Stroke Risk Prediction With Hybrid Deep Transfer Learning Framework

Jie Chen, Yingru Chen, Jianqiang Li, Jia Wang, Zijie Lin, and Asoke K. Nandi

***Abstract*—Stroke has become a leading cause of death and long-term disability in the world with no effective treatment. Deep learning-based approaches have the potential to outperform existing stroke risk prediction models, but they rely on large well-labeled data. Due to the strict privacy protection policy in health-care systems, stroke data is usually distributed among different hospitals in small pieces. In addition, the positive and negative instances of such data are extremely imbalanced. Transfer learning can solve small data issue by exploiting the knowledge of a correlated domain, especially when multiple source of data are available. In this work, we propose a novel Hybrid Deep Transfer Learning-based Stroke Risk Prediction (HDTL-SRP) scheme to exploit the knowledge structure from multiple correlated sources (i.e., external stroke data, chronic diseases data, such as hypertension and diabetes). The proposed framework has been extensively tested in synthetic and real-world scenarios, and it outperforms the state-of-the-art stroke risk prediction models. It also shows the potential of real-world deployment among multiple hospitals aided with 5 G/B5G infrastructures.**

***Index Terms*—stroke risk prediction, transfer learning, generative adversarial networks, active learning, Bayesian optimization.**

Manuscript received December 14, 2020; revised May 18, 2021; accepted June 5, 2021. Date of publication June 11, 2021; date of current version January 5, 2022. This work was supported in part by the National Key R&D Program of China under Grant 2020YFA0908700, in part by the National Nature Science Foundation of China under Grants U1713212, 62072315, 62073225, 61806130, 61836005, and 62006157, in part by the Natural Science Foundation of Guangdong Province-Outstanding Youth Program under Grant 2019B151502018, in part by the Guangdong "Pearl River Talent Recruitment Program" under Grant 2019ZT08X603, in part by the Technology Research Project of Shenzhen City under Grant JSGG20180507182904693, and in part by the Public Technology Platform of Shenzhen City under Grant GGFW2018021118145859. *(Corresponding author: Jianqiang Li.)*

Jie Chen, Jianqiang Li, Jia Wang, and Zijie Lin are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: chenjie@szu.edu.cn; lijq@szu.edu.cn; jia.wang@szu.edu.cn; 2060271036@email.szu.edu.cn).

Yingru Chen is with the Huazhong University of Science and Technology Union Shenzhen Hospital (Nanshan Hospital), Shenzhen 518051, China (e-mail: chencyr261@163.com).

Asoke K. Nandi is with the Electronic and Electrical Engineering, Brunel University London, Uxbridge, Middlesex UB8 3PH, U.K., and also with Shenzhen University, Shenzhen 518060, China (e-mail: asoke.nandi@brunel.ac.uk).

Digital Object Identifier 10.1109/JBHI.2021.3088750

I. Introduction

STROKE is one of the most prevalent diseases which could lead to death or long-term disability among elderly people all over the world. In a recent report [1], around 795 000 people experience a new or recurrent stroke each year in the US; one stroke incident occurs in approximately every 40 seconds. Among the patients who suffered strokes, one in five would die within one year [2]. For the survivals, the cost of treatment and rehabilitation becomes an extremely high burden to their families and the health-care system. From 2014 to 2015, the direct and indirect cost due to stroke incidents was about 45.5 billion US dollars [3]. Thus, accurate stroke prediction is highly desirable so that the cost can be reduced with early interventions to delay the onset of and to reduce the risks of stroke.

There exist several works which exploit medical data (e.g., electronic health record and retinal image) to develop Stroke Risk Prediction (SRP) Models. These methods can be broadly categorized into classical machine learning approaches [4], [5] (e.g., Support Vector Machine (SVM), Decision Tree, Logistic Regression) and deep learning-based approaches [6]–[11]. It is reported that deep neural network (DNN) can achieve best performance in stroke prediction [8]. However, a well-known drawback is that such model relies on the availability of large well-labeled data. In real-world scenario, the quantity of reliable data that is required may not be readily available [12]. Due to strict privacy protection policy in health-care system, sharing stroke data between hospitals is usually difficult. Thus, the full set of stroke data tends to be distributed among multiple hospitals in small subsets. In addition, stroke data contains extremely imbalanced positive and negative instances. Thus, the DNN-based SRP models could work poorly in real-world deployment [13].

Though the stroke data is small, some common chronic diseases (e.g., hypertension and diabetes) have sufficiently larger data and are highly correlated with stroke development in clinical trials [14], [15]. When multiple correlated sources are available, Transfer Learning (TL) approaches offer a suitable framework to address small data issue [16], [17]. Most of existing TL works are single transfer approaches including feature transfer [18], [19], instance transfer [20]–[22], network transfer [23], [24]. A recent work [25] proposed a hybrid adapted-embedding method and empirically showed that hybrid transfer outperforms single transfer approaches. Transfer learning is also used in

Meta-learning framework for low-resource predictive modeling with patient EHRs [26]. However, existing approaches do not consider the issue of imbalanced labels in the target domain. In contrast, this work proposes a hybrid transfer approach that incorporates generative instance transfer coupled with active selection which can exploit external stroke data to address the label imbalance issue. The generative instance transfer can share high-quality synthetic stroke data for training SRP model while preserving patients' privacy [27] and active instance selection allows the most informative generated instances to be transferred to the target domain [28]. Furthermore, the training and inference of framework are designed in a distributed fashion such that it can take advantage of high data transmission and stringent latency in 5 G/B5G cellular network [29]–[31].

The proposed framework can achieve a better ability in establishing SRP model. However, the parameters such as the number of transferred layer and the transferred sequence of different source domains are vital factors for model performance. Common methods such as grid and random search for parameter tuning are often inefficient due to the search space being too large [32]. Bayesian Optimization (BO) is an approach for model-based global optimization of black-box function and the most universally used model for BO is a Gaussian process due to its simplicity and flexibility in constructing a probabilistic model of objective function [33]. Therefore, BO is used to find the best parameter in SRP model.

The contributions of this work are as follows:

- This work proposes a novel Hybrid Deep Transfer Learning-based Stroke Risk Prediction (HDTL-SRP) framework that allows simultaneous exploitation of multiple correlated sources of medical data (e.g., hypertension, diabetes, and external stroke data) to train a SRP model in a local hospital where only small and imbalanced stroke data is available (Section IV).
- The proposed framework is extensively compared with the state-of-the-art SRP models in synthetic scenario of stroke risk prediction. In addition, our approach is tested in a real-world dataset which contains 2426 stroke incidents recorded in three collaborating hospitals during 2012-2017. The empirical results show that the performance of HDTL-SRP outperform its counterparts in both synthetic and real-world scenarios (Section V).
- The proposed HDTL-SRP framework can be deployed among multiple hospitals to take advantage of the distributed medical data while preserving the patients' privacy (Section V-B). In addition, this decentralized framework can gain efficiency when the 5 G/B5G infrastructures are available among hospitals.
- In this work, Bayesian Optimization is used to find the best parameters such as the number of transferred layer and the transferred sequence of different source domains for HDTL-SRP model (Section V-C).

II. Related Work

Several studies have used classical machine-learning/deep learning methods to construct SRP model. Khosla *et al.* [4] proposed a feature selection algorithm combined with SVM and carried out experiments in a dataset that has 4988 examples with 299 occurrences of stroke. Miguel *et al.* [5] used machine learning techniques to predict the functional outcome of a patient three months after the initial stroke in a dataset with 425 acute stroke patients. Lim *et al.* [7] used a deep learning approach to develop SRP model using retinal images and achieved acceptable performance. But collecting retinal images is often time-consuming and expensive. From the above works, we can find that the amount of stroke data is less and the dataset is imbalanced. To solve the imbalance issue, Liu *et al.* [11] proposed a DNN-based model to predict stroke on an imbalanced dataset which contains 43 400 records of patients with 783 occurrence of stroke. However, the above methods cannot address the small data and imbalance issues of stroke data.

There exist some works to solve the issues of small and imbalanced data. They can be approximately classified as re-sampling and data augmentation. In re-sampling approach, data is under-sampled or over-sampled from original data. A classic over-sampling method is SMOTE [34]. It generates samples between the minority and their nearest neighbor. Some widely used data augmentation methods are geometric transformation, flipping, rotation, cropping, and so on [35]. However, most of the above methods can only be applied to image data. Another data augmentation method is generative adversarial networks (GAN) [36]. It can generate artificial instances to maintain feature similar to the original dataset. Meanwhile, compared to other techniques, GAN can reduce information leakage. The above methods solve the mentioned issues. But they cannot exploit extra data when there are other data resources. Transfer learning (TL) can address small data issue by training a model on a related big dataset and then using knowledge of this model in target task, especially when multiple correlated sources are available [37]. But transfer learning cannot handle imbalance issue. Inspired by the ideas of GAN and TL, we propose a hybrid deep transfer learning framework to address the issues of small and imbalanced data.

Intuitively, centralizing all available data from other hospitals can build a better SRP model when amounts of stroke data available in local hospitals are small. Secure Multiparty Computation (SMC) [38] model involves multiple party data and protects data privacy. But it demands each party knows nothing except its input and output which is difficult to achieve. Another recently proposed framework, federated learning [39], is also used to protect multiple party data privacy. It can exploit multiple distributed data to establish a better model. In federated learning, data in different places would not be transmitted and the model is encrypted during training. But federated learning framework cannot solve the issue of small data. Notably, the above methods have not been deployed in SRP application.

