN while the results for B-CNN and IT-CNN are jagged.

One can see that the B-CNN outperforms the AT-CNN and IT-CNN for almost all models. However, there are two interesting observations. First, the difference between those classifiers is usually small and in the percentage range. Second, the B-CNN is not always the best model. In order to quantify this observation we show in Fig. 6 (b) the (relative) difference between the performances of the B-CNN and AT-CNN and IT-CNN. That means for each model

$$\Delta F\text{-micro} = \frac{F\text{-micro(B-CNN)} - F\text{-micro(CNN)}}{max\Big(\{F\text{-micro(B-CNN)}, F\text{-micro(CNN)}\}\Big)} \tag{11}$$

is calculated and the results are sorted from lowest to highest score. Here a negative score indicates a better performance of AT-CNN or IT-CNN, while a positive score shows the B-CNN is better. From Fig. 6 (b), one can see that there are 23 models for which the IT-CNN and 198 models for the AT-CNN outperforms the B-CNN.

In Fig. 6 (c) and (d) we show the results of a similar analysis for the F-macro score. Overall, the results are very similar to the F-micro score in Fig. 6 (a) and (b) confirming the above observations.

In order to compare the results between the subclasses, we average the scores for each subclass. These results are shown in Fig. 7. Here the bars correspond to the mean value of the corresponding F-scores in Fig. 6 and the error bars are the standard error. The standard error increases again for a decreasing number of classes (left to right), similar to the results in Fig. 6. Furthermore, also the values of the mean F-scores seems to increase for AT-CNN and IT-CNN from 9 to 2 classes but not the B-CNN. To confirm this, we perform a one-dimensional linear regression by considering the scores for each of the 8 categories as a samples. As a result we obtain the following slopes of linear regression models with corresponding p-values (see Eqn. 12 to 17, as shown at the bottom of the next page).

From these p-values, we can see that the slopes for the AT-CNN and IT-CNN are significant for a significance level of $\alpha = 0.05$ while the slopes for the B-CNN are not significant. This analysis confirms our qualitative observations.

We would like to remark that the number of classes can also be seen as difficulty level of a classification. However, this difficulty level is different to the one studied in the previous section where we introduced essentially noise into the data by flipping labels. In contrast, in this section no noise as

such was introduced but the number of classes was varied which increases the variability of the data for a decreasing number of classes because the number of combinations increases.

E. TIME COMPLEXITY

Finally, we compare the efficiency of the B-CNN with the modified CNN models by studying their runtime. We do this not only for the original 10 classification problem but also for fewer classes to see if the number of classes has an influence on this.

In order to study problems with less than 10 classes we generate again new datasets by randomly selecting n classes from the 10 available phenotypes. For a given n with $2 < n \leq 10$ this results in $\binom{10}{n}$ different datasets. In total this gives

$$\sum_{n=2}^{10} \binom{10}{n} = 1013 \tag{18}$$

different datasets each corresponding to a n-class classification problem.

The results for these datasets is shown in Fig. 8. Here the x-axis shows all 1013 models from 10 to 2 classes and the y-axis shows the corresponding runtime of the classifier models in seconds.

Overall, one can see that the runtimes of the AT-CNN are essentially the same regardless of the number of classes. The results for the IT-CNN are almost identical to the AT-CNN, for this reason we did not add them to Fig. 8. In contrast, the B-CNN becomes slower with an increasing numbers of classes. This behavior is reasonable since the binary CNN performs more and more ''individual'' binary classifications the larger the number of classes whereas for the AT-CNN this remains constant. Importantly, for a 10-class classifier (right-hand-side in Fig. 8) the difference between both classifiers is more than a factor of 10. A linear regression analysis for the runtimes of the B-CNN shows that the growth is linear with a slope of $\beta = 200.22$ and a p-value of $p = 2e-16$.

Overall, these results demonstrate the need for finding a substitute for a B-CNN because when we have a verly large number of classes the conversion of a multi-label classification problem into a binary multi-class classification problems is very inefficient and becomes even prohibitive in the limit.

V. DISCUSSION

In general, strategies for multi-label classification can be divided into two categories. The first transforms the multi-label classification problem into multiple binary classifications resulting in many ''one-vs-rest'' comparisons whereas the second predicts multiple labels simultaneously. Hence, the latter approaches are genuine multi-label classifiers in the sense that they require a new learning paradigm [12]. A very popular example of transformation-based approaches is binary relevance, e.g., B-CNN whereas threshold-based learners, e.g., M-CNN, AT-CNN or IT-CNN are examples for the second category. It is important to note that the first category does usually not use label dependency information at all, while the second category utilize this information either explicitly or implicitly in make the prediction of the labels.

In order to obtain an informed assessment of our findings, first, we discuss and summarize performance differences between binary relevance methods and methods based on multi-label learning from the literature. Such a comparison can be divided into two categories by distinguishing the used methodology for the binary relevance method: (I) BR based on traditional methods and (II) BR based on deep learning methods.

In Fig. 9, we show a summary of these results reported in the literature. Specifically, the forest plot shows Δ Score values between a proposed method and a BR model for various studies. Specifically, the x-axis corresponds to Δ Score values where a positive value indicates a better performance of the proposed multi-label classifier and a negative value means a better performance of the BR baseline model. The multi-label classifiers used in these studies are: CNN-RNN [5], DSRM-DNN [45], SGM [49], MAGNET by [36], ML-NET [10], EncDec [35], LACO [56] and JBNN [16].

