MLGAN: a Meta-Learning based Generative Adversarial Network adapter for rare disease differentiation tasks

Rui Li
University at Buffalo
Buffalo, NY, United States
rli35@buffalo.edu

Andrew Wen
UTHealth Houston
Houston, TX, United States
Andrew.Wen@uth.tmc.edu

Jing Gao
Purdue University
West Lafayette, IN, United States
jinggao@purdue.edu

Hongfang Liu
UTHealth Houston
Houston, TX, United States
Hongfang.Liu@uth.tmc.edu

ABSTRACT

Rare disease diagnosis is very challenging due to the rarity and lack of scientific knowledge. Many patients with rare diseases take years to get diagnosed and many stay misdiagnosed or are not diagnosed. Comparing with traditional diagnosis prediction task, rare disease detection has the unique challenges of low prevalence and label noise. In this paper, we propose **M**eta-**L**earning based **G**enerative **A**dversarial Network module MLGAN, a rare disease detection enhancement module that can adapt any existing diagnosis prediction methods to rare disease detection task. We use generative adversarial network to generate synthetic positive embeddings and we use Meta-Weight-Net to automatically assign weight to real data and synthetic data. MLGAN helps us to leverage the time-aware sequential modeling ability in diagnosis prediction methods, and also mitigate the low prevalence and label noise of rare disease dataset. We empirically show that MLGAN can greatly boost the prediction performance and have good robustness on four real-world rare disease datasets. We release our code at https://github.com/ruilialice/MLGAN.

CCS CONCEPTS

• **Computing methodologies** → Machine learning; Artificial intelligence.

KEYWORDS

rare disease diagnosis, meta learning, generative adversarial networks

ACM Reference Format:
Rui Li, Andrew Wen, Jing Gao, and Hongfang Liu. 2023. MLGAN: a Meta-Learning based Generative Adversarial Network adapter for rare disease differentiation tasks. In *Proceedings of the 14th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics (BCB'23).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
BCB'23, September 3-6, 2023, Houston, TX, USA
© 2023 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Rare diseases, defined in the United States as those that affect fewer than 200,000 people, are collectively common despite their individual rarity. The more than 7000 rare diseases in existence collectively affect approximately 350 million people worldwide, incurring heavy social and financial burdens. Therefore, diagnosis and treatment of rare diseases is of great interest. Conversely, however, this task is inherently challenging. Due to their individual rarity, a positive diagnosis for rare disease can be subject to significant diagnostic odyssey, with 40% receiving an incorrect diagnosis at some point in their medical history, and 25% spending between 5 to 30 years before a correct diagnosis is given [14]. This issue is further exacerbated by the fact that scientific knowledge surrounding these diseases is esoteric, and relevant information consequently tends to be poorly documented. Meanwhile, a broad diversity of confounding disorders and relatively common symptoms may hide underlying rare diseases, and initial misdiagnosis is common [30].

Rare disease diagnosis through machine learning or deep learning is therefore of great interest, due to the potential for drastically reducing this diagnostic odyssey. We cast the rare disease detection problem as a diagnosis prediction problem, which aims to predict the patient's potential diseases for their next admission based on their historical EHR record. With the fast development of deep learning techniques, many diagnosis prediction approaches have been proposed. A key idea behind many of these approaches is to use sequential models such as LSTM [18] to learn the representation of longitudinal EHR data. However, these methods cannot be applied to the rare disease detection task directly.

Comparing with traditional diagnosis prediction, rare disease detection has the following challenges: (1) the prevalence is low. The insufficient amount of positive data hinders the model to extract the disease patterns, and the model may show poor robustness and generalization ability. (2) The label of the negative data may contain noise. Due to the difficulty associated with rare disease detection, patients with rare disease may be undiagnosed or misidentified as more common diseases with similar symptoms. In other words, the absence of a rare disease diagnosis does not mean the patient does not have the rare disease. Regarding these patients as negative samples introduces label noise.

To mitigate the issue of small positive data set sizes, plenty of data augmentation methods have been employed. A majority of

existing methods use generative models such as generative adversarial networks (GAN) to generate synthetic positive data [6, 9, 35]. Besides, [29] designs MaskEHR to randomly mask some diagnosis codes in some visits, and use the masked data as synthetic data, and further utilizes reinforcement learning selector to select the high-quality data. The common issue of the above mentioned approaches is that the synthetic data may be unreliable and treating the synthetic data and the real data equally may hurt the performance and make the model less convincing.

With respect to the issue of poor label quality, in traditional classification settings, undiagnosed and misdiagnosed patients would both be regarded as negative samples, which drastically increases the difficulty for models to learn the correct classifications. However, if we can "borrow" information from patients with uncertain diagnosis, the model can augment the pattern of rare disease class and learn more robust embeddings. This setting shares some similarity with an existing setting that has been studied in the data mining community: positive-unlabeled learning, where the training data consists of positive (P) and unlabeled (U) samples, and each unlabeled sample can belong to either the positive or negative (N) class [3]. Some previous works identify reliable N data in U data, and then perform ordinary supervised learning [26]. Other approaches regard U data as weighted P and N data simultaneously [21]. However, due to the similar symptoms of rare disease and the differential diagnosis, it is hard to identify reliable negative samples in unlabeled samples.

Due to the fast development of deep learning technique, many diagnosis prediction approaches have been proposed, which demonstrate outstanding ability in capturing time-aware sequential information in EHR data [2, 5, 27, 28]. However, these models typically fail to tackle these relatively unique challenges associated with rare disease diagnosis prediction. If the above mentioned two issues are addressed, we believe that methods in diagnosis prediction can still help the rare disease detection. Consequently, we introduce a **M**eta-**L**earning based **G**enerative **A**dversarial **N**etwork module MLGAN, a rare disease detection enhancement module that can be used to adapt existing diagnosis prediction methods to rare disease detection task. MLGAN employs a complementary GAN to generate synthetic embeddings for positive sample. Meanwhile, to mitigate the unreliability of the synthetic data, MLGAN utilizes Meta-Weight-Net[32] to impose weight on the real and synthetic sample loss. Our main contributions are summarized as follows:

(1) We propose a novel and effective module MLGAN that can adapt diagnosis prediction methods to rare disease detection task. MLGAN helps us to leverage the time-aware sequential modeling ability in diagnosis prediction methods while also mitigating the low prevalence and label noise associated with rare disease datasets.
(2) We incorporate Meta-Weight-Net into the module to automatically assign weights to real data and synthetic data generated by the complementary GAN. The Meta-Weight-Net can automatically assign smaller weight to synthetic data, so as to mitigate the relative unreliability of synthetic data.
(3) We empirically show that combining MLGAN and existing diagnosis prediction methods, the rare disease detection performance can be greatly boosted on four real-world datasets. These datasets contain patients with a rare disease of interest, which belong to case group, and patients diagnosed with the diseases that have similar symptoms or clinical presentations, which belong to control group.