The parameter of Network Weight transfer module is an import factor of HDTL-SRP model performance. Traditionally, manual search and grid search [40] are strategies for hyper-parameter optimization in a neural network. For the same time budget, random search [41] finds better models than grid search results by effectively searching a larger and less promising configuration space. But they find a proper hyper-parameter randomly or rely on the expert's experience. Genetic Algorithm (GA) [42]–[44], is an evolutionary search algorithm used to solve optimization. However, GA demands enough initial sample points in hyper-parameter optimization so that the optimization efficiency is usually low. Bayesian Optimization (BO) [45]

TABLE I
NOTATION TABLE

$\mathcal{S}_{\text{ST}}$	Stroke source domain.
$\mathcal{S}_{\text{HT}}$	Hypertension source domain.
$\mathcal{S}_{\text{DB}}$	Diabetes source domain.
$\mathcal{T}_{\text{ST}}$	Stroke target doamin.
$\mathbf{x}, X$	Feature vector, feature vector set.
τ	Batch size of selected samples.
Y	Label space.
$\mathbf{r}$	$\mathbf{x}_i^T W + \mathbf{b}$
$\mathbf{b}$	Bias.
$\mathbf{y}$	Actual label.
$\hat{\mathbf{y}}$	Predicited Label.
W	Weight matrix of DNN.
$ReLU(\cdot)$	Rectified linear function.
$tanh(\cdot)$	Hyperbolic tangent function.
$\xi(\cdot)$	Sigmoid function.
L	Loss function.
D	Discriminator.
G	Generator.
z	Sampled Noise.
$\mathbb{E}[.]$	Expected value.
$V(\cdot)$	Value function of GAN.
X_g	Generated samples features matrix.
X_c	Features matrix of chosen samples.
X_p	Features matrix of positive samples.
$\mathbb{H}[.]$	Entropy.
θ	Model prameters.
$\mathcal{D}_t, \mathcal{D}_v$	The data of training set or validation set.
$\mathbf{l}$	The layer number vector which need to optimize.
$\mathbf{c}$	optimized parameter.
$\kappa(\cdot,\cdot)$	Kernel function.
S	Parameter set.
$\Phi(\cdot)$	Cumulative distribution function.
$\phi(\cdot)$	Probability density function.
μ	mean of Gaussian process.
σ	variance of Gaussian process.

is a flexible approach in hyper-parameter optimization, while BO based on Gaussian processes [46] achieve successful implementation, as the Spearmint system [47].

III. FORMULATION OF STROKE RISK PREDICTION

Traditionally, the problem of SRP assumes that a set of stroke data $\mathcal{T}$ is available locally for training SRP models (e.g., DNN, SVM, and Decision Tree). Each point in $\mathcal{T}$ is a tuple (X, y) where X is a feature matrix that contains the patient's attributes extracted from his/her medical records and $y \in \{+1, -1\}$ is a binary label with $+1$ indicating an occurrence of stroke incident (The important notations are listed in Table I). SRP model aims to learn the underlying function $f(.)$ from stroke data. then, given feature X^* of a new patient, SRP model can automatically return a prediction $\hat{y} = f(X^*)$ to the doctor.

In real-world scenarios, the stroke data $\mathcal{T}$ in a specific local hospital tends to be small and imbalanced: (1) In the same city, patients' records are unevenly distributed among multiple hospitals; intuitively, the hospitals which are located far away from denser-populated districts or have reduced level facilities would attract fewer patients. In addition, it is extremely strict to share stroke data among hospitals due to privacy protection policy; (2) As stroke are rare incidents among all patients' hospital visits, the positive versus negative instances of stroke dataset are highly imbalanced. Such small and imbalanced stroke data could result in a poorly-performed SRP model using the traditional workflow.

To address the issue with small stroke data, this work is motivated by researches in health-care domain. Clinical trials have revealed that hypertension is one of the most important risk factors in the development of stroke [14] and diabetes is a strong determinant of or factor in ischemic stroke due to its impact on cardiovascular system among middle-aged women [15]. As hypertension and diabetes are common chronic diseases among elderly patients, much more records of hypertension and diabetes are available than that of stroke. Then, the question is *how to exploit effectively data from hypertension/diabetes to improve SRP models?*

The above question can be formulated under the deep transfer learning framework [48]. Briefly, the SRP model is designed as a DNN which can be first trained in source domains using hypertension $\mathcal{S}_{\text{HT}}$ or diabetes $\mathcal{S}_{\text{DB}}$ data; then, the weights of the trained network can be transferred to the target domain of stroke where the DNN is fine-tuned using local stroke data $\mathcal{T}_{\text{ST}}$. However, the imbalanced labels in the target domain still could cause performance degeneration when fine-tuning the DNN. To address this issue without violating the privacy protection policy, this work also attempts to incorporate more data from external sources by applying generative adversarial networks (GAN) [36] in the source domains.

In this work, we formulate the learning task in the target domain as stroke risk prediction in a local hospital (see Fig. 1). As the stroke data in the target domain is usually insufficient, the learning task can be improved with the help from various source domains of different diseases (e.g., Hypertension/Diabetes). However, due to the privacy protection policy among the health-care systems, the medical datasets are collected and stored in different hospitals; direct exchange of datasets between hospitals is not feasible. A reasonable solution is to formulate a source domain as a specific disease at a certain hospital. Then, the goal is to improve the learning task in target domain (i.e., stroke risk prediction at local hospital) via transferring knowledge structures from multiple source domains across both local and external hospitals.

In Fig. 1, the target and source domains are illustrated in a scenario where one local hospital and K external hospitals are considered. The core of the system is the proposed Hybrid Deep Transfer Learning-based Stroke Risk Prediction (HDTL-SRP) framework for exploiting all relevant sources of data to train an accurate SRP model. The details of HDTL-SRP will be described in the following section.

IV. HYBRID DEEP TRANSFER LEARNING FOR STROKE RISK PREDICTION

To alleviate the issues caused by small and imbalanced stroke data, this work proposes a Hybrid Deep Transfer Learning (HDTL) approach that transfers knowledge structure from multiple source domains distributed among multiple hospitals to the target domain of stroke. The proposed HDTL-SRP framework works in a distributed fashion without having to share directly the patients' records between hospitals. It consists of three

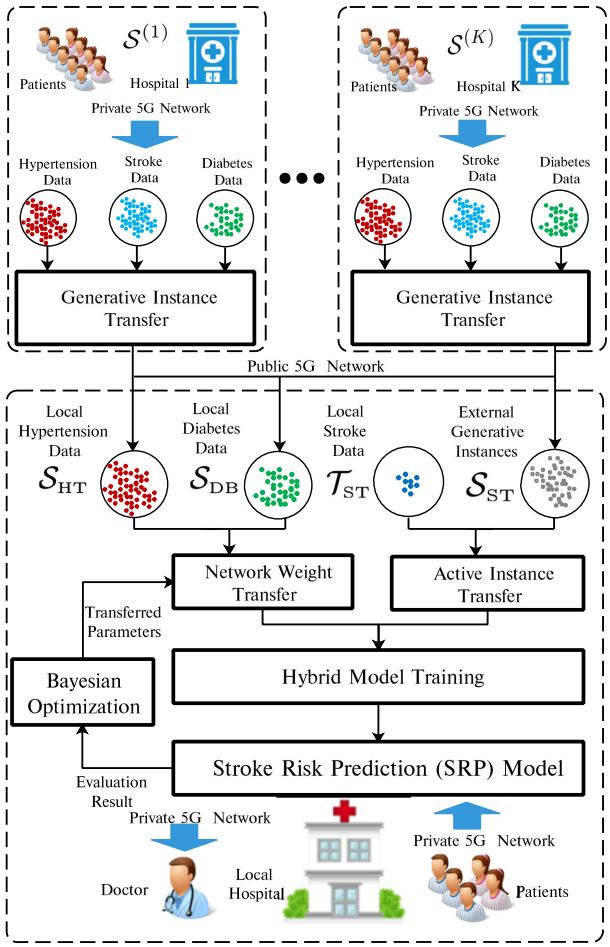

Fig. 1. Deployment of HDTL-SRP framework in multiple hospitals with 5 G infrastructures.

components: (1) Generative Instance Transfer (GIT) applying GAN in external data to generate synthetic instances for model training purpose, (2) Network Weight Transfer (NWT) making use of data from highly correlated diseases (i.e., hypertension or diabetes), (3) Bayesian Optimization (BO) to find the best transferred parameters, and (4) Active Instance Transfer (AIT) selecting more informative synthetic stroke instances to create a balanced stroke dataset which is then exploited to fine-tune SRP model. In the following, we will describe each component in detail.

A. Generative Instance Transfer Using External Stroke Data

Intuitively, the hospitals of higher rank or those located closer to densely-populated districts tend to own more electronic health records (EHR) on strokes. However, due to the strict data protection policy in health-care domain for preserving patients' privacy, the invaluable stroke data cannot be easily shared for training SRP model. To address this issue, the GIT component of HDTL-SRP is deployed in each hospital; it can exploit the historical EHR of the stroke instances to train a GAN [36] model. Then, the knowledge structure hidden in the stroke data can be transferred to the target domain via synthetic generative instances.

Specifically, in an arbitrary hospital i, the GAN model consists of a generator G and a discriminator D which are both multilayer perceptrons specified by θ_g and θ_d, respectively. The generator G aims to generate synthetic instances that cannot be distinguished from the positive instances in stoke data $\mathcal{S}_{\mathrm{ST}}^{(\mathrm{i})}$. On the other hand, the discriminator D aims to determine successfully whether an input instance X is real or fake. The parameters θ_g and θ_d are optimized by playing a minmax game according to the objective function:

$$\min_{\theta_g}\max_{\theta_d} V(D,G) = \mathbb{E}_{X\sim p_{\mathrm{stroke}}(X)}\left[\log D(X;\theta_d)\right] + \mathbb{E}_{\mathbf{z}\sim p_{\mathrm{noise}}(\mathbf{z})}\left[\log(1-D(G(\mathbf{z};\theta_g);\theta_d))\right] \quad (1)$$

where $p_{\mathrm{noise}}(\mathbf{z})$ is a Gaussian noise distribution, $p_{\mathrm{stroke}}(X)$ is the distribution of the real positive stroke instances in source domain $\mathcal{S}_{\mathrm{ST}}^{(i)}$, and $\mathbb{E}_p[\cdot]$ represents the expectation w.r.t. specific distribution p (e.g., $p_{\mathrm{noise}}(\mathbf{z})$ or $p_{\mathrm{stroke}}(X)$. After the training of GAN model converges, the generator G can be exploited to generate stroke instances from hospital i to the target domain. As the generated instances do not correspond to any physical patients, no privacy leakage is of any concern.

B. Network Weight Transfer Using Chronic Disease Data

NWT module of HDTL-SRP is designed to incorporate data from source domains of other highly correlated chronic diseases, such as hypertension or diabetes, which tend to have more health records. In this work, the SRP model $\mathcal{M}$ is chosen to be an M-layered DNN where hidden variables in the i-th layer is specified as $\mathbf{h}_i = \phi(\mathbf{h}_{i-1}^T W_i + \mathbf{b}_i)$ where W_i and $\mathbf{b}_i$ represent the weight matrix and bias vector at i-th hidden layer, respectively. Here, $\mathbf{h}_0$ in the first layer is the vectorized form of X and $\phi(\cdot)$ is a non-linear activation function which can be chosen as the rectified linear function $ReLU(\mathbf{h}) \triangleq \min(0,\mathbf{h})$ or the hyperbolic tangent function $\tanh(\mathbf{h}) \triangleq (1-\exp(-2\mathbf{h}))/(1+\exp(-2\mathbf{h}))$ or the sigmoid function $\xi(\mathbf{h}) \triangleq 1/(1+\exp(-\mathbf{h}))$ (used in the output layer).