An example for studying a binary relevance method based on traditional machine learning methods is from [5]. They used a BR model based on linearSVM as base classifier and compared it with a CNN-RNN architecture. For the dataset Reuters-21578 they found the CNN-RNN is inferior to the BR model by 16% respectively 2.6% for F-macro and F-micro. In contrast, for the dataset RCV1-v2, the BR model was slightly worse by 0.4% for F-micro and for F-macro it was 3.5% worse. They argued that the size of the dataset greatly influences the performance of their model which is the

$$\beta_{F_{micro}(\text{B-CNN})} = -0.0004098, \quad \beta_{F_{macro}(\text{B-CNN})} = 0 \tag{12}$$
$$\beta_{F_{micro}(\text{AT-CNN})} = -0.0081840, \quad \beta_{F_{macro}(\text{AT-CNN})} = -0.010379 \tag{13}$$
$$\beta_{F_{micro}(\text{IT-CNN})} = -0.0068947, \quad \beta_{F_{macro}(\text{IT-CNN})} = -0.007499 \tag{14}$$
$$\mathrm{P}_{F_{micro}(\text{B-CNN})} = 0.395, \quad \mathrm{P}_{F_{macro}(\text{B-CNN})} = 1 \tag{15}$$
$$\mathrm{P}_{F_{micro}(\text{AT-CNN})} = 7.66e-07, \quad \mathrm{P}_{F_{macro}(\text{AT-CNN})} = 3.30e-07 \tag{16}$$
$$\mathrm{P}_{F_{micro}(\text{IT-CNN})} = 0.000195, \quad \mathrm{P}_{F_{macro}(\text{IT-CNN})} = 3.29e-05 \tag{17}$$

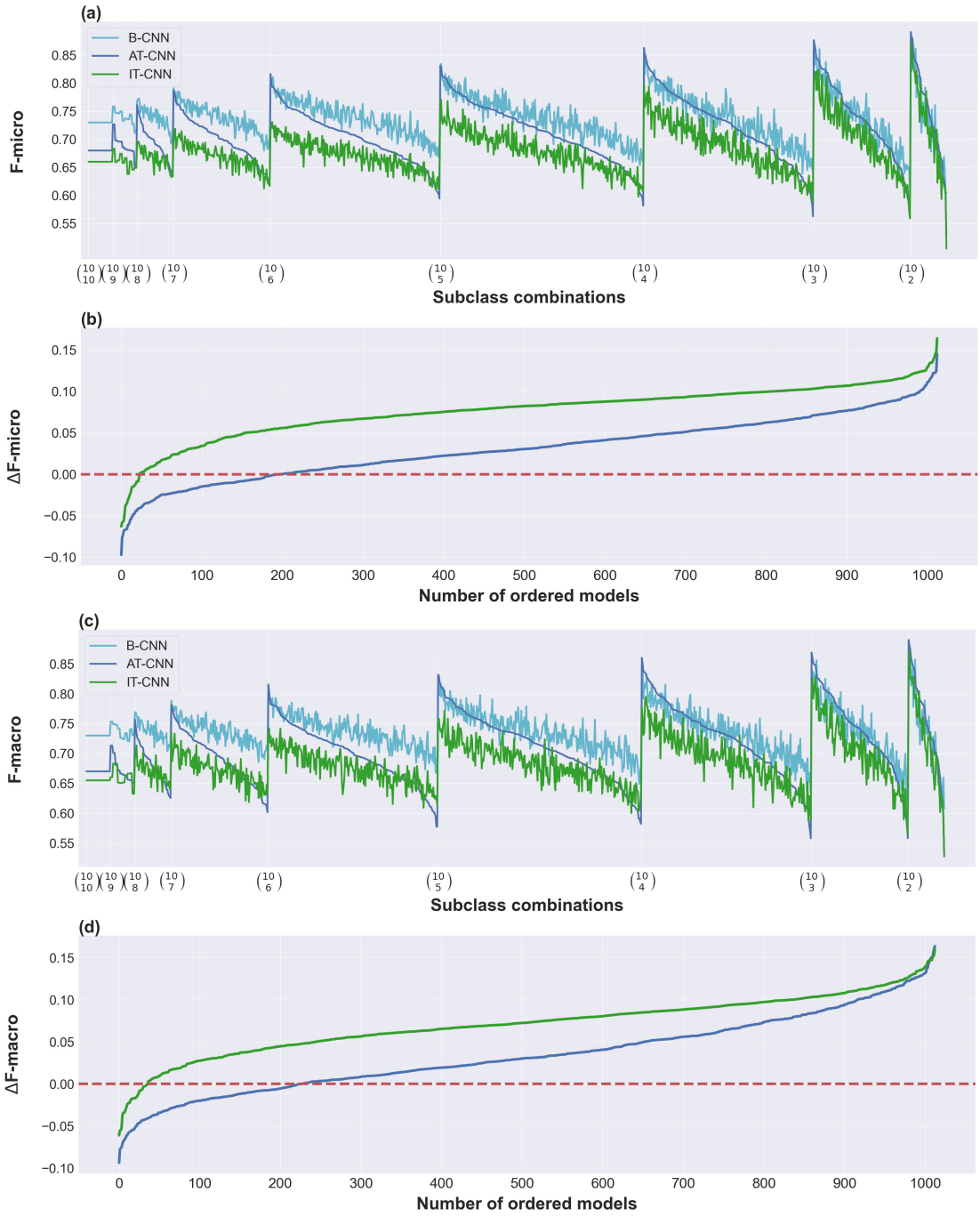

FIGURE 6. Results for different subclass classifications. (a) : Performance measured by F-micro for B-CNN, AT-CNN and IT-CNN. (b): Shown is ΔF-micro for the difference between B-CNN and AT-CNN and IT-CNN. (c): Performance measured by F-macro for B-CNN, AT-CNN and IT-CNN. (d): Shown is ΔF-macro for the difference between B-CNN and AT-CNN and IT-CNN.