2 RELATED WORK

Rare Disease Detection refers to the task of distinguishing patients with a rare disease from patients diagnosed with diseases sharing similar symptoms or clinical presentations. One of the biggest challenges is the low prevalence of rare disease. Thus, the dataset is highly imbalanced and the rare disease class has a very low number of observations. To address this challenge, most previous research mainly focus on generating synthetic samples to augment the positive data [6, 9, 29]. Medgan [6] ignores longitudinal event sequences and generates high-dimensional discrete variables as synthetic data. [29] proposes MaskEHR, which uses a reinforcement learning based selector to automatically select the high-quality positive samples from samples generated synthetically by masking information in existing positive samples. CONAN [9] uses the embedding of negative samples as seeds to generate complementary positive patterns with a complementary GAN. However, the common issue of these methods is that the synthetic data may not be as reliable as the real ones. Treating the synthetic data and the real data equally may degrade the performance.

Positive-Unlabeled Learning refers to the task of learning a model capable of distinguishing between positive and unlabeled samples. The unlabeled sample belongs to either the positive or negative class [3]. PU learning has a long history and is widely used in bioinformatics and computational biology [22]. Most methods first identify reliable negative samples, and learn the model based on the labeled positives and reliable negatives samples [17, 36]. Unbiased PU learning regards unlabeled data as weighted positive and negative data simultaneously [13], and the unbiased risk estimator [11, 12, 33] has been proposed. PU learning based on these unbiased risk estimators is the current state of the art and nnPU [20] is one of the most representative methods which is more robust against overfitting. Besides, some semi-supervised [31] PU learning methods are proposed. Beyond these approaches, self-supervised pretraining [10] has been applied to improve the performance.

Diagnosis Prediction refers to the task of predicting future diagnoses from a patient's historical medical record. Specifically, the aim is to predict a set of diagnosis codes that will be present at the next medical encounter given an input of a sequence of medical information (e.g., diagnoses, medications, lab tests, etc.) associated with previously occurring encounters. A key component of accomplishing this task is to use sequential models to project longitudinal medical information into a low-dimensional embedding. Various methods have been proposed to address this problem [2, 5, 23, 24, 27, 28]. They differ from each other in the strategies involved to learn such an embedding. Among these methods, Dipole [28] makes prediction based on sequences of medical codes only. It feeds the visit embeddings into a bidirectional LSTM and designs three attention mechanisms to learn the patient embedding.

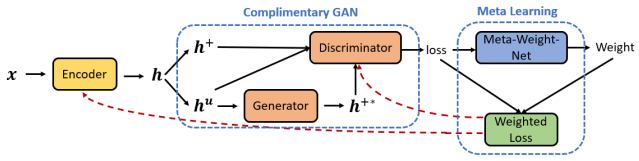


Figure 1: Framework for MLGAN.

HiTANet [27] incorporates information on the interval between different admissions into the transformer model, and proposes hierarchical time-aware attention networks to generate the patient embedding. Although these methods achieve good performance for diagnosis prediction, they can not be applied to rare disease detection directly due to the unique challenges of rare disease detection.

Classification on Imbalanced Datasets is common in real-world classification problems, such as fraud detection and medical diagnosis. The few observations of the minor class results in high false negative rates. Many approaches can mitigate the imbalance problem. At the data level, solutions include different forms of re-sampling and data augmentation, such as oversampling the minor class, undersampling of the major class and SMOTE [4]. At the model level, ensemble learning improves the generalization ability, including bagging and boosting. And sample re-weighting methods such as focal loss [25] assign more weights to hard or easily misclassified examples. Recently, Meta-Weight-Net [32] is proposed. It is inspired by the meta learning idea [16] and uses a multi-layer perceptron to map the training loss to a sample weight, and updates the weight calculator and classifier iteratively.

3 METHODOLOGY

For each patient, the clinical record $\boldsymbol{x}$ can be viewed as a sequence of encounters, where each encounter record may contain multiple diagnosis codes. The label y is binary indicating whether a patient is diagnosed with the specific rare disease we are aiming to differentiate. If y is positive, the patient is diagnosed with the rare disease and belongs to positive class. By contrast, if y is negative, the patient could either have the rare disease but not be diagnosed or the patient has a diagnosis with similar symptoms or clinical presentation. Such patients are left unlabeled. We define our task as follows: given a patient's longitudinal diagnosis information $\boldsymbol{x}$ and any existing diagnosis prediction model, we aim to design a module that can directly adapt existing diagnosis prediction models to rare disease detection task, and boost the prediction performance.

Figure 1 shows the overview of the proposed MLGAN framework, consisting of two major components: (1) the complementary GAN component that generates the synthetic embedding of positive samples, and (2) the meta learning component that uses the Meta-Weight-Net to impose weights on both real and synthetic sample loss. After a well-trained complementary GAN is obtained, the generator is used to generate the embedding of the synthetic positive samples to augment the positive data, and the discriminator is used as the classifier to classify whether the patient has a certain rare disease. In the ensuing subsections, we will first introduce the two components in greater detail, and then describe the overall training process tying these components together.

3.1 Complementary GAN

Inspired by CONAN[9], we utilize complementary GANs as part of MLGAN to generate synthetic positively labeled samples. In Figure 1, $\boldsymbol{x}$ represents real data, which contains both the positive samples and the unlabeled samples, Encoder (*Enc*) refers to any patient embedding model, which can learn the embedding of the patients. In our experiment, we select three widely-used diagnosis prediction models as the encoder, and evaluate our module's ability to adapt the selected diagnosis prediction models for the rare disease detection task. $\boldsymbol{x}$ is fed into the encoder, and we obtain the patient embeddings $\boldsymbol{h}$. $\boldsymbol{h}$ contains the embeddings of positive samples $\boldsymbol{h}^+$ and the embeddings of unlabeled samples $\boldsymbol{h}^u$. Here, $\boldsymbol{h}^+ = Enc(\boldsymbol{x}^+, \boldsymbol{\theta_e}) \sim p_{h^+}$ and $\boldsymbol{h}^u = Enc(\boldsymbol{x}^u, \boldsymbol{\theta_e}) \sim p_{h^u}$, $\boldsymbol{\theta_e}$ are the parameters of the encoder.

$\boldsymbol{h}^u$ is fed into the generator of the complementary GAN, denoted as G, and we obtain the embedding of the synthetic positive samples $\boldsymbol{h}^{+*} = G(\boldsymbol{h}^u, \boldsymbol{\theta_g})$, $\boldsymbol{\theta_g}$ are the parameters of the generator. In other words, the generator of the complementary GAN converts the embeddings of the unlabeled samples to the embeddings of the positive sample. By contrast, the generator of the traditional GAN generates synthetic embeddings from Gaussian noise. The synthetic embedding generated by the traditional GAN may follow the same distribution of the embedding of real positive samples in the training set, while the synthetic embedding generated by the complementary GAN may display different distribution and serve as pattern augmentation for positive samples.