Then, a loss function L is specified as the cross-entropy between the predicted labels using $\mathcal{M}$ and true labels:

$$L(\mathcal{M}) \triangleq -\frac{1}{|\mathcal{S}|}\sum_{j=1}^{|\mathcal{S}|}\left[y_j\log(\widehat{y}_j) + (1-y_j)\log(1-\widehat{y}_j)\right]. \quad (2)$$

The DNN model can be trained in a specific source domain using either hypertension $\mathcal{S}_{\mathrm{HT}}$ or diabetes $\mathcal{S}_{\mathrm{DB}}$ data. Finally, given a parameter m, NWT transfers the network structure and weights of the first m layers to the target domain $\mathcal{T}_{\mathrm{ST}}$. The weights of the other layers (m-th to M-th) of SRP model will be first randomly initialized and then fine-tuned using stroke data. The above procedure is shown in Fig. 2.

C. Network Parameters Selection Using Bayesian Optimization

In network weight transfer approach, while multiple source domains are available, the parameters such as the number of transferred layer and the transferred sequence of different source

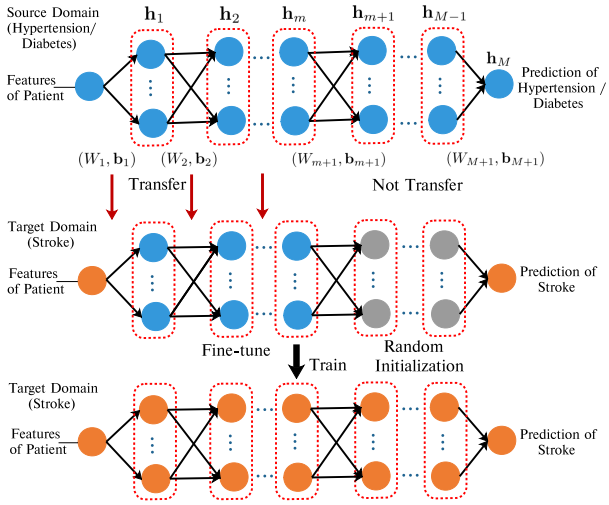

Fig. 2. Network Weight Transfer for DNN-based SRP model from the source domain of chronic disease to the target domain of stroke.

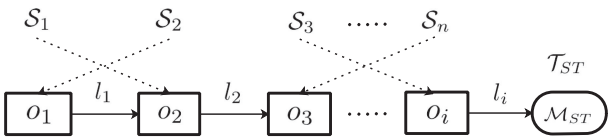

Fig. 3. Network Weight Transfer using multiple source domains.

domains are important factors of SRP model performance. To construct the best SRP model while n related source domains are available, as shown in Fig 3, we need to find the parameters that make the model performance the best, including the transferred layer number of i-th source domain $\mathcal{S}_i$ and the transferred sequence. The transferred layer number of i-th source domain $\mathcal{S}_i$ and the transferred sequence should be evaluated. Traditionally, to find the best parameters, all parameters need to be evaluated, which is time-consuming. To address this issue, Gaussian process-based Bayesian Optimization approach (BO) [47] is used to get the optimal parameters quickly. We use Levenshtein Distance (LD) [49] to get the similarity of different transfer sequence. Then, Multiple Dimensional Scaling (MDS) [50] algorithm is used to get low dimension search space for BO.

A candidate of NWT configuration $\Delta = (\mathbf{o}, \mathbf{l})$ is specified by both the transferring order $\mathbf{o}$ among multiple source domains and the number of transferred layers $\mathbf{l}$ between a source domain and its consecutive source domain. Transferring order of multiple source domains and the number of transferred layers are defined as $\mathbf{o} = (o_1, o_2, \ldots, o_i)$ and $\mathbf{l} = (l_1, l_2, \ldots, l_i)$, respectively, where $o_i \in \{1, \ldots, n\}$, $l_i \in \{1, \ldots, M\}$. Meanwhile, $i \leq n$ indicates at most n source domains are evaluated, and $o_i \neq o_j$ implies each source domain will be evaluated at most once. When transferring the network structure among multiple source domains by order, $o_i = k$ and $l_i = m$ indicates the k-th source domains will be in the i-th place (see Fig. 3) and the first m-th layers of k-th source domains will be transferred to its consecutive domain, respectively.

To find the best candidate for specifying a machine learning model, Genetic Algorithm [43], [44] and Bayesian Optimization [47] can be used if the input space is an Euclidean space. However, the NWT configuration Δ is in non-Euclidean space. We project the parameter of transferred sequence into Euclidean space by two steps. First, for any two transferring orders $\mathbf{o}$ and $\mathbf{o}'$, we compute the minimum edit distance $\mathrm{MED}(\mathbf{o}, \mathbf{o}')$ as the distance measure of these two sequences [49]. So, given N sequence instances, we can calculate the minimum edit distance matrix, where each element denotes the distance between two different sequence instances. Second, after the dimensionality reduction using MDS, each instance is reduced to a k-dimensional vector which represents its position in low dimensional space. In this work, we use BO to find the best sequence and transferred layer number. Here, we normalize o_i and l_i as 0 to 1. So the parameter need to be optimized is defined as $\mathbf{c} \in [0, 1]^{k+|\mathbf{n}|}$.

The parameter is evaluated using training data $\mathcal{D}_t$ and its performance is validated in validating data $\mathcal{D}_v$. We define the validation error on $\mathcal{D}_v$ as the output of the objective function $f(\cdot)$. Therefore, the best parameter can be represented as

$$\mathbf{c}^* = \underset{\mathbf{c} \in [0,1]^{k+|\mathbf{n}|}}{\arg\min} \; f(\mathbf{c} | \mathcal{D}_t, \mathcal{D}_v). \tag{3}$$

In this work, the objective function $f(\cdot)$ is modeled by a Gaussian process which can be fully specified by its mean function $m(\cdot)$ and kernel function $\kappa(\cdot, \cdot)$, $f(\cdot) \sim \mathcal{N}(m(\cdot), \kappa(\cdot, \cdot))$. For simplicity, we assume the mean function as $\mathbf{0}$. $\kappa(\cdot, \cdot)$ is the kernel function which can be chosen as Radial Basis Function $\kappa(\mathbf{c}, \mathbf{c}') = \exp(-\|\mathbf{c} - \mathbf{c}'\|^2 / (2\delta^2))$. Here, we split the candidates in two parts of evaluated set $S = \{\mathbf{c}_1, \mathbf{c}_2, \cdots\}$ and unevaluated set $S' = \{\mathbf{c}'_1, \mathbf{c}'_2, \cdots\}$. We define the covariance matrix as K_{SS}, $K_{SS'}$, $K_{S'S}$ and $K_{S'S'}$. For $|S| \times |S'|$ covariance matrix $K_{SS'}$, each element $[K_{SS'}]_{i,j}$ indicates the value of kernel function $\kappa(\mathbf{c}_i, \mathbf{c}'_j)$. The other three matrices are constructed in the same way. Therefore, given a set S' need to be evaluated, it is to output its corresponding prediction $f_{S'}$. The Gaussian process can be represented as

$$\begin{pmatrix} f_S \\ f_{S'} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} K_{SS} & K_{SS'} \\ K_{S'S} & K_{S'S'} \end{pmatrix} \right). \tag{4}$$

So, we can get the predictive distribution of $f(\mathbf{c}')$ which is a normal distribution with mean and variance as

$$\mu(\mathbf{c}') = K_{S'S} K_{SS}^{-1} y(\mathbf{c}) \tag{5}$$

and

$$\sigma(\mathbf{c}') = K_{S'S'} - K_{S'S} K_{SS}^{-1} K_{SS'}. \tag{6}$$

To trade off exploration of search space and exploitation of current promising areas, we need to make use of acquisition function [51]. We use the expected improvement (EI) [52] function as our acquisition function. The expectation can be

calculated as

$$\alpha_{EI}(\mathbf{c}') = (\mu(\mathbf{c}') - f(\mathbf{c}^*))\Phi\left(\frac{\mu(\mathbf{c}') - f(\mathbf{c}^*)}{\sigma(\mathbf{c}')}\right) + \sigma(\mathbf{c}')\phi\left(\frac{\mu(\mathbf{c}') - f(\mathbf{c}^*)}{\sigma(\mathbf{c}')}\right) \tag{7}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are cumulative distribution function and probability density function of a standard normal distribution, respectively. To find the next point to evaluate, we need to maximize the expectation as

$$\mathbf{c}' = \underset{\mathbf{c}\in[0,1]^{k+|\mathbf{n}|}}{\arg\max}\ \alpha_{EI}(\mathbf{c}). \tag{8}$$

Repeating the above step can achieve the effect of 3.

Given a $\mathbf{c}' = (c'_1, \ldots, c'_{k+|\mathbf{n}|})$, we project it to the closest NWT configuration $(\mathbf{o}', \mathbf{l}')$ specifically. For the transferred sequence, c'_1 to c'_k is used to compute the distance with each NWT configuration in normalized space. Then the nearest one is projected to its original space as $\mathbf{o}'$. Then, for c'_{k+1} to $c'_{k+|\mathbf{n}|}$, compute each c'_i with the value of transferred layer number in normalized space and project it to its original space as l'_i. Finally, we get the NWT configuration $(\mathbf{o}', \mathbf{l}')$ and evaluate its performance.