reason why the BR model surpasses their CNN-RNN model for rather small datasets. Du et al. [10] proposed ML-Net which utilizes a document embedding and an additional label-decision module. They compared their architecture also to a SVM-based BR method for 3 different medical classification tasks: Hallmarks of cancer classification, Chemical exposure assessments and Diagnosis code assignment. The results from all three tasks showed that ML-Net outperforms the BR model for a F-sample score. Specifically, for the Hallmarks of cancer (1580 PubMed abstracts with 10 classes) they found an improvement of 13.8% over the BR, for the Chemical exposure assessments (3661 PubMed abstracts with 32 classes) there is a 3.8% improvement over BR, and for the Diagnosis code assignment (22,815 samples with 7,042 classes from MIMIC-III) there is a 7.7% better performance. Interestingly, the BR model is superior for the latter two datasets, which have many classes (>30) and large sample sizes (>3000), when precision is used as a score.

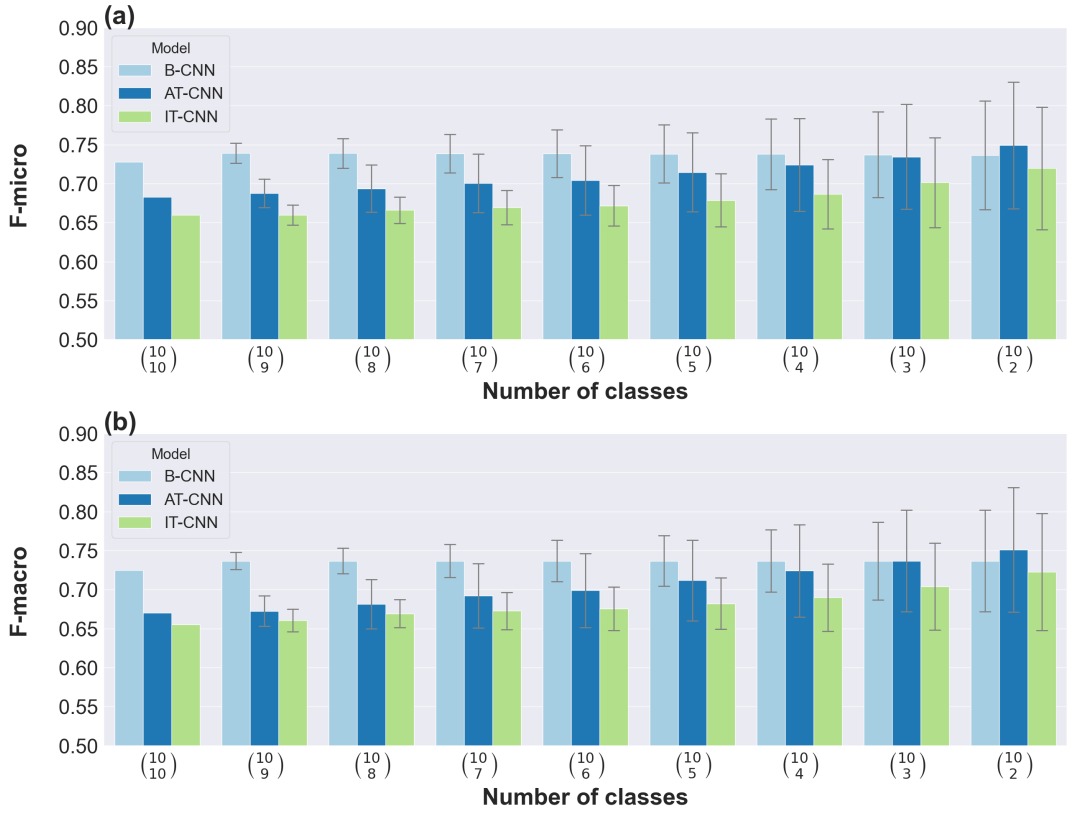

FIGURE 7. Mean F-micro and F-macro for the B-CNN, AT-CNN and IT-CNN for the results shown in Fig. 6. The error bars correspond to the standard error.