Both $\boldsymbol{h}^+$ and $\boldsymbol{h}^{+*}$ are fed into the discriminator. The discriminator tries to better classify the real positive sample $\boldsymbol{h}^+$ and the synthetic positive sample $\boldsymbol{h}^{+*}$, and it is trained to maximize

$$\begin{aligned}\mathcal{L}(D, \boldsymbol{\theta_d}) =& \mathbb{E}_{h^+ \sim p_{h^+}}[log(D(\boldsymbol{h}^+, \boldsymbol{\theta_d}))] \\ &+ \mathbb{E}_{h^u \sim p_{h^u}}[log(1 - D(G(\boldsymbol{h}^u, \boldsymbol{\theta_g}), \boldsymbol{\theta_d}))]\end{aligned} \tag{1}$$

in which $\boldsymbol{\theta_d}$ and $\boldsymbol{\theta_g}$ are the corresponding parameters of the discriminator and generator. $D(\cdot, \boldsymbol{\theta_d})$ is a multilayer perceptron that outputs a single scalar. $D(\boldsymbol{h}, \boldsymbol{\theta_d})$ represents the probability that $\boldsymbol{h}$ comes from the real positive sample.

The generator tries to fool the discriminator, and it is trained to minimize

$$\begin{aligned}\mathcal{L}(G, \boldsymbol{\theta_g}) =& \mathbb{E}_{h^u \sim p_{h^u}}[log(1 - D(G(\boldsymbol{h}^u, \boldsymbol{\theta_g}), \boldsymbol{\theta_d}))] \\ &+ \lambda \| G(\boldsymbol{h}^u, \boldsymbol{\theta_g}) - \boldsymbol{h}^u \|_2\end{aligned} \tag{2}$$

λ is the hyper-parameter. Compared with traditional GANs, the second term of Eq.(2) measures the change between the original unlabeled samples and the converted samples, to ensure that $\boldsymbol{h}^{+*}$ does not vary much from $\boldsymbol{h}^u$. The generator and the discriminator are trained together adversarially, until the synthetic embedding of the positive samples generated by the generator can fool the discriminator.

3.2 Meta-Weight-Net

To address the class imbalance problem, sample re-weighting methods impose weights to the loss of different samples to enhance the robustness. Recently, Meta-Weight-Net (MW-Net) [32] has been proposed as a sample re-weighting method, and has been widely used in many classification tasks. We utilize the MW-Net to alleviate the class imbalance issue in rare disease detection. MW-Net uses

a multi-layer perceptron (MLP) mapping training loss to sample weight, and then iterating between weight recalculation and classifier updates. Classifier parameters depend on the sample weight calculated by MLP.

The optimal encoder parameter $\boldsymbol{\theta_e}$ and discriminator parameter $\boldsymbol{\theta_d}$ are calculated by minimizing the following weighted loss:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta_m}; \boldsymbol{\theta_e}, \boldsymbol{\theta_d}) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{V}(\ell(y_i, f(\boldsymbol{x}_i, \boldsymbol{\theta_e}, \boldsymbol{\theta_d})), \boldsymbol{\theta_m}) \cdot \ell(y_i, f(\boldsymbol{x}_i, \boldsymbol{\theta_e}, \boldsymbol{\theta_d})) \tag{3}$$

Here, $\boldsymbol{\theta}_m$ refers to MW-Net parameters, $f(\boldsymbol{x}_i, \boldsymbol{\theta_e}, \boldsymbol{\theta_d})$ is the classification logit, $\ell(y_i, \cdot)$ is the classification loss, and $\mathcal{V}(\ell(y_i, \cdot), \boldsymbol{\theta}_m)$ is the weight learned by MW-Net. As shown in Figure. 1, the input of MW-Net contains the classification loss of both real samples and synthetic embeddings. For real samples, $f(\boldsymbol{x}_i, \boldsymbol{\theta_e}, \boldsymbol{\theta_d}) = D(Enc(\boldsymbol{x}_i, \boldsymbol{\theta_e}), \boldsymbol{\theta_d})$, and y_i is the real label. Because we do not have the ground truth of the unlabeled samples, and most of the unlabeled samples are negative samples, thus we regard the unlabeled samples as negative. For synthetic samples, $f(\boldsymbol{x}_i, \boldsymbol{\theta_e}, \boldsymbol{\theta_d}) = D(G(Enc(\boldsymbol{x}_i^u, \boldsymbol{\theta_e}), \boldsymbol{\theta_g}), \boldsymbol{\theta_d}) = D(\boldsymbol{h}_i^{+*}, \boldsymbol{\theta_d})$, and y_i is positive. The unlabeled data $\boldsymbol{x}_i^u$ is fed into the encoder to get the embedding, which is subsequently fed to the generator to convert to synthetic positive embedding. $\boldsymbol{\theta_g}$ is fixed when training MW-Net.

The MW-Net parameter $\boldsymbol{\theta_m}$ is optimized based on meta learning [1]. The meta-data set $\hat{\mathcal{D}}$ is a small amount of balanced data. $\boldsymbol{\theta_m}$ is obtained by minimizing the following meta-loss:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta_e}, \boldsymbol{\theta_d}; \boldsymbol{\theta_m}) = \frac{1}{M}\sum_{i=1}^{M} \ell(y_i^{meta}, D(Enc(\boldsymbol{x}_i^{meta}, \boldsymbol{\theta_e}), \boldsymbol{\theta_d})) \tag{4}$$

$(\boldsymbol{x}_i^{meta}, y_i^{meta})$ is the meta-data, and M is the number of meta samples. $\boldsymbol{\theta_e}$ and $\boldsymbol{\theta_d}$ are the encoder and discriminator parameters respectively, which are related to $\boldsymbol{\theta}_m$. In practice, instead of solving Eq.(3) and Eq.(4) directly, we follow the method presented in [32] to instead update the classifier parameter $\boldsymbol{\theta_e}$ and $\boldsymbol{\theta_d}$, and MW-Net parameter $\boldsymbol{\theta_m}$ in the following three steps using stochastic gradient descent (SGD).

Firstly, based on the current $\boldsymbol{\theta}_m$, the intermediate variables $\hat{\boldsymbol{\theta}}_e$ and $\hat{\boldsymbol{\theta}}_d$ are calculated as

$$\begin{aligned}\hat{\boldsymbol{\theta}}_e &= \boldsymbol{\theta}_e - \alpha\frac{1}{N}\sum_{i=1}^{N} \mathcal{V}(\ell(y_i, f(\boldsymbol{x}_i, \boldsymbol{\theta_e}, \boldsymbol{\theta_d})), \boldsymbol{\theta_m})\nabla_{\boldsymbol{\theta_e}}\ell(y_i, f(\boldsymbol{x}_i, \boldsymbol{\theta_e}, \boldsymbol{\theta_d}))|_{\boldsymbol{\theta_e}} \\ \hat{\boldsymbol{\theta}}_d &= \boldsymbol{\theta}_d - \alpha\frac{1}{N}\sum_{i=1}^{N} \mathcal{V}(\ell(y_i, f(\boldsymbol{x}_i, \boldsymbol{\theta_e}, \boldsymbol{\theta_d})), \boldsymbol{\theta_m})\nabla_{\boldsymbol{\theta_d}}\ell(y_i, f(\boldsymbol{x}_i, \boldsymbol{\theta_e}, \boldsymbol{\theta_d}))|_{\boldsymbol{\theta_d}}\end{aligned} \tag{5}$$

α is the learning rate, and N is the batch size. In other words, in every training iteration, a batch of training samples are selected, and the encoder parameter $\boldsymbol{\theta}_e$ and discriminator parameter $\boldsymbol{\theta}_d$ move along the descent direction of the objective loss in Eq. (3) to get the intermediate variable $\hat{\boldsymbol{\theta}}_e$ and $\hat{\boldsymbol{\theta}}_d$.