D. Active Instance Transfer and Model Training

To fine-tune DNN network transferred from NWT module in the target domain $\mathcal{T}_{\text{ST}}$, we still need to balance the positive and negative instances in stroke data. Thanks to the GIT module, sufficient and abundant generative instances have been assimilated in a candidate set $\mathcal{S}_{\text{ST}}$. It leads to the question: *which instances in $\mathcal{S}_{ST}$ should be selected to $\mathcal{T}_{ST}$?* To answer this question, AIT component of HDTL-SRP exploits an active learning strategy to select the instances that are the most informative for training the SRP model. Formally, the most informative instance X^* can be iteratively selected according to

$$X^* = \underset{X\in\mathcal{S}_{\text{ST}}}{\arg\max}\, \mathbb{H}(X) \tag{9}$$

where $\mathbb{H}(X)$ is the entropy of each generative instance $\mathbb{H}(X) = -\sum_{y\in\{+1,-1\}} p(y|X)\log p(y|X)$ and $p(y|X)$ can be evaluate using output layer (i.e., using σ function) of DNN. The selected instance will be put into the target domain to form gradually a balanced stroke data $\mathcal{T}'_{\text{ST}} = \mathcal{T}_{\text{ST}} \bigcup (X^*, +1)$. Finally, the stroke data $\mathcal{T}'_{\text{ST}}$ is exploited to train the SRP model.

As shown in Fig. 1, the overall HDTL-SRP framework can be deployed by running Algorithm 1 in hospitals with external sources of stroke data and running Algorithm 2 in the hospital hosting chronic disease data and target domain. The proposed methods will be empirically evaluated in the next section.

Algorithm 1: Generative Instance Transfer (GIT).

Input:
External stroke data $\mathcal{S}_{\text{ST}}$, No. of requested instances N, batch size I;
Output:
Synthetic stroke data $\mathcal{S}'_{\text{ST}}$;

1: **for** each training epoch **do**
2: Sample I real stroke instances $\{X_1, \ldots, X_I\}$ from $\mathcal{S}_{\text{ST}}$;
3: Update the discriminator by ascending its stochastic gradient w.r.t θ_d:

$$\bigtriangledown_{\theta_d} \frac{1}{I}\sum_{i=1}^{I}[\log D(X_i) + \log(1 - D(G(\mathbf{z})))]\,;$$

4: Update the generator by descending its stochastic gradient w.r.t θ_g:

$$\bigtriangledown_{\theta_g} \frac{1}{I}\sum_{i=1}^{I}\log(1 - D(G(\mathbf{z})));$$

5: **end for** until convergence
6: **return** $\mathcal{S}'_{\text{ST}} \leftarrow$ Generate N instances using G.

V. Experiments and Results

Dataset Description. The data used to test the proposed method is collected from EHR databases of three hospitals located at the same city.[1] Collaborating with medical doctors from the department of neurology, we select 23 attributes (e.g., basic health records, blood tests, see TABLE II) as the features of patients during Jan. 2012 and Dec. 2014. In addition, all the monitored patients have their diseases (i.e., hypertension, diabetes, and stroke) labeled from Jan. 2015 to Dec. 2017. As three hospitals have different ranks (III-A > II-A > II-B), the number of qualified health records is distributed unevenly (see Table III). We aim to address the challenging task of predicting stroke in lower-level hospital (II-B) which are constrained by small and imbalance stroke data, and there are useful data from other hospitals. As we can see, the stroke data owned by the top-ranked hospital is over ten times of the lowest-ranked hospital.

Experimental Settings. To verify the proposed method, we conduct experiments on both synthetic and real-world scenarios.

Synthetic Scenario: Due to a data-sharing agreement for research purpose among three hospitals, we are allowed to put all the health records together. To simulate flexibly higher or lower-level hospitals which could own different sizes of medical data, we conduct experiments in different settings but they offer similar results. For simplicity, we discuss one setting in the synthetic experiments. First, we randomly select 10 test sets among all the health records with stroke disease; each set consists of 100 (100) positive (negative) instances. The target domain $\mathcal{T}_{\text{ST}}$ is an imbalanced set that consists of 100 (1000) positive (negative) stroke instances. The source domain of chronic disease (i.e., $\mathcal{S}_{\text{HT}}$ and $\mathcal{S}_{\text{DB}}$) assumes to be balanced which is consistent with the

[1]The patient data were anonymised, so we had no knowledge of any names and we simply used anonymised datasets in our studies. The usage of data is exempted from IRB approval by Huazhong University of Science and Technology Union Shenzhen Hospital (aka. Nanshan Hospital), Shenzhen, China.

Algorithm 2 Network Wight Transfer coupled with Active Instance Transfer for SRP model training (NWT+BO+AIT)

Input:
Hypertension (Diabetes) data $\mathcal{S}_{\text{HT}}$ ($\mathcal{S}_{\text{DB}}$), stroke data in target domain $\mathcal{T}_{\text{ST}}$, layers of DNN model M, objective function f, No. of instances from each external source N, initial design $\mathbf{c}_{1:t}$, No. of iteration using BO to evaluated T;
Output:
DNN-based SRP model $\mathcal{M}$;

$\mathcal{S}_{\text{ST}} \leftarrow$ Request N instances by invoking GIT (see Algorithm 1) in each external hospital;
$\tau \leftarrow$ difference between no. of negative and no. of positive instances in $\mathcal{T}_{\text{ST}}$;
for i in $\{1, \ldots, t\}$ **do**
  Map transferred setting according to $\mathbf{c_i}$ and then construct corresponding model $\mathcal{M}_i$;
  for j in $\{1, \ldots, \tau\}$ **do**
    $X \leftarrow$ actively select one generative instance from $\mathcal{S}_{\text{ST}}$;
    $\mathcal{T}'_{\text{ST}} \leftarrow \mathcal{T}_{\text{ST}} \bigcup (X, +1)$;
    Train and update $\mathcal{M}_i$ using $\mathcal{T}'_{\text{ST}}$;
  end for
  $f_i \leftarrow$ validate $\mathcal{M}_i$ using validating set;
  Update Gaussian process model;
end for
for k in $\{t+1, \ldots, T\}$ **do**
  Select next parameter $\mathbf{c}_k = \arg\max_{\mathbf{c} \in [0,1]^{k+|\mathbf{n}|}} \alpha_{EI}(\mathbf{c})$;
  Map transferred setting according to $\mathbf{c_k}$ and then construct corresponding model $\mathcal{M}_k$;
  for j in $\{1, \ldots, \tau\}$ **do**
    $X \leftarrow$ actively select one generative instance from $\mathcal{S}_{\text{ST}}$;
    $\mathcal{T}'_{\text{ST}} \leftarrow \mathcal{T}_{\text{ST}} \bigcup (X, +1)$;
    Train and update $\mathcal{M}_k$ using $\mathcal{T}'_{\text{ST}}$;
  end for
  $f_k \leftarrow$ validate $\mathcal{M}_k$ using validating set;
  Update Gaussian process model;
end for
return $\mathbf{c}_{best} = \arg\min_{\mathbf{c} \in \{\mathbf{c}_1, \ldots, \mathbf{c}_T\}} f(\mathbf{c})$

real-world statistic (see Table III). The positive and negative instances are both 10 000. For the source domain from external stoke data $\mathcal{S}_{\text{ST}}$, the positive and negative instances are chosen to be 1000 and 10 000, respectively.

Real-world Scenario: We assume that stroke risk prediction in the II-B (i.e., the lowest-ranked) hospital is in the target domain $\mathcal{T}_{\text{ST}}^{(\text{II-B})}$; hypertension/diabetes data in II-B hospital ($\mathcal{S}_{\text{HT}}^{(\text{II-B})}$ / $\mathcal{S}_{\text{DB}}^{(\text{II-B})}$) and external stroke data from II-A and III-A hospitals ($\mathcal{S}_{\text{ST}}^{(\text{II-A})}$ and $\mathcal{S}_{\text{ST}}^{(\text{III-A})}$) are some useful sources. The datasets are divided randomly with a ratio of 8:1:1 for training, validating, and testing respectively.

The SRP model is designed as a 9-layered DNN. The first 8 layers have 12 neurons each and the last layer has 1 neuron using cross-entropy as loss function. The optimizer is Adam with learning rate 0.001. Then, the proposed HDTL-SRP framework is tested in the above two scenarios. The results are averaged over 10 random settings.

TABLE II
THE 23 ATTRIBUTES AFTER PREPROCESSING

Attribute	Mean	Range
Age	51.40	27-89
Number of neutrophils	4.65	0.05-51.36
Number of lymphocytes	2.02	0.11-52.5
Number of eosinophils	0.16	0.01-6.35
Number of basophils	0.01	0.00-1.60
Total protein	69.15	27.30-124.10
Albumin	42.60	11.60-59.60
Globulin	26.54	11.20-78.10
Total bilirubin	11.18	0.50-291.40
Direct bilirubin	3.13	0.08-161.70
Potassium	4.05	2.10-19.50
Sodium	141.33	96.00-164.60
Calcium	2.31	1.24-4.37
Urea nitrogen	5.27	0.20-64.00
AST/ALT	1.14	0.10-19.00
Triglyceride	1.98	0.02-65.44
High density lipoprotein	1.30	0.14-4.73
Low density lipoprotein	2.95	0.01-15.53
Average red blood cell	92.17	48.60-133.60
Thrombin time	12.73	0.47-66.10
International normalized ratio	0.99	0.30-8.18
Activated partial thromboplastin time	32.96	13.70-119.00
Fibrinogen	3.65	0.69-11.04

TABLE III
DATASETS COLLECTED BY THREE COLLABORATING HOSPITALS

Disease	Rank	Domain	Positive	Negative
Stroke	II-B	Target	128	2159
Hypertension	II-B	Source	529	469
Diabetes	II-B	Source	1638	1056
Stroke	II-A	Source	415	6477
Hypertension	II-A	Source	3502	3752
Diabetes	II-A	Source	5168	3318
Stroke	III-A	Source	1883	22207
Hypertension	III-A	Source	13769	11414
Diabetes	III-A	Source	12196	10711

Comparison. We compare the proposed HDTL-SRP with existing SRP methods, such as SVM [5], decision tree (DT) [5], random forest (RF) [5] and DNN [8]. To justify that the AIT component can effectively work with imbalanced stroke data, we also compare with existing SRP models coupled with oversampled algorithm called SMOTE [34] which is often used to address issue with imbalanced labels.

Performance Metric. Given the prediction and the ground truth, we can compute true positive (TP), true negative (TN), false positive (FP) and false negative (FN) for the test set. Then, the performance of the proposed approach is evaluated using four metrics: (1) Accuracy (i.e., (TP+TN)/(TP+TN+FP+FN)), (2) Recall (i.e., TP/(TP+FN)), (3) F1 score, and (4) Area Under the ROC Curve (AUC) [7]. For all the above metrics, the values 0 and 1 represent the worst and the best performance, respectively. To evaluate the efficiency of BO, we use validation error [51] which is the classification error in validating set.