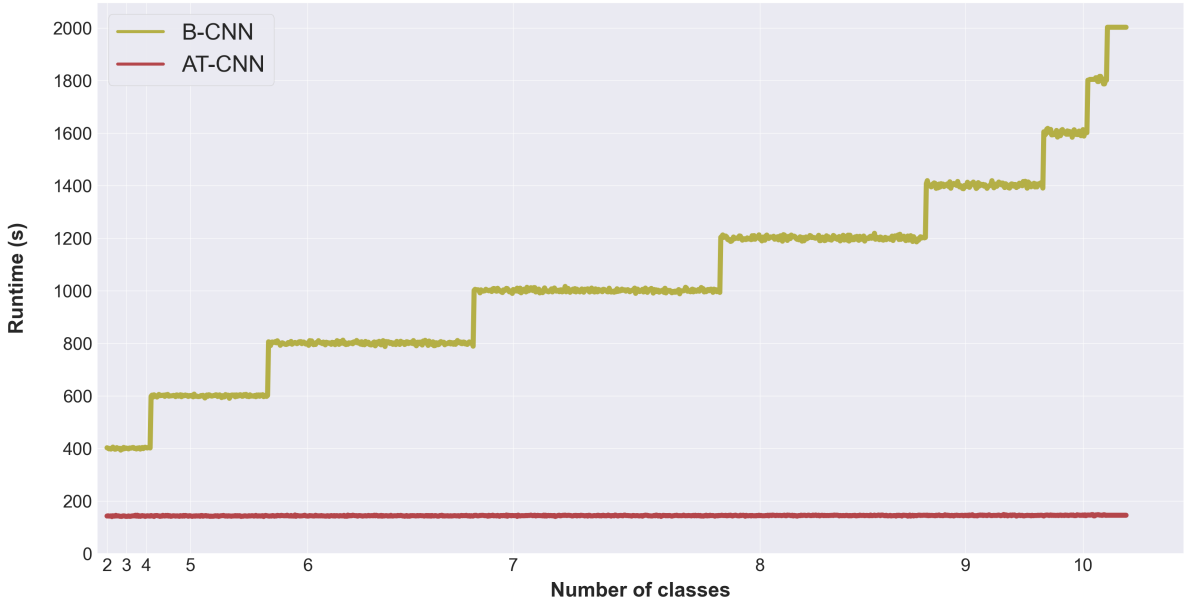

FIGURE 8. Time complexity of the classification models. Shown are the running times for the B-CNN and AT-CNN. The x-axis shows the number of classes and the y-axis the runtime in seconds.

The study by Wang et al. [45] compared their model called DSRM-DNN (dynamic semantic representation model and deep neural network) with a BR model based on a SVM. The most significant improvement of DSRM-DNN