The MW-Net parameters are then updated based on the intermediate variable $\hat{\boldsymbol{\theta}}_e$ and $\hat{\boldsymbol{\theta}}_d$.

$$\boldsymbol{\theta}_m = \boldsymbol{\theta}_m - \beta\frac{1}{M}\sum_{i=1}^{M} \nabla_{\boldsymbol{\theta}_m}\ell(y_i^{meta}, D(Enc(\boldsymbol{x}_i^{meta}, \hat{\boldsymbol{\theta}}_e), \hat{\boldsymbol{\theta}}_d))|_{\boldsymbol{\theta}_m} \tag{6}$$

β is the learning rate of MW-Net. Here, $\boldsymbol{\theta}_m$ moves along the objective gradient of Eq. (4) calculated on the meta-data.

Given the updated $\boldsymbol{\theta}_m$, we then update the classifier parameter, which is also presented as the red dashed line in Figure 1.

$$\begin{aligned}\boldsymbol{\theta}_e &= \boldsymbol{\theta}_e - \alpha\frac{1}{N}\sum_{i=1}^{N} \mathcal{V}(\ell(y_i, f(\boldsymbol{x}_i, \boldsymbol{\theta_e}, \boldsymbol{\theta_d})), \boldsymbol{\theta_m})\nabla_{\boldsymbol{\theta_e}}\ell(y_i, f(\boldsymbol{x}_i, \boldsymbol{\theta_e}, \boldsymbol{\theta_d}))|_{\boldsymbol{\theta_e}} \\ \boldsymbol{\theta}_d &= \boldsymbol{\theta}_d - \alpha\frac{1}{N}\sum_{i=1}^{N} \mathcal{V}(\ell(y_i, f(\boldsymbol{x}_i, \boldsymbol{\theta_e}, \boldsymbol{\theta_d})), \boldsymbol{\theta_m})\nabla_{\boldsymbol{\theta_d}}\ell(y_i, f(\boldsymbol{x}_i, \boldsymbol{\theta_e}, \boldsymbol{\theta_d}))|_{\boldsymbol{\theta_d}}\end{aligned} \tag{7}$$

3.3 Overall Training Process

In this part, we describe how the aforementioned two components are trained together. We tried three settings. For the first setting, when we train the MW-Net, the encoder is no longer updated, and only the discriminator is updated. For the second setting, when we train the MW-Net, both the encoder and discriminator are updated, and the embedding of the synthetic positive samples $\boldsymbol{h}^{+*}$ are updated with the encoder. In the third setting, when we train the MW-Net, both the encoder and discriminator are updated, but we fix the embedding of the synthetic positive samples $\boldsymbol{h}^{+*}$. The third setting achieves the best performance, and we present the third setting in this paper.

The overall training process is listed in Algorithm 1. In the first step, we train the complementary GAN, which is in the pseudo code line 1-7. The real data $\boldsymbol{x}$ is fed into the encoder to obtain the embedding $\boldsymbol{h}$. The generator converts embedding of unlabeled samples $\boldsymbol{h}^u$ to synthetic positive embedding $\boldsymbol{h}^{+*}$. The discriminator is then trained to discriminate between $\boldsymbol{h}^+$ and $\boldsymbol{h}^{+*}$. After the first step, the parameters of generator are fixed, and we use the generator to generate synthetic positive embeddings $\boldsymbol{h}^{+*}$ to augment the pattern of positive sample.

In the second step, we train Meta-Weight-Net, corresponding to the pseudo code on line 9-15. The input contains the real data and synthetic positive embeddings. The parameters of the generator $\boldsymbol{\theta}_g$ are fixed, and the parameters of encoder $\boldsymbol{\theta}_e$ and discriminator $\boldsymbol{\theta}_d$ are updated. At this step, the discriminator serves as the classifier, and it is used to classify whether the patient has a rare disease based on the embedding. Overall, the discriminator is trained twice.

Although our method is designed for binary classification, it can also be adapted to multi-class classification. In the latter setting, multiple rare diseases are considered at the same time, and a rare disease recommendation is performed for each patient. To handle this setting, in the second step, we need to replace the discriminator with a multi-class classification layer, and keep other parts in methodology.

4 EXPERIMENT

4.1 Experimental Setup

Dataset We leverage data from Mayo Clinic Enterprise Data Warehouse, which consists of data from the Rochester, Florida, and Arizona campuses as well as 49 sites across Minnesota, Wisconsin, and Iowa from the Mayo Clinic Health System. We sampled the data between 1994-2023 as part of the study. Four specific rare diseases were chosen as our focus:

Algorithm 1 Rare Disease Detection Enhancement Module Training

Input:
Training data $\mathcal{D}$, meta-data set $\hat{\mathcal{D}}$, training epoch for complementary GAN N_g, training epoch for Meta-Weight-Net N_m;
Output:
Well-trained encoder and discriminator
// Train the complementary GAN
for $i = 0, 1, \cdots, N_g$ **do**
  for every mini-batch of $\mathcal{D}$ **do**
    Obtain the patient embedding through the encoder;
    Maximize the discriminator loss by Eq. (1);
    minimize the generator loss by Eq. (2);
  end for
end for
Use the complementary GAN to generate the embeddings of synthetic positive sample $\boldsymbol{h}^{+*}$
// Train the Meta-Weight-Net
for $i = 0, 1, \cdots, N_m$ **do**
  for every mini-batch of $\mathcal{D}$, $\boldsymbol{h}^{+*}$ and $\hat{\mathcal{D}}$ **do**
    Compute the intermediate variable $\hat{\boldsymbol{\theta}}_e$ and $\hat{\boldsymbol{\theta}}_d$ by Eq. (5);
    Update $\boldsymbol{\theta_m}$ by Eq. (6);
    Update $\boldsymbol{\theta_e}$ and $\boldsymbol{\theta_d}$ by Eq. (7);
  end for
end for

(1) **Idiopathic Pulmonary Fibrosis (IPF)**: a condition in which the lungs become scarred and breathing becomes increasingly difficult.
(2) **Mastocytosis (MAS)**: a genetic immune disorder in which certain cells (mast cells) grow abnormally and cause a range of symptoms, including diarrhea and bone pain.
(3) **Hypereosinophilic Syndrome (HES)**: a group of blood disorders that occur when you have high numbers of eosinophils.
(4) **Rare Kidney Stones (RKS)**: a group of diseases of hereditary nephrolithiasis, including primary hyperoxaluria, cystinuria, APRT deficiency and dent disease [8].