TABLE IV
PERFORMANCE OF NETWORK WEIGHT TRANSFER LEARNING VERSUS NO TRANSFER (IMBALANCED STROKE DATA)

SRP Method	Accuracy	Recall	F1 Score	AUC
SVM (No Transfer)	0.695 ± 0.020	0.385 ± 0.007	0.552 ± 0.005	0.701 ± 0.010
DT (No Transfer)	0.712 ± 0.015	0.463 ± 0.017	0.602 ± 0.019	0.711 ± 0.015
RF (No Transfer)	0.675 ± 0.012	0.332 ± 0.024	0.504 ± 0.027	0.685 ± 0.010
DNN (No Transfer)	0.719 ± 0.016	0.468 ± 0.029	0.611 ± 0.026	0.776 ± 0.013
DNN+Weight-sharing ($\mathcal{S}_{\text{HT}}$)	0.482 ± 0.013	0.294 ± 0.012	0.361 ± 0.014	0.482 ± 0.011
DNN+Weight-sharing ($\mathcal{S}_{\text{DB}}$)	0.487 ± 0.018	0.416 ± 0.017	0.446 ± 0.014	0.467 ± 0.011
DNN+Pre-train-fine-tuning ($\mathcal{S}_{\text{HT}}$)	0.700 ± 0.012	0.463 ± 0.012	0.607 ± 0.016	0.763 ± 0.020
DNN+Pre-train-fine-tuning ($\mathcal{S}_{\text{DB}}$)	0.688 ± 0.018	0.455 ± 0.019	0.593 ± 0.018	0.740 ± 0.012
DNN+NWT ($\mathcal{S}_{\text{HT}}$)	0.725 ± 0.024	0.489 ± 0.041	$\mathbf{0.625 \pm 0.021}$	0.771 ± 0.029
DNN+NWT ($\mathcal{S}_{\text{DB}}$)	$\mathbf{0.729 \pm 0.019}$	$\mathbf{0.491 \pm 0.026}$	0.625 ± 0.026	$\mathbf{0.781 \pm 0.027}$

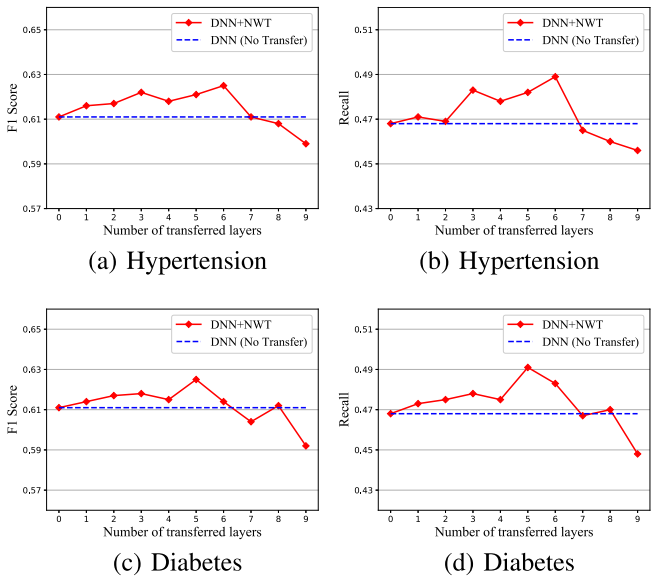

Fig. 4. Performance of DNN-based SRP models (Imbalanced stroke data in target domain).

A. Synthetic Experiments

Performance of Network Weight Transfer: We first conduct experiments in the synthetic scenario to test the effectiveness of NWT module which aims to transfer knowledge structure in chronic disease data to stroke domain. The DNN-based SRP model is trained using either $\mathcal{S}_{\text{HT}}$ or $\mathcal{S}_{\text{DB}}$ and transferred the first m layers to the target domain for fine-tuning using $\mathcal{T}_{\text{ST}}$. As the results shown in Table IV, a combination of DNN and NWT (DNN+NWT) outperforms those SRP methods which only rely on stroke data. The results indicate the NWT can effectively transfer knowledge in the chronic disease data to stroke domain. We also compare the proposed DNN+NWT with transer learning basedlines (i.e., weight-sharing and fine-tuning of a pre-trained full model). The results show that the proposed DNN+NWT method still outperforms the baselines in both scenarios of transferring knowledge from hypertension and diabetes data to the stroke prediction task.

We further investigate how the number m of transferred layers affect the performance of SRP model. As shown in Fig. 4, the performance of SRP improves when m increases at the beginning and then declines at certain point ($m = 6$ in Fig. 4 a-b, $m = 5$ in Fig. 4 c-d). When $m = 9$, the performance of DNN+NWT is even worse than DNN with no transferred knowledge. This is because the hidden features of layers in deep neural network tend to transit from general to specific [23]; the transferred hidden features become too specific to be adapted using $\mathcal{T}_{\text{ST}}$ when m is too large.

Performance of Hybrid Transfer: In Table V, we evaluate the performance of the proposed HDTL-SRP model which can simultaneously exploit data from both chronic disease and external stroke data; the performance of HDTL-SRP is compared with that of existing SRP models coupled with oversampling or generative techniques. We can see that the overall performance of HDTL-SRP is better than a combination of DNN and AIT (DNN+AIT) (no network weight transfer from chronic disease). This results again indicate that NWT component of HDTL-SRP can effectively transfer knowledge from chronic disease data to target domain for improving stroke prediction. We also compare the performance of transferring real-world stroke data (DNN+Real-world) and transferring randomly selected stroke data from generated data using GAN (DNN+GAN). As we can see, compared with the method directly using the real-world stroke data, the performance of those methods using synthetic data (i.e., DNN+SMOTE, DNN+GAN, DNN+AIT) degrade to some extent. This could be the price for preserving the data privacy. Interestingly, the performance of active selection method (DNN+AIT) is only slightly different from that of DNN+Real-world. This indicates that the AIT component can select more informative data points from the source domain. In addition, we can observe that DNN coupled with AIT outperforms DNN coupled with SMOTE. This indicates that the AIT component of HDTL-SRP can better transfer instances from external stroke data to target stroke domain.

The performance of HDTL-SRP is also evaluated by varying the number m of transferred layers (See Fig. 5). Similar to Fig. 4, due to the representation specificity, the performance of HDTL-SRP drops when $m = 9$; however, the performance decline is not as drastic. In addition, HDTL-SRP consistently outperforms DNN+AIT (no transfer from chronic disease domain). This implies that the NWT component can perform better when the target domain has balanced stroke data.

B. Real-World Experiments

In this experiment, HDTL-SRP framework is tested in a more realistic scenario. Among three collaborating hospitals

TABLE V
PERFORMANCE OF HDTL-SRP (BALANCING STROKE DATA IN TARGET DOMAIN)

SRP Method	Accuracy	Recall	F1 Score	AUC
SVM+SMOTE	0.731 ± 0.021	0.545 ± 0.002	0.658 ± 0.020	0.809 ± 0.021
DT+SMOTE	0.707 ± 0.021	0.504 ± 0.025	0.642 ± 0.026	0.767 ± 0.021
RF+SMOTE	0.715 ± 0.018	0.495 ± 0.030	0.645 ± 0.029	0.755 ± 0.017
DNN+Real-world($\mathcal{S}_{\mathrm{ST}}$)	0.739 ± 0.012	0.652 ± 0.017	0.725 ± 0.035	0.817 ± 0.013
DNN+SMOTE	0.737 ± 0.018	0.530 ± 0.044	0.667 ± 0.033	0.818 ± 0.028
DNN+GAN($\mathcal{S}_{\mathrm{ST}}$)	0.725 ± 0.027	0.612 ± 0.030	0.691 ± 0.031	0.798 ± 0.027
DNN+AIT ($\mathcal{S}_{\mathrm{ST}}$)	0.745 ± 0.023	0.630 ± 0.020	0.716 ± 0.025	0.819 ± 0.025
HDTL-SRP ($\mathcal{S}_{\mathrm{HT}}$,$\mathcal{S}_{\mathrm{ST}}$)	0.747 ± 0.032	0.712 ± 0.045	$\mathbf{0.757 \pm 0.035}$	0.825 ± 0.035
HDTL-SRP ($\mathcal{S}_{\mathrm{DB}}$,$\mathcal{S}_{\mathrm{ST}}$)	$\mathbf{0.757 \pm 0.032}$	$\mathbf{0.715 \pm 0.039}$	0.749 ± 0.034	$\mathbf{0.834 \pm 0.038}$