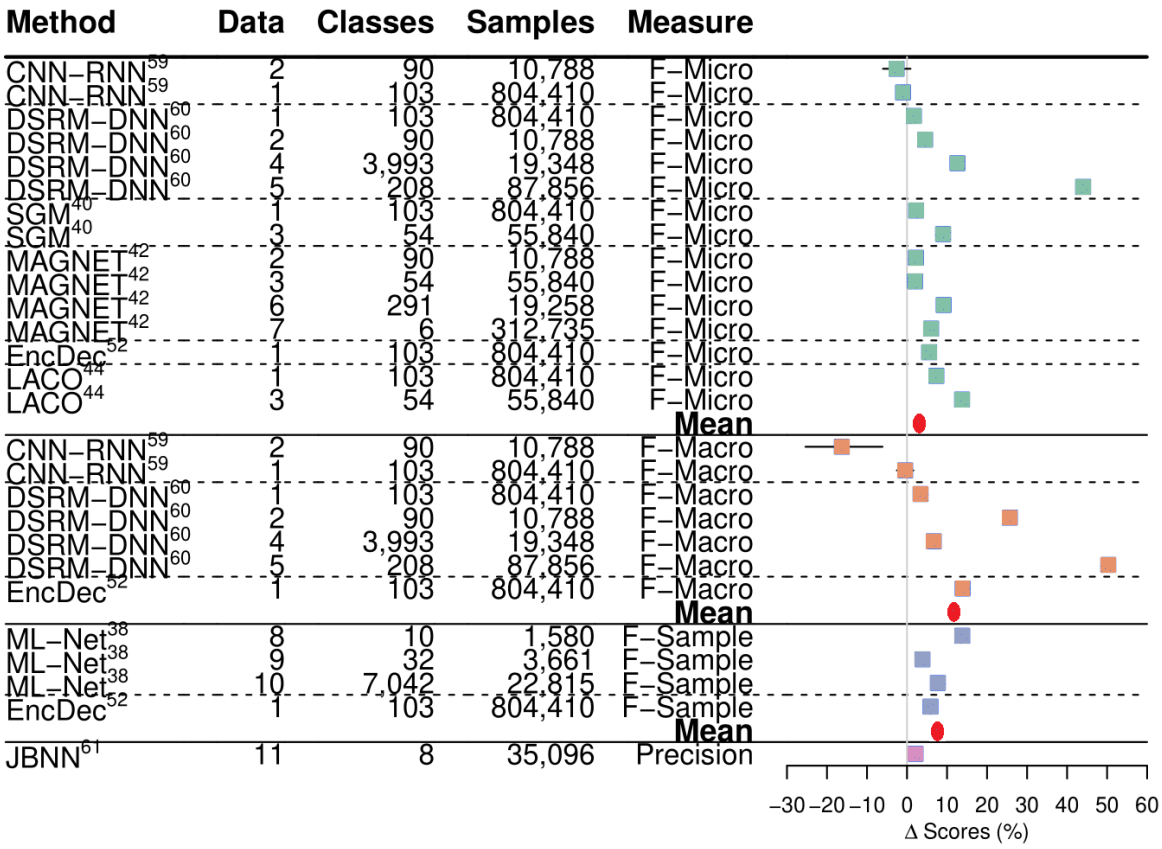

FIGURE 9. Forest plot showing ΔF score between a proposed method and binary relevance from various studies. The x-axis corresponds to Δ Score whereas a positive value indicates a better performance of the proposed method and a negative value means better performance of the BR baseline. The color indicates different measures (for F-Micro, F-Macro and F-Sample), and the index number from the 'Data' column indicates the used dataset: (1) RCV1-v2, (2) Reuters-21578, (3) AAPD, (4) EUR-LEX, (5) Bookmarks, (6) Slashdot, (7) Toxic, (8) Hallmarks of cancer, (9) Chemical exposure, (10) Diagnosis code assignments, (11) Ren-CECps.

over BR they found for the Reuters-21578 and EUR-Lex data giving 25% and 15% improvement for the F-micro and F-macro respectively. Yang et al. [49] used their sequence generation model SGM for solving multi-label text classification problems. They tested their method for data from RCV1-V2 and AAPD, and showed that the best performing method is SGM with 2.3% and 9.0% improvement over traditional BR for F-micro on RCV1-V2 and AAPD respectively. Pal et al. introduced a graph based attention deep learning architecture MAGNET [36] to explicitly incorporate label dependency information. They reported the performance of F-micro for their method against a traditional BR, showing 2.3%, 2.0%, 10.0%, and 6.4% improvement for F-micro on data from Reuters-21578, AAPD, Slashdot and Toxic respectively. Finally, Zhang et al. [56] proposed the method LACO which utilizes BERT and label dependency information. Their method showed the most competitive performance with respect to F-score against other deep learning architectures and a traditional BR. The results showed that LACO has 13.8% and 3.0% performance gain for F-micro on AAPD and RCV1-v2 respectively.

Overall, from such studies one can see that a BR model based on traditional machine learning techniques is still competitive even in comparison with modern deep learning multi-label based architectures. However, it should be emphasized that such a comparison is not the best choice and below we will return to this issue in more detail.

It is important to highlight that there are limited publications that offer comparisons for deep learning-based Binary Relevance (BR) models. However, an example is [35] where the performance of feedforward neural network-based binary relevance, feedforward neural network label-powerset and vanilla multi-label learning neural network [34] with RNN and encoder-decoder (called EncDec) is compared for data from Reuters-21578 and RCV1-v2. The results for the Reuters-21578 data demonstrate a 2.6% and 11.9% better performance for F-micro and F-macro respectively using an encoder-decoder against BR, while for RCV1-v2, the encoder-decoder is capable of improving over BR on F-micro and F-macro by 5.5% and 13.9% respectively. He et al. proposed a joint learning architecture to tackle multi-label learning as binary learning in one network structure in which

the network does not need to pre-define an extra thresholding function but can utilize label-dependency information [16]. They compared their architecture (called JBNN) using a binary transformed version and a multi-label learning version with the same neural network infrastructure for an emotion classification task using the Ren-CECps corpus. As a result, they found 6.26%,2.5%,4.9%,4.8% and 2.1% improvement over BR on the ranking loss, Hamming loss, one-error, coverage and average precision respectively. Unfortunately, from the provided description given in the proceeding papers, it remains unclear if the results obtained by the NN-based BR model from these studies are optimized to obtain the best possible performance.

When comparing the results from the literature, as summarized in Fig. 9, with our findings, we notice some differences. While also we observe cases where genuine multi-label classifiers (e.g. AT-CNN or IT-CNN) are better than a BR model, in general, the BR model is better. However, the difference between the BR model and AT-CNN or IT-CNN is usually within a few percentages (see Table 1, Fig. 4 and Fig. 6). In order to sheet light on this, we use available methods from the literature (discussed above) that reported a better performance compared to (their) BR models, and included those in our analysis. Specifically, we used MAGNET [36], SGM [49] and LACO [56]. Interestingly, from our analysis, we find that MAGNET, SGM and LACO are not only worse than our BR model but also our multi-label classifiers, i.e., AT-CNN and IT-CNN (see Table 1). While LACO is better than MAGNET and SGM, its performance is only comparable to M-CNN which is still worse than AT-CNN and IT-CNN.

From these results follow two observations. First, the results from the literature need to be interpreted with care because the quality of the conducted research cannot be verified. Second, even when the literature results are correct, a qualitatively different dataset can lead to different results, as demonstrated by our analysis (see Table 1). Overall, this places our findings and the results from the literature into perspective and underlines the following: (I) The analysis of our datasets is difficulty - because otherwise MAGNET, SGM and LACO would perform better. (II) The methods AT-CNN and IT-CNN are performing well for our datasets and in relation to the BR model. (III) AT-CNN and IT-CNN are performing especially well compared to MAGNET, SGM and LACO.

Methodologically, we would like to point out that our study introduced two new methods by augmenting a CNN with threshold functions. The reason for studying the influence of such threshold functions is that models utilizing a constant threshold, such as M-CNN (also studied in this paper), suffer from of lack of flexibility that usually translates into a poorer performance. Importantly, by a comparison of models with learned threshold functions, i.e., between adaptive-threshold CNN (AT-CNN) or implicit-threshold CNN (IT-CNN) and M-CNN, we could show that the performance improves greatly because the learned thresholds can adopt to characteristics from the data which influences the decision of a prediction.