For each of these four rare diseases, a list of possible differential diagnoses was first determined. The list consists of those diseases that share a similar clinical presentation. Then we extract all the case patients' records within two years before they are diagnosed with the rare disease. For a list of the ICD-10 codes used to define these rare diseases and their differentials, please refer to the Appendix. A rare disease patient cohort (case) and a differential patient cohort (control) was then constructed. For both cohorts, all patient diagnoses in the two years preceding the initial relevant disease diagnosis was retrieved, grouped temporally at an encounter level. For a case patient, we compute the encounter number. And we select the control patients with the same encounter number as the pair of the case patient. The relation of the case/control pair is one to many, which means that one case patient has multiple control patients as the pair. As both ICD 9 and ICD 10 codes simultaneously have been used in our dataset timeframe, ICD 9 codes were converted to ICD 10 codes using the General Equivalence Mappings (GEMS) [15]. Table 1 shows additional dataset details. All four datasets were severely skewed. Among these, MAS is the most imbalanced with 1.78% positive samples, and IPF is the least imbalanced.

Table 1: Statistics of dataset

statistic	dataset			
	IPF	MAS	HES	RKS
# of cases	3,320	1,710	5,317	2,243
# of controls	28,534	94,336	149,812	48,554
positive rate	10.40%	1.78%	3.43%	4.42%
# of ICD-10 Dx	10,744	15,526	17,277	13,764
avg # of admission	5.19	3.20	4.28	4.70
avg # of Dx per admission	6.98	4.82	4.19	5.11

Baselines We select three base models that are widely-used for any diagnosis prediction task, and we compare the performance with and without our module.

- `GRU` [7]: GRU feeds the visit embedding into GRU, and the hidden state of the last visit is directly used for prediction.
- `Dipole` [28]: Dipole feeds the visit embedding into a bidirectional LSTM, and uses the attention mechanism to measure the attention of hidden states for different visits. The weighted sum of hidden states is used for prediction.
- `HiTANet` [27]: HiTANet proposes the hierarchical time-aware attention network, it computes the local and global attention weight for each visit and combines them to generate the patient representations.

We also consider two methods that are specially designed for the rare disease detection task, and one state of the art method for PU learning.

- `nnPU` [20]: nnPU uses a non-negative risk estimator for PU learning, and is more robust against overfitting.
- `medGAN` [6]: medGAN generates high-dimensional discrete variables as synthetic EHR data. We use medGAN to generate patient records with rare disease and feed the patient representation into prediction layer.
- `CONAN` [9]: CONAN learns the self-attentive and hierarchical embedding for patient representation and uses the embedding of negative samples as seeds to generate complementary patterns with a complementary GAN.

Evaluation Measures Because the positive percentage in the unlabeled samples is extremely low, when we evaluate the performance, for testing data, we assume that the unlabeled examples all belong to the negative class. We use the following four metrics:

(1) **Area Under the Receiver Operating Characteristic Curve**: *AUROC* measures the entire two-dimensional area underneath the entire ROC curve.
(2) **Average Precision**: $AP = \sum_n (Rec_n - Rec_{n-1}) \cdot Pre_n$, it computes the weighted sum of precision, with the increase in recall from the previous threshold used as the weight, where Pre_n and Rec_n are the precision and recall at the n-th threshold.
(3) **F1 score**: $F1 = 2(Pre \cdot Rec)/(Pre + Rec)$, where Pre is precision and Rec is recall.

(4) **Recall**: $Rec = T_p/(T_p + F_n)$, where T_p is the number of true positive, and F_n is the number of false negative.

Implementation Details We implement all models with Pytorch, with a patient embedding dimension of 128. For all baselines, the learning rate used was $1e-4$, and Adam [19] was used as the optimizer. For baseline nnPU and medgan, instead of using longitudinal data, they use high-dimensional discrete variables as input. Following the setting in medgan [6], we construct a multi-hot vector as input. such that each ICD code in the dataset is mapped to a specific index in the input vector and the value of the vector at that index is 1 if a patient has that specific ICD code, or 0 otherwise. The Meta-Weight-Net is a multilayer perceptron with one hidden layer containing 100 dimensions. For all baselines, We randomly split the datasets into the training, validation and testing sets based on the number of patients in a 0.75:0.1:0.15 ratio. For the testing set, we assume that the unlabeled samples all belong to the negative class. For our module, we use all positive samples and randomly select the same number of negative samples in the original validation set as the meta data. We use SGD as the optimizer. The batch size of meta data is 100, and the batch size of training data is 512. For base model GRU and Dipole, the learning rate of Meta-Weight-Net is $1e-4$. And the initial learning rate of the embedding model and the discriminator is $1e-1$, and it is divided by 10 after per 80 epochs. For base model HiTANet, the learning rate of Meta-Weight-Net is $1e-5$. And the initial learning rate of the embedding model and the discriminator is $1e-2$, and it is divided by 10 after per 400 epochs. For the complementary GAN, we use the same setting of CONAN [9]. Both generator and discriminator has two hidden layers with 128 dimensions, and the output layer of the generator has the same dimension of the patient embedding. The training epoch of the Meta-Weight-Net is 1000, and we use grid search in range $[500, 1000, 1500, 2000]$ to find the best parameter for the training epoch of GAN.

4.2 Experimental Result

Performance Comparison Table 2 shows the performance of all baselines and the performance of three base models combined with our MLGAN module. Among the three methods that are specially designed for rare disease detection task, CONAN achieves the best result. It uses the complementary GAN to generate the embedding of synthetic samples that lie in between the positive and negative samples, which further help the discriminator to update its hyperplane. Neither nnPU nor medgan achieves good performance. We suspect this is due to their use of a multi-hot vector as input thus discarding the rich longitudinal information. Nevertheless, medgan achieves comparatively better results than nnPU. We use medgan to generate positive samples, and the generated samples and real samples are fed into the encoder to get the embedding, and the embedding is fed into the prediction layer. This step can be regarded as data augmentation for positive samples. But the performance of medgan is still less satisfactory, which indicates that the synthetic samples may largely share the same distribution as the positive samples and thus contribute little to augment the classification of "borderline" patients.

For the three base models, comparing the GRU, Dipole achieves better performance on three datasets except IPF. Dipole uses the bidirectional RNNs and apply the location-based attention mechanism to make prediction, which are able to capture visit dependency for long sequences. HiTANet achieves the best performance on F1 score and recall. It is because HiTANet uses the time-aware transformer to aggregate the visit representations with local time embeddings, and it also adopts a time-aware key-query attention mechanism to assign global weights to different time steps.

After combination with MLGAN, significantly improved performance is observed for all base models across all three datasets. For HiTANet, we observed an increase in recall of 0.359 for IPF, 0.276 for RKS, 0.142 for HES, and 0.053 for MAS; as well as increase in AUROC, AP and F1. For GRU and Dipole, F1 and recall also increase significantly, auroc and auprc have comparable performance. Meanwhile, comparing with methods designed specially for rare disease detection task, such as medgan and CONAN, HiTANet+MLGAN has much better result. These results suggest that our module can be used to directly adapt existing diagnosis prediction models to the rare disease detection task, and boost the performance significantly.

Ablation Study To further study the contribution of each component in MLGAN, we conduct ablation studies. Here, we select base model HiTANet and compare two reduced models with some components of MLGAN removed.