TABLE VI
PERFORMANCE OF HDTL-SRP FRAMEWORK IN REAL-WORLD EXPERIMENTS

Test	Source Domain	Approach	Accuracy	Recall	F1 Score	AUC
#1	N.A.	SVM+SMOTE	0.715 ± 0.023	0.532 ± 0.038	0.653 ± 0.026	0.740 ± 0.019
#2	N.A.	DT+SMOTE	0.693 ± 0.021	0.513 ± 0.047	0.634 ± 0.028	0.721 ± 0.024
#3	N.A.	RF+SMOTE	0.675 ± 0.011	0.475 ± 0.026	0.622 ± 0.024	0.735 ± 0.013
#4	N.A.	DNN+SMOTE	0.717 ± 0.012	0.510 ± 0.024	0.667 ± 0.033	0.783 ± 0.014
#5	N.A.	No Transfer	0.628 ± 0.017	0.425 ± 0.027	0.565 ± 0.027	0.695 ± 0.019
#6	$\mathcal{S}_{\mathrm{HT}}^{(\text{II-B})}$	Weight	0.711 ± 0.016	0.461 ± 0.033	0.612 ± 0.029	0.734 ± 0.032
#7	$\mathcal{S}_{\mathrm{DB}}^{(\text{II-B})}$	Weight	0.704 ± 0.018	0.463 ± 0.034	0.611 ± 0.031	0.741 ± 0.031
#8	$\mathcal{S}_{\mathrm{ST}}^{(\text{II-A})}$	Instance	0.707 ± 0.027	0.521 ± 0.051	0.639 ± 0.042	0.753 ± 0.024
#9	$\mathcal{S}_{\mathrm{ST}}^{(\text{III-A})}$	Instance	0.736 ± 0.030	0.570 ± 0.045	0.683 ± 0.041	0.763 ± 0.031
#10	$\mathcal{S}_{\mathrm{ST}}^{(\text{II-A})}$, $\mathcal{S}_{\mathrm{ST}}^{(\text{III-A})}$	Instance	0.754 ± 0.027	0.630 ± 0.047	0.718 ± 0.036	0.788 ± 0.027
#11	$\mathcal{S}_{\mathrm{HT}}^{(\text{II-B})}$, $\mathcal{S}_{\mathrm{ST}}^{(\text{II-A})}$	Hybrid	0.745 ± 0.023	0.586 ± 0.037	0.693 ± 0.031	0.790 ± 0.031
#12	$\mathcal{S}_{\mathrm{DB}}^{(\text{II-B})}$, $\mathcal{S}_{\mathrm{ST}}^{(\text{II-A})}$	Hybrid	0.751 ± 0.026	0.609 ± 0.033	0.706 ± 0.031	0.803 ± 0.033
#13	$\mathcal{S}_{\mathrm{HT}}^{(\text{II-B})}$, $\mathcal{S}_{\mathrm{ST}}^{(\text{III-A})}$	Hybrid	0.774 ± 0.026	0.637 ± 0.038	0.734 ± 0.032	0.804 ± 0.033
#14	$\mathcal{S}_{\mathrm{DB}}^{(\text{II-B})}$, $\mathcal{S}_{\mathrm{ST}}^{(\text{III-A})}$	Hybrid	0.774 ± 0.031	0.628 ± 0.029	0.731 ± 0.022	0.811 ± 0.038
#15	$\mathcal{S}_{\mathrm{HT}}^{(\text{II-B})}$, $\mathcal{S}_{\mathrm{ST}}^{(\text{II-A})}$, $\mathcal{S}_{\mathrm{ST}}^{(\text{III-A})}$	Hybrid	0.778 ± 0.023	0.652 ± 0.047	0.756 ± 0.032	0.816 ± 0.030
#16	$\mathcal{S}_{\mathrm{DB}}^{(\text{II-B})}$, $\mathcal{S}_{\mathrm{ST}}^{(\text{II-A})}$, $\mathcal{S}_{\mathrm{ST}}^{(\text{III-A})}$	Hybrid	0.782 ± 0.025	0.669 ± 0.041	0.757 ± 0.031	0.821 ± 0.031
#17	$\mathcal{S}_{\mathrm{HT}}^{(\text{II-B})}$, $\mathcal{S}_{\mathrm{HT}}^{(\text{II-A})}$, $\mathcal{S}_{\mathrm{HT}}^{(\text{III-A})}$, $\mathcal{S}_{\mathrm{ST}}^{(\text{II-A})}$, $\mathcal{S}_{\mathrm{ST}}^{(\text{III-A})}$	Hybrid	$\mathbf{0.787 \pm 0.017}$	0.672 ± 0.024	0.761 ± 0.031	0.835 ± 0.029
#18	$\mathcal{S}_{\mathrm{DB}}^{(\text{II-B})}$, $\mathcal{S}_{\mathrm{DB}}^{(\text{II-A})}$, $\mathcal{S}_{\mathrm{DB}}^{(\text{III-A})}$, $\mathcal{S}_{\mathrm{ST}}^{(\text{II-A})}$, $\mathcal{S}_{\mathrm{ST}}^{(\text{III-A})}$	Hybrid	0.784 ± 0.038	$\mathbf{0.678 \pm 0.033}$	$\mathbf{0.772 \pm 0.021}$	$\mathbf{0.844 \pm 0.026}$

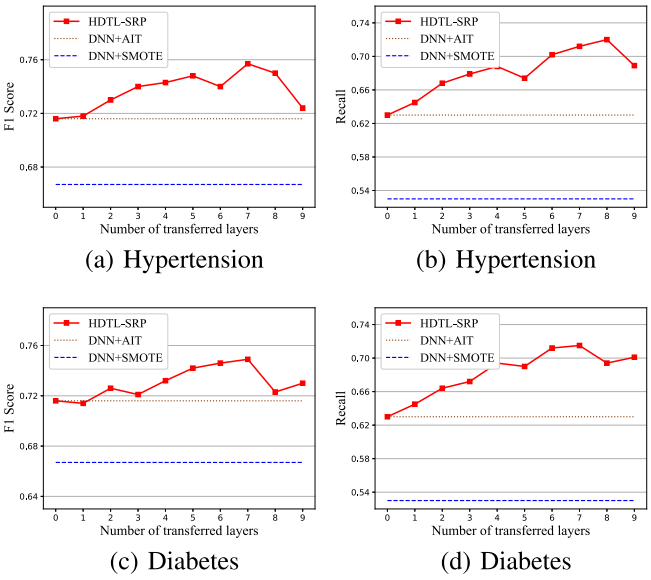

Fig. 5. Performance of DNN-based SRP model (Balancing stroke data in target domain).

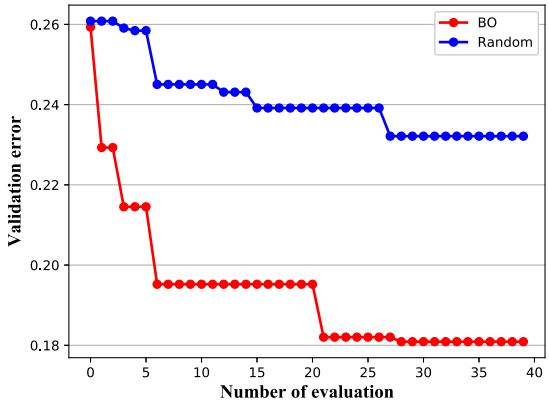

Fig. 6. Performance of Bayesian Optimization.

(see Table III), the higher-ranked III-A and II-A hospitals host source domains of stroke $\mathcal{S}_{\mathrm{ST}}^{(\text{III-A})}$ and $\mathcal{S}_{\mathrm{ST}}^{(\text{II-A})}$, respectively; the lower-ranked II-B hospital hosts target domain of stroke $\mathcal{T}_{\mathrm{ST}}^{(\text{II-B})}$, source domain of chronic disease: hypertension $\mathcal{S}_{\mathrm{HT}}^{(\text{II-B})}$ and diabetes $\mathcal{S}_{\mathrm{DB}}^{(\text{II-B})}$. Then, Algorithm 1 is deployed in III-A and II-A hospitals and Algorithm 2 is deployed in the II-B hospital. Based on HDTL-SRP framework, we test all possible transfer approaches using different combinations of source domains for training SRP model in the target domain.

The results are shown in Table VI. As we can see, although the weight-based tests (#6 and #7) can exploit other useful sources, their performances are worse than baselines (#1-#4). This is because the dataset of the former is extremely imbalanced. The method with no transfer has the poorest performance (see #5)