There is another point worth highlighting and that is the use of label-dependency information. While the three models MAGNET, SGM and LACO explicitly rely on such an information, our methods, AT-CNN and IT-CNN, do not utilize such an information in an explicit way. In fact, we intentionally refrained from using this information to optimize a multi-label classifier that extracts as much information as possible from a dataset without label-dependency information. Although, when designing the thresholding function we use the information from labels to form the thresholds, the way such information is utilized is considerably different as in MAGNET, SGM or LACO. Considering this fact, it is even more remarkable that both AT-CNN and IT-CNN outperform MAGNET, SGM and LACO. If and how label-dependency information could be fully utilized to improve our deep learning models is an interesting question that requires a thorough analysis. For reasons of clarity, we want to add that also a base CNN multi-label structure, as used in our study, can still exploit label-dependency information to some extend, however, in an implicit manner. Typically, this happens in higher layers of the networks in a self-supervised fashion, i.e., it is not enforced from outside.

Regarding BR models, it is important to realize that there is no unique choice for such a model but there are alternatives. Specifically, for our analysis we tested different BR models and as a result we found a CNN-based binary relevance model to be most competitive. That means, we selected the best BR model among a set of alternative candidates. In contrast, in the literature the baseline BR models are either using SVMs or basic feed-forward neural networks which are in most cases too simple to deal with complex multi-label classification tasks. Furthermore, from the literature it is unclear if the decision for a particular BR model has been made based on a comparative analysis with alternative BR models or if the selection has been made imprudently. The latter includes choices where the optimal BR model has not been chosen. Hence, considering potential limitations of the used BR models when looking at the literature results in Fig. 9 one obtains a different view regarding the interpretation of the observed margins which are consistent with our interpretations above.

Overall, it is interesting to observe that despite the fact that advanced variants of deep neural networks are showing promising results for multi-label classification, BR models are not obsolete. This is especially true when the number of classes is moderate or small as in our study. Given the linear increase in runtime over the number of classes of a BR model, as shown in Fig. 8, the number of classes can even be surprisingly large, scaling with the size of the available computer cluster. Interestingly, we noticed a lack in the literature studying such cases, i.e., differences between deep learning based BR models and genuine multi-label classifiers for classification problems with a moderate number of classes. Based on our findings, we think that an optimized BR model might

be able to consistently compete with a genuine multi-label classifier, when no explicit label-dependency information is used. Unfortunately, this can only be studied for small to a medium number of classes because a computational analysis for a large number of classes becomes computationally prohibitive. However, maybe the more interesting question is the following: Can a multi-label classifier with label-dependency information systematically outperform an optimized BR model for classification tasks with a moderate number of classes? As discussed above, here it is very important to emphasize "optimized" because selecting any BR model does not lead to a fair comparison and a definite answer to this question.

VI. CONCLUSION

In this paper, we study multi-label classifiers for text classification of electronic health records (eHR). We compare deep learning based models for binary relevance (BR) and genuine multi-label models, including two newly introduced models AT-CNN and IT-CNN. As main results, we find the following: First, a CNN-based BR model is overall best, however, genuine multi-label methods using learned thresholds, i.e., adaptive-threshold CNN (AT-CNN) and implicit-threshold CNN (IT-CNN) are good approximations of the prediction performance. Second, AT-CNN and IT-CNN perform better than previously published methods on our dataset, e.g., Clinical-BERT [1], MAGNET [36], SGM [49] and LACO [56]. Third, the obtained results are robust for a varying number of classes and various noise levels. Forth, the runtime of AT-CNN and IT-CNN is much more efficient compared to the BR model and the speed advantage for a 10 class classification task is over a factor of 10. Fifth, when comparing BR models to genuine multi-label classifiers it is important to optimize also the BR model because its choice is not unique and many alternatives are possible. Otherwise the comparison becomes unfair and non-representative. In general, we suggest to use the same machine learning or artificial intelligence model type for the BR model and the genuine multi-label classifier when comparing their performance.

Overall, our findings are not only of methodological interest, because we show that deep learning based multi-label classifiers can benefit from well designed thresholding functions, but provide also guidelines for the general comparison between BR and multi-label models.

AUTHOR CONTRIBUTIONS STATEMENT

FES conseived the study, wrote the paper and interpreted the results. ZY performed the analysis, wrote the paper and interpreted the results.

CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests.

Data Availability: The datasets used and/or analyzed during the current study available from the corresponding author on reasonable request.

REFERENCES

[1] E. Alsentzer, J. R. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, and M. B. A. McDermott, "Publicly available clinical BERT embeddings," 2019, *arXiv:1904.03323*.

[2] T. Baumel, J. Nassour-Kassis, R. Cohen, M. Elhadad, and N. Elhadad, "Multi-label classification of patient notes: Case study on ICD code assignment," in *Proc. AAAI Joint Workshop Health Intell. (W3PHIAI)*, AAAI Press, 2018, pp. 409–416.

[3] M. R. Boland, N. P. Tatonetti, and G. Hripcsak, "Development and validation of a classification approach for extracting severity automatically from electronic health records," *J. Biomed. Semantics*, vol. 6, no. 1, pp. 1–13, Dec. 2015.