- $MLGAN^{m-}$: MLGAN without the complimentary GAN component, thus containing only the Meta-Weight-Net component.
- $MLGAN^{g-}$: MLGAN without the Meta-Weight-Net component, thus containing only the complimentary GAN component.

Table 3 shows the ablation study result on four datasets for HiTANet. Comparing with the base model, we observe that $MLGAN^{m-}$ has better performance for all datasets, except for MAS, suggesting that Meta-Weight-Net helps to detect positive samples and decrease the false negative rate. For HES, $MLGAN^{m-}$ has lower F1 score, specifically due to a higher false positive rate, suggesting that $MLGAN^{m-}$ tends to classify negative samples as positive. We also observe that $MLGAN^{g-}$ achieves comparable or better performance on four evaluation metrics for all datasets, except for HES. This suggests the complementary patterns generated by the complementary GAN can augment the positive samples, and further help the discriminator update its hyperplane by maximizing a margin between the generated samples. MLGAN in sum borrows the advantage of both $MLGAN^{m-}$ and $MLGAN^{g-}$. The numbers in bold are the best performance on the metric. We observe that MLGAN achieves the best performance on all metrics for HES. For IPF and RKS, MLGAN has the best performance on AUROC, F1 and recall. For MAS, MLGAN has the best performance on AP, F1 and recall.

The ablation study of MLGAN tells us (1) the Meta-Weight-Net is able to assign more weight to positive samples and further helps to detect positive samples and decrease the false negative rate; (2) the complementary GAN can generate augmented patterns of positive samples and increase AUROC and AP; (3) MLGAN borrows the advantage of Meta-Weight-Net and the complementary GAN, and the performance is greatly boosted.

Sample Weight Visualization In order to investigate how Meta-Weight-Net assigns weights to different types of samples, we plot the weight distribution of three types of training samples in

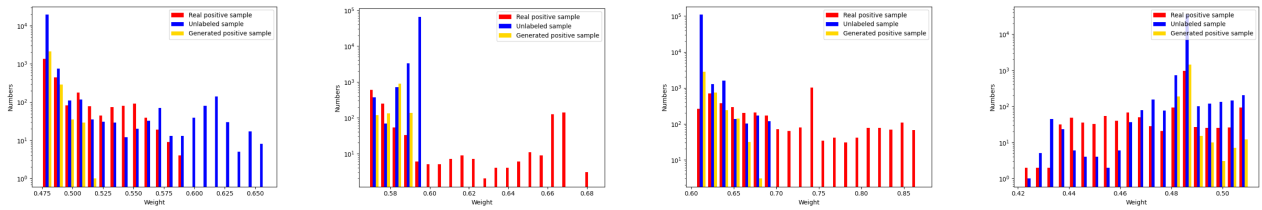
Table 2: Performance Comparison

Methods	IPF				MAS			
	AUROC	AP	F1	Recall	AUROC	AP	F1	Recall
GRU	0.902	0.646	0.594	0.496	0.772	0.25	0.197	0.113
GRU+MLGAN	0.904 (↑0.002)	0.643 (↓0.003)	0.655 (↑0.061)	0.683 (↑0.187)	0.756 (↓0.016)	0.226 (↓0.024)	0.251 (↑0.054)	0.284 (↑0.171)
Dipole	0.895	0.619	0.564	0.461	0.767	0.240	0.202	0.117
Dipole+MLGAN	0.904 (↑0.009)	0.625 (↑0.006)	0.632 (↑0.068)	0.709 (↑0.248)	0.755 (↓0.012)	0.219 (↓0.021)	0.254 (↑0.052)	0.319 (↑0.202)
HiTANet	0.932	0.667	0.564	0.450	0.912	0.485	0.541	0.398
HiTANet+MLGAN	0.954 (↑0.022)	0.681 (↑0.014)	0.705 (↑0.141)	0.809 (↑0.359)	0.925 (↑0.013)	0.629 (↑0.144)	0.584 (↑0.043)	0.451 (↑0.053)
nnPU	0.832	0.481	0.519	0.518	0.627	0.135	0.145	0.082
medgan	0.895	0.612	0.515	0.395	0.789	0.223	0.142	0.078
CONAN	0.920	0.629	0.593	0.536	0.915	0.236	0.405	0.389
Methods	HES				RKS			
	AUROC	AP	F1	Recall	AUROC	AP	F1	Recall
GRU	0.817	0.350	0.289	0.178	0.814	0.466	0.426	0.288
GRU+MLGAN	0.809 (↓0.008)	0.307 (↓0.043)	0.325 (↑0.036)	0.300 (↑0.122)	0.826 (↑0.012)	0.446 (↓0.020)	0.460 (↑0.034)	0.347 (↑0.059)
Dipole	0.823	0.360	0.299	0.186	0.817	0.475	0.435	0.295
Dipole+MLGAN	0.832 (↑0.009)	0.313 (↓0.047)	0.331 (↑0.032)	0.318 (↑0.132)	0.831 (↑0.014)	0.468 (↓0.007)	0.450 (↑0.15)	0.410 (↑0.115)
HiTANet	0.836	0.407	0.364	0.246	0.911	0.561	0.487	0.377
HiTANet+MLGAN	0.915 (↑0.079)	0.482 (↑0.075)	0.438 (↑0.074)	0.388 (↑0.142)	0.960 (↑0.049)	0.576 (↑0.015)	0.583 (↑0.096)	0.653 (↑0.276)
nnPU	0.708	0.252	0.250	0.144	0.739	0.375	0.361	0.228
medgan	0.818	0.345	0.271	0.161	0.785	0.444	0.460	0.315
CONAN	0.908	0.396	0.381	0.276	0.931	0.563	0.482	0.397

Table 3: Ablation Study on HiTANet

Methods	IPF				MAS			
	AUROC	AP	F1	Recall	AUROC	AP	F1	Recall
HiTANet	0.932	0.667	0.564	0.450	0.912	0.485	0.541	0.398
HiTANet+MLGAN	**0.954**	0.681	**0.705**	**0.809**	0.925	**0.629**	**0.584**	**0.451**
HiTANet+MLGAN^{m-}	0.940	**0.684**	0.627	0.534	0.721	0.332	0.576	0.342
HiTANet+MLGAN^{g-}	0.921	0.651	0.635	0.560	**0.927**	0.581	0.523	0.377
Methods	HES				RKS			
	AUROC	AP	F1	Recall	AUROC	auprc	F1	recall
HiTANet	0.836	0.407	0.364	0.246	0.911	0.561	0.487	0.377
HiTANet+MLGAN	**0.915**	**0.482**	**0.438**	**0.388**	**0.960**	0.576	**0.583**	**0.653**
HiTANet+MLGAN^{m-}	0.837	0.358	0.346	0.275	0.939	0.531	0.519	0.549
HiTANet+MLGAN^{g-}	0.863	0.412	0.347	0.244	0.916	**0.599**	0.480	0.323

Figure 2 for HiTANet+MLGAN. The x-axis represents the weight and uses a linear scale, while the y-axis represents the number of samples using a base-10 logarithmic scale. The real positive sample refers to the positive samples in the training set, the generated positive sample refers to the positive samples generated by the complimentary GAN, and the unlabeled sample refers to the unlabeled samples in the training set. It can be seen that the real positive samples have larger weights, and unlabeled samples have smaller weights, which implies that Meta-Weight-Net helps the discriminator pay more attention to the real positive samples than unlabeled samples. Meanwhile, we can observe that the real positive samples have larger weights comparing with the generated positive samples, which indicates that even if the complimentary GAN can augment the positive patterns, the generated samples are not as reliable as the real ones. Without Meta-Weight-Net, the discriminator will equally treat the generated samples and the real samples, which is not reasonable. This supports the necessity of incorporating Meta-Weight-Net in assigning weights to different types of samples.