TABLE VII
THE TEST PERFORMANCE OF EACH EVALUATION OF BO

Test	Structure	Accuracy	Recall	F1 Score	AUC
#1	None	0.741 ± 0.027	0.621 ± 0.021	0.703 ± 0.015	0.771 ± 0.024
#2	$\mathcal{S}_{\mathrm{HT}} \xrightarrow{5} \mathcal{T}_{\mathrm{ST}}$	0.771 ± 0.012	0.649 ± 0.015	0.741 ± 0.015	0.805 ± 0.022
#3	None	0.745 ± 0.028	0.625 ± 0.022	0.708 ± 0.026	0.776 ± 0.015
#4	$\mathcal{S}_{\mathrm{HT}} \xrightarrow{6} \mathcal{S}_{\mathrm{DB}} \xrightarrow{4} \mathcal{T}_{\mathrm{ST}}$	0.785 ± 0.030	0.663 ± 0.022	0.759 ± 0.024	0.822 ± 0.015
#5	None	0.755 ± 0.014	0.634 ± 0.017	0.721 ± 0.018	0.787 ± 0.032
#6	$\mathcal{S}_{\mathrm{HT}} \xrightarrow{8} \mathcal{T}_{\mathrm{ST}}$	0.785 ± 0.031	0.663 ± 0.012	0.759 ± 0.017	0.822 ± 0.022
#7	$\mathcal{S}_{\mathrm{DB}} \xrightarrow{7} \mathcal{S}_{\mathrm{HT}} \xrightarrow{1} \mathcal{T}_{\mathrm{ST}}$	0.805 ± 0.017	0.681 ± 0.030	0.783 ± 0.033	0.844 ± 0.014
#8	$\mathcal{S}_{\mathrm{HT}} \xrightarrow{7} \mathcal{S}_{\mathrm{DB}} \xrightarrow{6} \mathcal{T}_{\mathrm{ST}}$	0.782 ± 0.018	0.660 ± 0.026	0.755 ± 0.013	0.818 ± 0.032
#9	$\mathcal{S}_{\mathrm{DB}} \xrightarrow{4} \mathcal{T}_{\mathrm{ST}}$	0.787 ± 0.021	0.665 ± 0.022	0.761 ± 0.023	0.824 ± 0.014
#10	$\mathcal{S}_{\mathrm{HT}} \xrightarrow{1} \mathcal{T}_{\mathrm{ST}}$	0.773 ± 0.022	0.651 ± 0.011	0.743 ± 0.032	0.808 ± 0.012
#11	None	0.752 ± 0.014	0.632 ± 0.026	0.717 ± 0.026	0.784 ± 0.024
#12	$\mathcal{S}_{\mathrm{DB}} \xrightarrow{6} \mathcal{S}_{\mathrm{HT}} \xrightarrow{2} \mathcal{T}_{\mathrm{ST}}$	0.788 ± 0.018	0.665 ± 0.018	0.762 ± 0.025	0.825 ± 0.031
#13	$\mathcal{S}_{\mathrm{HT}} \xrightarrow{7} \mathcal{S}_{\mathrm{DB}} \xrightarrow{4} \mathcal{T}_{\mathrm{ST}}$	0.801 ± 0.031	0.676 ± 0.026	0.777 ± 0.011	0.838 ± 0.033
#14	$\mathcal{S}_{\mathrm{HT}} \xrightarrow{8} \mathcal{S}_{\mathrm{DB}} \xrightarrow{3} \mathcal{T}_{\mathrm{ST}}$	0.805 ± 0.020	0.677 ± 0.018	0.778 ± 0.022	0.839 ± 0.030
#15	$\mathcal{S}_{\mathrm{HT}} \xrightarrow{7} \mathcal{S}_{\mathrm{DB}} \xrightarrow{1} \mathcal{T}_{\mathrm{ST}}$	0.797 ± 0.021	0.674 ± 0.019	0.774 ± 0.013	0.836 ± 0.015
#16	$\mathcal{S}_{\mathrm{DB}} \xrightarrow{5} \mathcal{S}_{\mathrm{HT}} \xrightarrow{4} \mathcal{T}_{\mathrm{ST}}$	0.804 ± 0.032	0.680 ± 0.013	0.782 ± 0.020	0.843 ± 0.011
#17	$\mathcal{S}_{\mathrm{DB}} \xrightarrow{3} \mathcal{S}_{\mathrm{HT}} \xrightarrow{2} \mathcal{T}_{\mathrm{ST}}$	0.781 ± 0.026	0.659 ± 0.029	0.754 ± 0.031	0.817 ± 0.024
#18	$\mathcal{S}_{\mathrm{DB}} \xrightarrow{4} \mathcal{T}_{\mathrm{ST}}$	0.785 ± 0.030	0.662 ± 0.012	0.758 ± 0.027	0.821 ± 0.025
#19	$\mathcal{S}_{\mathrm{HT}} \xrightarrow{2} \mathcal{T}_{\mathrm{ST}}$	0.773 ± 0.015	0.651 ± 0.015	0.743 ± 0.032	0.808 ± 0.014
#20	$\mathcal{S}_{\mathrm{DB}} \xrightarrow{6} \mathcal{S}_{\mathrm{HT}} \xrightarrow{5} \mathcal{T}_{\mathrm{ST}}$	0.793 ± 0.028	0.670 ± 0.014	0.768 ± 0.030	0.831 ± 0.019
#21	None	0.756 ± 0.015	0.635 ± 0.020	0.722 ± 0.030	0.788 ± 0.025
#22	$\mathcal{S}_{\mathrm{HT}} \xrightarrow{7} \mathcal{S}_{\mathrm{DB}} \xrightarrow{4} \mathcal{T}_{\mathrm{ST}}$	0.812 ± 0.027	0.684 ± 0.029	0.796 ± 0.030	0.854 ± 0.025
#23	$\mathcal{S}_{\mathrm{DB}} \xrightarrow{7} \mathcal{T}_{\mathrm{ST}}$	0.794 ± 0.014	0.671 ± 0.02	0.770 ± 0.032	0.832 ± 0.022
#24	$\mathcal{S}_{\mathrm{DB}} \xrightarrow{7} \mathcal{S}_{\mathrm{HT}} \xrightarrow{2} \mathcal{T}_{\mathrm{ST}}$	0.795 ± 0.012	0.672 ± 0.022	0.771 ± 0.025	0.833 ± 0.028
#25	$\mathcal{S}_{\mathrm{DB}} \xrightarrow{6} \mathcal{S}_{\mathrm{HT}} \xrightarrow{4} \mathcal{T}_{\mathrm{ST}}$	0.811 ± 0.018	0.687 ± 0.029	0.791 ± 0.015	0.851 ± 0.012
#26	$\mathcal{S}_{\mathrm{DB}} \xrightarrow{7} \mathcal{S}_{\mathrm{HT}} \xrightarrow{5} \mathcal{T}_{\mathrm{ST}}$	0.799 ± 0.028	0.676 ± 0.023	0.776 ± 0.032	0.838 ± 0.027
#27	$\mathcal{S}_{\mathrm{HT}} \xrightarrow{5} \mathcal{S}_{\mathrm{DB}} \xrightarrow{2} \mathcal{T}_{\mathrm{ST}}$	0.790 ± 0.024	0.667 ± 0.019	0.764 ± 0.016	0.827 ± 0.024
#28	$\mathcal{S}_{\mathrm{DB}} \xrightarrow{6} \mathcal{S}_{\mathrm{HT}} \xrightarrow{2} \mathcal{T}_{\mathrm{ST}}$	0.798 ± 0.023	0.674 ± 0.011	0.774 ± 0.017	0.836 ± 0.022
#29	$\mathcal{S}_{\mathrm{HT}} \xrightarrow{3} \mathcal{S}_{\mathrm{DB}} \xrightarrow{4} \mathcal{T}_{\mathrm{ST}}$	$\mathbf{0.819 \pm 0.019}$	$\mathbf{0.695 \pm 0.024}$	$\mathbf{0.801 \pm 0.023}$	$\mathbf{0.861 \pm 0.024}$
#30	$\mathcal{S}_{\mathrm{HT}} \xrightarrow{6} \mathcal{S}_{\mathrm{DB}} \xrightarrow{8} \mathcal{T}_{\mathrm{ST}}$	0.805 ± 0.019	0.682 ± 0.013	0.784 ± 0.014	0.845 ± 0.012
#31	$\mathcal{S}_{\mathrm{DB}} \xrightarrow{4} \mathcal{T}_{\mathrm{ST}}$	0.785 ± 0.028	0.663 ± 0.020	0.759 ± 0.028	0.822 ± 0.012
#32	$\mathcal{S}_{\mathrm{HT}} \xrightarrow{2} \mathcal{T}_{\mathrm{ST}}$	0.774 ± 0.016	0.652 ± 0.032	0.745 ± 0.033	0.809 ± 0.020
#33	$\mathcal{S}_{\mathrm{HT}} \xrightarrow{2} \mathcal{T}_{\mathrm{ST}}$	0.763 ± 0.015	0.641 ± 0.024	0.731 ± 0.030	0.796 ± 0.031
#34	$\mathcal{S}_{\mathrm{HT}} \xrightarrow{2} \mathcal{S}_{\mathrm{DB}} \xrightarrow{7} \mathcal{T}_{\mathrm{ST}}$	0.812 ± 0.030	0.688 ± 0.025	0.793 ± 0.031	0.853 ± 0.022
#35	$\mathcal{S}_{\mathrm{DB}} \xrightarrow{6} \mathcal{S}_{\mathrm{HT}} \xrightarrow{6} \mathcal{T}_{\mathrm{ST}}$	0.798 ± 0.015	0.675 ± 0.025	0.775 ± 0.019	0.837 ± 0.027
#36	$\mathcal{S}_{\mathrm{HT}} \xrightarrow{5} \mathcal{S}_{\mathrm{DB}} \xrightarrow{4} \mathcal{T}_{\mathrm{ST}}$	0.803 ± 0.016	0.680 ± 0.024	0.781 ± 0.024	0.842 ± 0.017
#37	$\mathcal{S}_{\mathrm{HT}} \xrightarrow{9} \mathcal{S}_{\mathrm{DB}} \xrightarrow{4} \mathcal{T}_{\mathrm{ST}}$	0.793 ± 0.013	0.670 ± 0.031	0.768 ± 0.029	0.831 ± 0.024
#38	$\mathcal{S}_{\mathrm{DB}} \xrightarrow{2} \mathcal{S}_{\mathrm{HT}} \xrightarrow{2} \mathcal{T}_{\mathrm{ST}}$	0.792 ± 0.030	0.669 ± 0.013	0.767 ± 0.018	0.829 ± 0.016
#39	$\mathcal{S}_{\mathrm{HT}} \xrightarrow{7} \mathcal{S}_{\mathrm{DB}} \xrightarrow{3} \mathcal{T}_{\mathrm{ST}}$	0.795 ± 0.018	0.672 ± 0.013	0.771 ± 0.025	0.833 ± 0.033
#40	$\mathcal{S}_{\mathrm{DB}} \xrightarrow{4} \mathcal{S}_{\mathrm{HT}} \xrightarrow{2} \mathcal{T}_{\mathrm{ST}}$	0.806 ± 0.026	0.682 ± 0.032	0.785 ± 0.022	0.846 ± 0.017

compared to other transferred methods as expected. It can also be observed that the instance-based tests (#8-#10) outperform the weight-based tests (#6-#7). This is because the external stroke data is more correlated to the target domain than the chronic disease data. In addition, results of tests #8-#10 indicate that the performance of SRP can be effectively improved when more external stroke data is used. The overall performance of hybrid transfer tests (#11-#16) is better than the single transfer approaches (#6-#11); among all the hybrid transfer tests, the performance of #17 and #18 outperform the rest. This implies that more source domains can achieve better performance. To sum up, HDTL-SRP framework can be practically deployed in the real-world scenario for effective performance of SRP tasks.

C. Network Structure Optimization in Multiple Sources Using Bayesian Optimization

To get the best transferred structure, we conduct experiment using Bayesian Optimization. As shown in Fig. 6, compared with selecting parameter randomly, BO can get lower validation error in the same evaluation number. To show clearly the procedure of BO, we depict each evaluation in Fig. 7 and TABLE VII. In Fig. 7, the number represents the optimized order of a candidate.

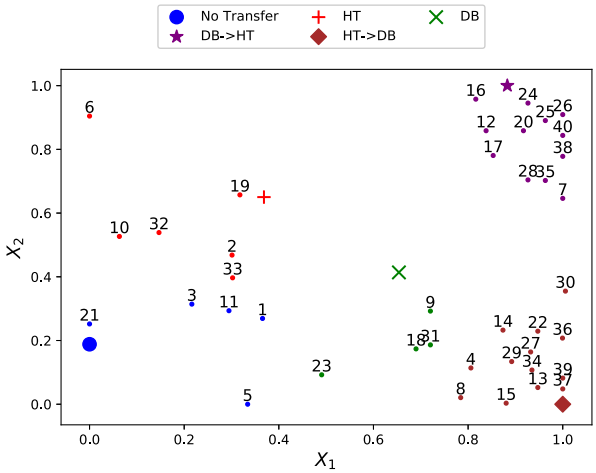

Fig. 7. Each evaluation and sequence of Bayesian Optimization.

We can see that during about the first 15 evaluations, the points are distributed in each candidate. Later bigger numbers tend to gather around upper and lower right corner (diamond and star). This indicates that BO explores the performance in the whole space in the beginning and then exploits in promising areas (candidates of multiple sources). In this experiment, $\mathcal{S}_{\text{HT}} \xrightarrow{i} \mathcal{S}_{\text{DB}} \xrightarrow{j} \mathcal{T}_{\text{ST}}$ represents the weight of first i layers of $\mathcal{S}_{\text{HT}}$ is transferred to $\mathcal{S}_{\text{DB}}$. After training the model in $\mathcal{S}_{\text{DB}}$, the weight of the first j layers of $\mathcal{S}_{\text{DB}}$ is transferred to $\mathcal{T}_{\text{ST}}$. TABLE VII records each evaluation of BO and its performance in testing set. As we can see, the 29-th evaluation $\mathcal{S}_{\text{HT}} \xrightarrow{3} \mathcal{S}_{\text{DB}} \xrightarrow{4} \mathcal{T}_{\text{ST}}$ achieves the best performance. Therefore, BO is able to find better models within same computation time.