[4] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognit.*, vol. 37, no. 9, pp. 1757–1771, Sep. 2004.

[5] G. Chen, D. Ye, Z. Xing, J. Chen, and E. Cambria, "Ensemble application of convolutional and recurrent neural networks for multi-label text categorization," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2377–2383.

[6] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun, "Doctor AI: Predicting clinical events via recurrent neural networks," in *Proc. Mach. Learn. Healthcare Conf.*, 2016, pp. 301–318.

[7] A. Clare and R. D. King, "Knowledge discovery in multi-label phenotype data," in *Proc. Eur. Conf. Princ. Data Mining Knowl. Discovery*. Berlin, Germany: Springer, 2001, pp. 42–53.

[8] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, Apr. 1960.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[10] J. Du, Q. Chen, Y. Peng, Y. Xiang, C. Tao, and Z. Lu, "ML-Net: Multi-label classification of biomedical texts with deep neural networks," *J. Amer. Med. Inform. Assoc.*, vol. 26, no. 11, pp. 1279–1285, Nov. 2019.

[11] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 14, 2001.

[12] F. Emmert-Streib and M. Dehmer, "Taxonomy of machine learning paradigms: A data-centric perspective," *WIREs Data Mining Knowl. Discovery*, vol. 12, no. 5, p. e1470, Sep. 2022.

[13] F. Emmert-Streib, Z. Yang, H. Feng, S. Tripathi, and M. Dehmer, "An introductory review of deep learning for prediction models with big data," *Frontiers Artif. Intell.*, vol. 3, p. 4, Feb. 2020.

[14] F. Gargiulo, S. Silvestri, and M. Ciampi, "Deep convolution neural network for extreme multi-label text classification," in *Proc. 11th Int. Joint Conf. Biomed. Eng. Syst. Technol.*, 2018, pp. 641–650.

[15] S. Gehrmann, F. Dernoncourt, Y. Li, E. T. Carlson, J. T. Wu, J. Welt, J. Foote, E. T. Moseley, D. W. Grant, P. D. Tyler, and L. A. Celi, "Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives," *PLoS ONE*, vol. 13, no. 2, Feb. 2018, Art. no. e0192360.

[16] H. He and R. Xia, "Joint binary neural network for multi-label learning with applications to emotion classification," in *Proc. CCF Int. Conf. Natural Lang. Process. Chin. Comput.* Cham, Switzerland: Springer, 2018, pp. 250–259.

[17] A. N. Jagannatha and H. Yu, "Bidirectional RNN for medical event detection in electronic health records," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, p. 473.

[18] A. Jagannatha and H. Yu, "Structured prediction models for RNN based sequence labeling in clinical text," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, p. 856.

[19] B. Jang, M. Kim, G. Harerimana, S.-U. Kang, and J. W. Kim, "Bi-LSTM model to increase accuracy in text classification: Combining word2vec CNN and attention mechanism," *Appl. Sci.*, vol. 10, no. 17, p. 5841, Aug. 2020.

[20] K. Jasinska, K. Dembczynski, R. Busa-Fekete, K. Pfannschmidt, T. Klerx, and E. Hullermeier, "Extreme F-measure maximization using sparse probability estimates," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1435–1444.

[21] P. B. Jensen, L. J. Jensen, and S. Brunak, "Mining electronic health records: Towards better research applications and clinical care," *Nature Rev. Genet.*, vol. 13, no. 6, pp. 395–405, Jun. 2012.

[22] A. E. W. Johnson, T. J. Pollard, L. Shen, L.-W.-H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, "MIMIC-III, a freely accessible critical care database," *Sci. Data*, vol. 3, no. 1, pp. 1–9, May 2016.

[23] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1746–1751.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[25] M. Li, Z. Fei, M. Zeng, F.-X. Wu, Y. Li, Y. Pan, and J. Wang, "Automated ICD-9 coding via a deep learning approach," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 4, pp. 1193–1202, Jul. 2019.

[26] Y. Li, Y. Song, and J. Luo, "Improving pairwise ranking for multi-label image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1837–1845.

[27] A. Liede, R. K. Hernandez, M. Roth, G. Calkins, K. Larrabee, and L. Nicacio, "Validation of international classification of diseases coding for bone metastases in electronic health records using technology-enabled abstraction," *Clin. Epidemiol.*, pp. 441–448, Nov. 2015, doi: 10.2147/CLEP.S92209.

[28] J. Liu, W.-C. Chang, Y. Wu, and Y. Yang, "Deep learning for extreme multi-label text classification," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2017, pp. 115–124.

[29] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," 2016, *arXiv:1605.05101*.

[30] Q. Ma, C. Yuan, W. Zhou, and S. Hu, "Label-specific dual graph neural network for multi-label text classification," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics, 11th Int. Joint Conf. Natural Lang. Process.*, vol. 1, 2021, pp. 3855–3864.

[31] 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.*, vol. 26, 2013.

[32] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning-based text classification: A comprehensive review," *ACM Comput. Surv.*, vol. 54, no. 3, pp. 1–40, Apr. 2022.

[33] J. Mullenbach, S. Wiegreffe, J. Duke, J. Sun, and J. Eisenstein, "Explainable prediction of medical codes from clinical text," 2018, *arXiv:1802.05695*.