(a) IPF dataset (b) MAS dataset (c) HES dataset (d) RKS dataset

Figure 2: Sample Weight Distribution Analysis on Training Data

Robustness In order to investigate how MLGAN will perform with different data quality, we compare the performance of HiTANet+MLGAN with different noise ratio in the positive samples of the training set. The noise ratio in the positive samples refers to the percentage of positive samples marked as unlabeled samples. For example, noise ratio 0.2 means that 20% positive samples in the training set are marked as unlabeled samples. Due to the positive unlabeled challenges associated with the rare disease detection task, a high noise ratio in the positive samples results in poor data quality, corresponding to datasets where a greater proportion of patients with the rare disease are undiagnosed or misdiagnosed. Among the four datasets, IPF is the least imbalanced. In this way, even if we randomly mark some positive samples as unlabeled samples, we still have a certain number of positive samples. Thus we check the robustness of our module on IPF dataset.

Figure 3 shows how HiTANet+MLGAN performs with different noise ratio on IPF dataset. The x-axis is the noise ratio in the positive samples of the training set, while validation and testing sets remain unchanged. We can observe that with the increase of the noise ratio, the performance of HiTANet drops rapidly on metric AP, F1 and Recall. However, HiTANet+MLGAN maintains good performance, especially on the AP and F1. Even with high noise ratio 0.6, there is only minor performance drop. The result illustrates that our module MLGAN is robust and the performance is maintained for dataset with poor data quality. This is very meaningful, because real-world rare disease datasets are likely to have significant label noise where rare diseases are left undiagnosed or misdiagnosed. The good robustness suggests that MLGAN can handle the label noise of real-world dataset with poor data quality.

4.3 Limitation

Our study has several limitations as follows: (1) There is no metric to measure the quality of the synthetic data generated by the complementary GAN. During the training of the complementary GAN, we can only fix the training epoch N_g and use the model of the last epoch, instead of saving the model with the best metric. Thus it is difficult to set N_g. (2) After we train the complementary GAN, the synthetic positive embedding is saved and no longer updated. However, when we train the Meta-Weight-Net, the encoder is updated and the embedding of the real samples are updated as well. Thus the synthetic positive embedding may follow a different distribution comparing with the real sample. It degrades the quality of the generated embedding. (3) The training of Meta-Weight-Net is time consuming, and choosing a suitable learning rate of the Meta-Weight-Net is not trivial.

5 CONCLUSION

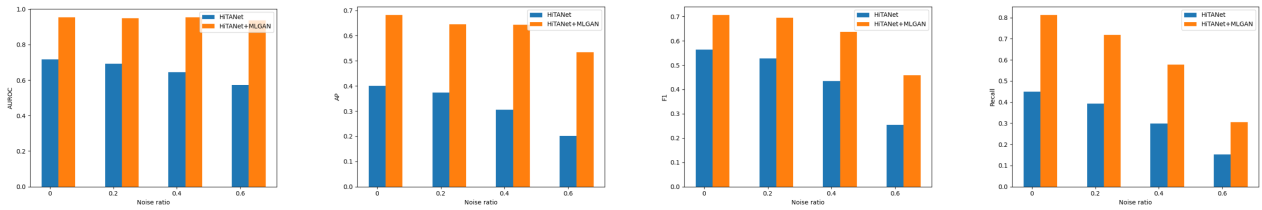
In this paper, we propose the module MLGAN that can be used to adapt existing diagnosis prediction methods to perform better on the rare disease detection task.MLGAN can leverage the time-aware sequential modeling ability in diagnosis prediction methods, and also mitigate the low prevalence and label noise of rare disease dataset. It combines the complementary GAN and Meta-Weight-Net. The complementary GAN uses the embedding of negative samples as seeds to generate complementary positive patterns. The Meta-Weight-Net can automatically assign weight to different type of sample loss, which mitigate the problem that the synthetic embeddings of EHR data may be unreliable. Meanwhile, we construct four real-world rare disease datasets, which contain the patients diagnosed with the rare disease and patients diagnosed with the differential diagnosis. Experiments demonstrate that our module MLGAN can greatly boost the performance of diagnosis prediction methods on rare disease detection task. Compared with methods designed specially for rare disease detection, diagnosis prediction methods combined with our module achieve comparable or better performance. Meanwhile, our module shows high robustness for datasets with high noise ratio, which is a likely constraint for real-world datasets. For the future work, we plan to use other generative models such as diffusion models [34] to generate more reliable synthetic samples for rare disease class, and further improves the performance of rare disease detection.

6 ACKNOWLEDGEMENTS

Research reported in this publication was supported by the National Center for Advancing Translational Science of the National Institutes of Health under award number U01TR002062. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. Use of patient data for this study was approved by the Mayo Clinic Institutional Review Board (#20-001137) for human subjects research.

REFERENCES

[1] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. 2016. Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems* 29 (2016).



(a) AUROC (b) AP (c) F1 (d) Recall

Figure 3: Performance comparison for IPF dataset with different noise ratio in positive samples

[2] Inci M Baytas, Cao Xiao, Xi Zhang, Fei Wang, Anil K Jain, and Jiayu Zhou. 2017. Patient subtyping via time-aware LSTM networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 65–74.

[3] Jessa Bekker and Jesse Davis. 2020. Learning from positive and unlabeled data: A survey. *Machine Learning* 109 (2020), 719–760.

[4] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[5] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. 2016. RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism. In *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (Eds.), Vol. 29. Curran Associates, Inc.

[6] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng Sun. 2017. Generating multi-label discrete patient records using generative adversarial networks. In *Machine learning for healthcare conference*. PMLR, 286–305.

[7] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *NeurIPS 2014 Workshop on Deep Learning* (2014).

[8] Rare Kidney Stone Consortium. 2015. Rare Kidney Stone Consortium. (2015). https://www.rarekidneystones.org/

[9] Limeng Cui, Siddharth Biswal, Lucas M Glass, Greg Lever, Jimeng Sun, and Cao Xiao. 2020. CONAN: complementary pattern augmentation for rare disease detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 614–621.

[10] Emilio Dorigatti, Jonas Schweisthal, Bernd Bischl, and Mina Rezaei. 2022. Robust and Efficient Imbalanced Positive-Unlabeled Learning with Self-supervision. *arXiv preprint arXiv:2209.02459* (2022).