VI. Conclusion

This work has addressed the issues of SRP with small and imbalanced stroke data. We have proposed a novel Hybrid Deep Transfer Learning-based Stroke Risk Prediction (HDTL-SRP) framework which consists of three key components: (1) Generative Instance Transfer (GIT) for making use of the external stroke data distribution among multiple hospitals while preserving the privacy, (2) Network Weight Transfer (NWT) for making use of data from highly correlated diseases (i.e., hypertension or diabetes), (3) Active Instance Transfer (AIT) for balancing the stroke data with the most informative generated instances. It is found that the proposed HDTL-SRP framework outperforms the state-of-the-art SRP models in both synthetic and real-world scenarios.

There are still several open questions for future work: how to (1) extend NWT to consider simultaneously multiple chronic diseases, (2) learn the optimized number of layers to be transferred automatically, (3) implement the system of other diseases as the health-care data share similar characteristics (i.e., small and imbalanced), and (4) improve the interpretability, which is a critical feature in health-care applications [53], of the SRP model as the interpretable mechanism could reveal the important knowledge structures to transfer.

References

[1] E. J. Benjamin, M. J. Blaha, and S. E. Chiuve, "Heart disease and stroke statistics—2017 update a report from the American Heart Association," *Circulation*, vol. 135, no. 10, pp. e146—e603, 2017.

[2] S. Koton *et al.*, "Stroke incidence and mortality trends in US communities, 1987 to 2011," *JAMA*, vol. 312, no. 3, pp. 259–268, 2014.

[3] E. J. Benjamin, P. Muntner, and M. S. Bittencourt, "Heart disease and stroke statistics—2019 update: A report from the American Heart Association," *Circulation*, vol. 139, no. 10, pp. e56–e528, 2019.

[4] A. Khosla, Y. Cao, C. C.-Y. Lin, H.-K. Chiu, J. Hu, and H. Lee, "An integrated machine learning approach to stroke prediction," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2010, pp. 183–192.

[5] M. Monteiro *et al.*, "Using machine learning to improve the prediction of functional outcome in ischemic stroke patients," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 15, no. 6, pp. 1953–1959, Nov./Dec. 2018.

[6] S. F. Sung, C. Y. Lin, and Y. H. Hu, "EMR-based phenotyping of ischemic stroke using supervised machine learning and text mining techniques," *IEEE J. Biomed. Health Inform.*, vol. 24, no. 10, pp. 2922–2931, Oct. 2020.

[7] G. Lim *et al.*, "Feature isolation for hypothesis testing in retinal imaging: An ischemic stroke prediction case study," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, pp. 9510–9515.

[8] S. Cheon, J. Kim, and J. Lim, "The use of deep learning to predict stroke patient mortality," *Int. J. Environ. Res. Public Health*, vol. 16, no. 11, 2019, Art. no. 1876.

[9] D. R. Pereira, P. P. R. Filho, G. H. de Rosa, J. P. Papa, and V. H. C. de Albuquerque, "Stroke lesion detection using convolutional neural networks," in *Proc. Int. Joint Conf. Neural Netw.*, 2018, pp. 1–6.

[10] D. Teoh, "Towards stroke prediction using electronic health records," *BMC Med. Informat. Decis. Mak.*, vol. 18, no. 1, pp. 1–11, 2018.

[11] T. Liu, W. Fan, and C. Wu, "A hybrid machine learning approach to cerebral stroke prediction based on imbalanced medical dataset," *Artif. Intell. Med.*, vol. 101, 2019, Art. no. 101723.

[12] F. Wang, L. P. Casalino, and D. Khullar, "Deep learning in medicine—promise, progress, and challenges," *JAMA Intern. Med.*, vol. 179, no. 3, pp. 293–294, 2019.

[13] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 843–852.

[14] A. O'Brien, C. Rajkumar, and C. J. Bulpitt, "Blood pressure lowering for the primary and secondary prevention of stroke: Treatment of hypertension reduces the risk of stroke," *J. Cardiovasc. Risk*, vol. 6, no. 4, pp. 203–205, 1999.

[15] J. E. Manson *et al.*, "A prospective study of maturity-onset diabetes mellitus and risk of coronary heart disease and stroke in women," *Arch. Intern. Med.*, vol. 151, no. 6, pp. 1141–1147, 1991.

[16] J. Lee, P. Sattigeri, and G. Wornell, "Learning new tricks from old dogs: Multi-source transfer learning from pre-trained networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 4372–4382.

[17] H. Chen *et al.*, "Standard plane localization in fetal ultrasound via domain transferred deep neural networks," *IEEE J. Biomed. Health Informat.*, vol. 19, no. 5, pp. 1627–1636, Sep. 2015.

[18] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 759–766.

[19] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Co-clustering based classification for out-of-domain documents," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2007, pp. 210–219.

[20] V. Cheplygina, I. P. Peña, J. H. Pedersen, D. A. Lynch, L. Sørensen, and M. de Bruijne, "Transfer learning for multicenter classification of chronic obstructive pulmonary disease," *IEEE J. Biomed. Health Inform.*, vol. 22, no. 5, pp. 1486–1496, Sep. 2018.

[21] Y. Xu *et al.*, "A unified framework for metric transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 6, pp. 1158–1171, Jun. 2017.

[22] C. Wan, R. Pan, and J. Li, "Bi-weighting domain adaptation for cross-language text classification," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011.

[23] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.

[24] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2013, pp. 7304–7308.

[25] T. Scott, K. Ridgeway, and M. C. Mozer, "Adapted deep embeddings: A synthesis of methods for k-shot inductive transfer learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 76–85.

[26] X. S. Zhang, F. Tang, H. H. Dodge, J. Zhou, and F. Wang, "Metapred: Meta-learning for clinical risk prediction with limited patient electronic health records," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 2487–2495.
[27] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, "Generating multi-label discrete patient records using generative adversarial networks," in *Proc. Mach. Learn. Healthcare Conf.*, 2017, pp. 286–305.
[28] O. Reyes, C. Morell, and S. Ventura, "Effective active learning strategy for multi-label learning," *Neurocomputing*, vol. 273, pp. 494–508, 2018.
[29] E. O'Connell D. Moore, and T. Newe, "Challenges associated with implementing 5G in manufacturing," in *Telecom*, Multidisciplinary Digital Publishing Institute, 2020, pp. 48–67.
[30] M. Polese, R. Jana, V. Kounev, K. Zhang, S. Deb, and M. Zorzi, "Machine learning at the edge: A data-driven architecture with applications to 5G cellular networks," *IEEE Trans. Mobile Comput.*, to be published, doi: 10.1109/TMC.2020.2999852.
[31] V. K. N. Lau, S. Cai, and M. Yu, "Decentralized state-driven multiple access and information fusion of mission-critical iot sensors for 5G wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 869–884, May 2020.
[32] J. Luketina, M. Berglund, K. Greff, and T. Raiko, "Scalable gradient-based tuning of continuous regularization hyperparameters," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2952–2960.
[33] J. Snoek *et al.*, "Scalable bayesian optimization using deep neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2171–2180.
[34] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.
[35] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, 2019.
[36] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[37] W. Li, Y. Zhao, X. Chen, Y. Xiao, and Y. Qin, "Detecting Alzheimer's disease on small dataset: A knowledge transfer perspective," *IEEE J. Biomed. Health Inform.*, vol. 23, no. 3, pp. 1234–1242, May 2019.
[38] P. Mohassel and Y. Zhang, "SecureML: A system for scalable privacy-preserving machine learning," in *Proc. IEEE Symp. Secur. Privacy*, 2017, pp. 19–38.
[39] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *TIST*, vol. 10, no. 2, pp. 1–19, 2019.
[40] J. Y. Hesterman, L. Caucci, M. A. Kupinski, H. H. Barrett, and L. R. Furenlid, "Maximum-likelihood estimation with a contracting-grid search algorithm," *IEEE Trans. Nucl. Sci.*, vol. 57, no. 3, pp. 1077–1084, Jun. 2010.
[41] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012.
[42] K. Mitra, K. Deb, and S. K. Gupta, "Multiobjective dynamic optimization of an industrial nylon 6 semibatch reactor using genetic algorithm," *J. Appl. Polym. Sci.*, vol. 69, no. 1, pp. 69–87, 1998.
[43] L. B. Jack and A. K. Nandi, "Fault detection using support vector machines and artificial neural networks, augmented by genetic algorithms," *Mech. Syst. Signal Process.*, vol. 16, no. 2, pp. 373–390, 2002.
[44] C. B. Kalayci, O. Polat, and S. M. Gupta, "A hybrid genetic algorithm for sequence-dependent disassembly line balancing problem," *Ann. Operations Res.*, vol. 242, no. 2, pp. 321–354, 2016.
[45] M. Pelikan *et al.*, "BOA: The Bayesian optimization algorithm," in *Proc. Genetic Evolutionary Comput. Conf.*, 1999, vol. 1, pp. 525–532.
[46] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*. Springer, 2003, pp. 63–71.
[47] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 2951–2959.
[48] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[49] S. Zhang, Y. Hu, and G. Bian, "Research on string similarity algorithm based on levenshtein distance," in *Proc. IEEE 2nd Adv. Inf. Technol., Electron. Automat. Control Conf.*, 2017, pp. 2247–2251.
[50] I. Borg and P. J. F. Groenen, *Modern Multidimensional Scaling: Theory and Applications*. Springer Science & Business Media, 2005.
[51] J. T. Springenberg, A. Klein, S. Falkner, and F. Hutter, "Bayesian optimization with robust Bayesian neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 4134–4142.
[52] J. Mockus, V. Tiesis, and A. Zilinskas, "The application of Bayesian methods for seeking the extremum," *Towards Glob. Optim.*, vol. 2, no. 117–129, pp. 117–129, 1978.
[53] F. Wang, R. Kaushal, and D. Khullar, "Should health care demand interpretable artificial intelligence or accept "black Box" medicine?," *Ann. Intern. Med.*, vol. 172, no. 1, pp. 59–60, 2019.