[34] J. Nam, J. Kim, E. L. Mencía, I. Gurevych, and J. Fürnkranz, "Large-scale multi-label text classification—Revisiting neural networks," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Berlin, Germany: Springer, 2014, pp. 437–452.

[35] J. Nam, E. L. Mencía, H. J. Kim, and J. Fürnkranz, "Maximizing subset accuracy with recurrent neural networks in multi-label classification," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, vol. 30, Nancy, France. Berlin, Germany: Springer, 2017, pp. 437–452..

[36] A. Pal, M. Selvakumar, and M. Sankarasubbu, "Multi-label text classification using attention-based graph neural network," 2020, *arXiv:2003.11644*.

[37] M. A. Parwez, M. Abulaish, and Jahiruddin, "Multi-label classification of microblogging texts using convolution neural network," *IEEE Access*, vol. 7, pp. 68678–68691, 2019.

[38] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[39] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1. New Orleans, LA, USA: Association for Computational Linguistics, Jun. 2018, pp. 2227–2237.

[40] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Mach. Learn.*, vol. 85, no. 3, pp. 333–359, Dec. 2011.

[41] R. F. Sarmiento and F. Dernoncourt, "Improving patient cohort identification using natural language processing," in *Secondary Analysis of Electronic Health Records*, 2016, pp. 405–417. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-43742-2_28

[42] M. J. Schuemie, E. Sen, G. W. Jong, E. M. van Soest, M. C. Sturkenboom, and J. A. Kors, "Automating classification of free-text electronic health records for epidemiological studies," *Pharmacoepidemiology Drug Saf.*, vol. 21, no. 6, pp. 651–658, Jun. 2012.

[43] H. Shi, P. Xie, Z. Hu, M. Zhang, and E. P. Xing, "Towards automated ICD coding using deep learning," 2017, *arXiv:1711.04075*.

[44] T. Vu, D. Q. Nguyen, and A. Nguyen, "A label attention model for ICD coding from clinical text," 2020, *arXiv:2007.06351*.

[45] T. Wang, L. Liu, N. Liu, H. Zhang, L. Zhang, and S. Feng, "A multi-label text classification method via dynamic semantic representation model and deep neural network," *Int. J. Speech Technol.*, vol. 50, no. 8, pp. 2339–2351, Aug. 2020.

[46] Y. Wu, M. Jiang, J. Lei, and H. Xu, "Named entity recognition in Chinese clinical text using deep neural network," *Stud. Health Technol. Informat.*, vol. 216, p. 624, Oct. 2015.

[47] S. Wunnava, X. Qin, T. Kakar, C. Sen, E. A. Rundensteiner, and X. Kong, "Adverse drug event detection from electronic health records using hierarchical recurrent neural networks with dual-level embedding," *Drug Saf.*, vol. 42, no. 1, pp. 113–122, Jan. 2019.

[48] P. Yadav, M. Steinbach, V. Kumar, and G. Simon, "Mining electronic health records (EHRs) a survey," *ACM Comput. Surv.*, vol. 50, no. 6, pp. 1–40, 2018.

[49] P. Yang, X. Sun, W. Li, S. Ma, W. Wu, and H. Wang, "SGM: Sequence generation model for multi-label classification," 2018, *arXiv:1806.04822*.

[50] Z. Yang, M. Dehmer, O. Yli-Harja, and F. Emmert-Streib, "Combining deep learning with token selection for patient phenotyping from electronic health records," *Sci. Rep.*, vol. 10, no. 1, pp. 1–18, Jan. 2020.

[51] J. Yao, K. Wang, and J. Yan, "Incorporating label co-occurrence into neural network-based models for multi-label text classification," *IEEE Access*, vol. 7, pp. 183580–183588, 2019.

[52] M.-L. Zhang, Y.-K. Li, X.-Y. Liu, and X. Geng, "Binary relevance for multi-label learning: An overview," *Frontiers Comput. Sci.*, vol. 12, no. 2, pp. 191–202, Apr. 2018.

[53] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 10, pp. 1338–1351, Oct. 2006.

[54] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, Jul. 2007.

[55] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014.

[56] X. Zhang, Q.-W. Zhang, Z. Yan, R. Liu, and Y. Cao, "Enhancing label correlation feedback in multi-label text classification via multi-task learning," 2021, *arXiv:2106.03103*.

[57] Y. Zhang, R. Chen, J. Tang, W. F. Stewart, and J. Sun, "LEAP: Learning to prescribe effective and safe treatment combinations for multimorbidity," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 1315–1324.

ZHEN YANG received the M.Sc. degree in data engineering and machine learning from Tampere University, Tampere, Finland, in 2019, where he is currently pursuing the Ph.D. degree in computing and electrical engineering. His research interests include natural language processing, multi-label text classifications, label dependency, and binary relevance methods.

FRANK EMMERT-STREIB received the B.Sc. and M.Sc. degrees in theoretical physics from the University of Siegen, Germany, and the Ph.D. degree in theoretical physics in physics and mathematics from the University of Bremen, Germany. He spent a sabbatical leave with the Department of Biostatistics and Computational Biology, Dana-Farber Cancer Institute, Harvard School of Public Health, USA, and the Statistics and Computational Biology Laboratory, CRUK Cambridge Research Institute, University of Cambridge, U.K. He is a Professor of data science with Tampere University, Finland, leading the Predictive Society and Data Analytics Laboratory.