[11] Marthinus Du Plessis, Gang Niu, and Masashi Sugiyama. 2015. Convex formulation for learning from positive and unlabeled data. In *International conference on machine learning*. PMLR, 1386–1394.

[12] Marthinus C Du Plessis, Gang Niu, and Masashi Sugiyama. 2014. Analysis of learning from positive and unlabeled data. *Advances in neural information processing systems* 27 (2014).

[13] Charles Elkan and Keith Noto. 2008. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 213–220.

[14] EURORDIS. 2007. Survey of the delay in diagnosis for 8 rare diseases in Europe (EurordisCare2). *Fact sheet EurordisCare 2* (2007).

[15] Centers for Medicare, Medicaid Services, and the National Center for Health Statistics. 2018. Diagnosis Code Set General Equivalence Mappings. (2018). https://www.cms.gov/medicare/coding/icd10/2018-icd-10-cm-and-gems

[16] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. 2018. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*. PMLR, 1568–1577.

[17] Gabriel Pui Cheong Fung, Jeffrey Xu Yu, Hongjun Lu, and Philip S Yu. 2005. Text classification without negative examples revisit. *IEEE transactions on Knowledge and Data Engineering* 18, 1 (2005), 6–20.

[18] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[19] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations*.

[20] Ryuichi Kiryo, Gang Niu, Marthinus C Du Plessis, and Masashi Sugiyama. 2017. Positive-unlabeled learning with non-negative risk estimator. *Advances in neural information processing systems* 30 (2017).

[21] Wee Sun Lee and Bing Liu. 2003. Learning with positive and unlabeled examples using weighted logistic regression. In *ICML*, Vol. 3. 448–455.

[22] Fuyi Li, Shuangyu Dong, André Leier, Meiya Han, Xudong Guo, Jing Xu, Xiaoyu Wang, Shirui Pan, Cangzhi Jia, Yang Zhang, et al. 2022. Positive-unlabeled learning in bioinformatics and computational biology: a brief review. *Briefings in bioinformatics* 23, 1 (2022), bbab461.

[23] Rui Li and Jing Gao. 2022. Multi-modal contrastive learning for healthcare data analytics. In *2022 IEEE 10th International Conference on Healthcare Informatics (ICHI)*. IEEE, 120–127.

[24] Rui Li, Fenglong Ma, and Jing Gao. 2021. Integrating multimodal electronic health records for diagnosis prediction. In *AMIA Annual Symposium Proceedings*, Vol. 2021. American Medical Informatics Association, 726.

[25] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[26] Bing Liu, Wee Sun Lee, Philip S Yu, and Xiaoli Li. 2002. Partially supervised classification of text documents. In *ICML*, Vol. 2. Sydney, NSW, 387–394.

[27] Junyu Luo, Muchao Ye, Cao Xiao, and Fenglong Ma. 2020. Hitanet: Hierarchical time-aware attention networks for risk prediction on electronic health records. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 647–656.

[28] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. 2017. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 1903–1911.

[29] Fenglong Ma, Yaqing Wang, Jing Gao, Houping Xiao, and Jing Zhou. 2020. Rare Disease Prediction by Generating Quality-Assured Electronic Health Records. In *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM, 514–522.

[30] rarediseaseday.org. 2020. Long diagnosis, misdiagnosis, or no diagnosis – how rare diseases go under. (2020). https://www.rarediseaseday.org/what-is-a-rare-disease/

[31] Tomoya Sakai, Gang Niu, and Masashi Sugiyama. 2018. Semi-supervised AUC optimization based on positive-unlabeled learning. *Machine Learning* 107 (2018), 767–794.

[32] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems* 32 (2019).

[33] Guangxin Su, Weitong Chen, and Miao Xu. 2021. Positive-Unlabeled Learning from Imbalanced Data.. In *IJCAI*. 2995–3001.

[34] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Yingxia Shao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. 2022. Diffusion models: A comprehensive survey of methods and applications. *arXiv preprint arXiv:2209.00796* (2022).

[35] Kezi Yu, Yunlong Wang, Yong Cai, Cao Xiao, Emily Zhao, Lucas Glass, and Jimeng Sun. 2019. Rare disease detection by sequence modeling with generative adversarial networks. In *Proceedings of International Conference on Machine Learning workshop*.

[36] Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. 2003. Learning with local and global consistency. *Advances in neural information processing systems* 16 (2003).

A ICD CODES OF RARE DISEASE AND DIFFERENTIAL DIAGNOSIS

Table 4 and Table 5 show the ICD codes of rare diseases incuding Idiopathic pulmonary fibrosis(IPF), mastocytosis(MAS), rare kidney stone(RKS) and hypereosinophilic syndrome (HES), the icd codes of the corresponding differential diagnosis is also listed. ICD codes

Table 5: ICD Codes of Rare Disease and Differential Diagnosis

rare disease	**ICD 9**	**ICD 10**
hypereosinophilic syndrome (HES)	288.3	D72.11X
differential diagnosis	**ICD 9**	**ICD 10**
Asthma	493.X	J45.X
Chronic Myelogenous Leukemia (CML)	205.1X	C92.1X
Eosinophilic Granulomatosis with Polyangiitis (Churg-Strauss Syndrome)	446.4	M30.1
Eosinophilia-Myalgia Syndrome	710.5	M35.89
Eosinophilic Fasciitis	728.89	M35.4
Eosinophilic Gastroenteritis	558.41	K52.81
Hodgkin Lymphoma	201.X	C81.X
Strongyloidiasis	127.2	B78.X

Table 4: ICD Codes of Rare Disease and Differential Diagnosis

rare disease	**ICD 9**	**ICD 10**
Idiopathic pulmonary fibrosis(IPF)	516.31	J84.112
differential diagnosis	**ICD 9**	**ICD 10**
Aspiration Pneumonitis and Pneumonia	507	J69.0
Viral Pneumonia	480.9	J12.9
Cardiogenic pulmonary edema	514	J81.1
Chemical Worker's Lung /Occupational asthma	502	J62.8
Chlamydial Pneumonia	483.1	J16.0
Bacterial Pneumonia	482.9	J15.9
Pneumococcal Infections (Streptococcus pneumoniae)	481	J13
Chronic aspiration pneumonia	507	J69.0
Restrictive Lung Disease	518.89	J98.4
rare disease	**ICD 9**	**ICD 10**
mastocytosis(MAS)	202.6 757.33	C96.2X D47.0X
differential diagnosis	**ICD 9**	**ICD 10**
Acute Urticaria	708.9	L50.9
Inflammatory Bowel Disease	555.X, 556.X	K52.9
Irritable Bowel Syndrome (IBS)	564.1	K58.X
Malabsorption	579.X	K90.X
Myeloproliferative Disease	238.79	D47.1, D47.4
rare disease	**ICD 9**	**ICD 10**
rare kidney stone(RKS)	271.8, 270.0, 276, 588.8X	E74.8X, N25.8X E72.53, E72.01 E79.8
differential diagnosis	**ICD 9**	**ICD 10**
kidney stone	592	N20.X

containing X in the last digit refers to all ICD codes begin with the prefix. For example, C96.2X refers to C96.20-C96.29.