IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING

# A Novel Semi-Supervised Learning Model for Smartphone-Based Health Telemonitoring

Nathan Gaw, Member, IEEE, Jing Li<sup>10</sup>, Member, IEEE, and Hyunsoo Yoon<sup>10</sup>, Member, IEEE

Abstract—Telemonitoring is the use of electronic devices such as smartphones to remotely monitor patients. It provides great convenience and enables timely medical decisions. To facilitate the decision making for each patient, a model is needed to translate the data collected by the patient's smartphone into a predicted score for his/her disease severity. To train a robust predictive model, semi-supervised learning (SSL) provides a viable approach by integrating both labeled and unlabeled samples to leverage all the available data from each patient. There are two challenging issues that need to be simultaneously addressed in using SSL for this problem: (1) feature selection from high-dimensional noisy telemonitoring data; and (2) instance selection from many, possibly redundant unlabeled samples. We propose a novel SSL model allowing for simultaneous feature and instance selection, namely the S2SSL model. We present a real-data application of telemonitoring for patients with Parkinson's Disease using their smartphone-collected activity data such as tapping and speaking. A total of 382 features were extracted from the activity data of each patient. 74 labeled and 563 unlabeled instances from 37 patients were used to train S2SSL. The trained model achieved a high accuracy of 0.828 correlation between the true and predicted disease severity scores on a validation dataset.

Note to Practitioners—Telemonitoring is an emerging health care platform enabled by smartphones and wearables. Because it allows for health data to be collected anytime and anywhere, patients can be frequently monitored and medical decisions can be made more timely and effectively. This paper addresses the data science challenges in leveraging the telemonitoring platform to benefit patient care. Specifically, we propose a new model, S2SSL, to tackle these challenges and provide better robustness, accuracy, and efficiency. This paper may be interesting to health care practitioners seeking advanced analytics capabilities to model and integrate the data collected through telemonitoring

Manuscript received 28 April 2022; revised 20 August 2022; accepted 3 October 2022. This article was recommended for publication by Associate Editor K. Paynabar and Editor X. Xie upon evaluation of the reviewers' comments. This work was supported in part by NIH under Grant U01 CA220378-01 and in part by NSF under Grant DMS-1903135 and Grant CMMI-1149602. (*Corresponding author: Hyunsoo Yoon.*)

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Western Institutional Review Board.

Nathan Gaw is with the Department of Operational Sciences, Air Force Institute of Technology, Wright-Patterson AFB, OH 43433 USA (e-mail: Nathan.Gaw@afit.edu).

Jing Li is with the H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: jing.li@isye.gatech.edu).

Hyunsoo Yoon is with the Department of Industrial Engineering, Yonsei University, Seodaemun-gu 03722, South Korea (e-mail: hs.yoon@yonsei.ac.kr).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASE.2022.3218132.

Digital Object Identifier 10.1109/TASE.2022.3218132

devices, with ultimate purposes of improving the decision in treating each patient and increasing patient access to specialized care.

1

*Index Terms*—Machine learning, statistical modeling, health care, mobile health, telemonitoring, Parkinson's disease.

# I. INTRODUCTION

**T**ELEMONITORING is the use of electronic devices to remotely monitor patients. Recent years have witnessed a surge of using smartphones for telemonitoring. According to a 2019 Pew Research Survey, approximately 81% of American adults own a smartphone [67]. Various sensors are equipped on a smartphone, such as a microphone, camera, accelerometer, and gyroscopes. Together with specially-designed apps, smartphones can collect abundant health data of the users.

In this paper, we focus on smartphone-based telemonitoring of Parkinson's Disease (PD). PD is the second most common neurodegenerative disorder (after Alzheimer's Disease). PD currently affects seven to ten million people worldwide [1]. Patients suffer from movement disorder, tremors, and voice impairment. There is no known cure for PD, but effective treatment may slow down the progression.

Conventionally, to assess PD severity, patients need to go to a specialized clinic to be examined. One of the commonly used clinical instruments to assist the examination is called the Movement Disorder Society Unified Parkinson's Disease Rating Scale (MDS-UPDRS) [3]. MDS-UPDRS is a survey with 65 questions. The total score summing over a patient's answers to these questions ranges from 0 to 64 indicating the worst possible disability. The limitation of in-clinic examination is that it cannot be done frequently due to practical constraints such as cost, clinical staffing, and logistic inconvenience. Thus, clinical visits of most patients range from four to six months, and this situation is even worse for patients living in remote, resourcepoor areas. Infrequent examinations make it difficult to closely track disease progression, which poses a significant challenge to timely adjustment of treatment for the optimal result.

To overcome the challenge and limitations of in-clinic examination, telemonitoring provides great promise. Several smartphones or smartwatch-based apps for PD have been developed in recent years such as mPower [2], Fox Wearable Companion [72], and PDMove [73]. These apps guide the user through some pre-designed activities to measure PD symptoms such as walking, tapping, and speaking. The activity data is collected by built-in sensors of the smartphone/smartwatch such as a microphone, accelerometer, and gyroscope. As the app can be used at patient self-designated times and places,

1545-5955 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information. this flexibility makes it possible to collect frequent data about each patient for timely assessment.

Accompanying the advantages of telemonitoring data are data science challenges regarding how to integrate the multimodal, high-dimensional, noisy datasets collected by the telemonitoring platform. Specifically, this paper focuses on the task of building a machine learning model to predict a disease severity score Y (e.g., MDS-UPDRS total score) based on the activity data X collected by the telemonitoring app (e.g., tapping, speaking, etc.). With this model, the disease severity of a patient can be predicted each time the patient uses the app to perform the activities. This allows the disease severity to be assessed in a timely manner, progression to be closely tracked, and treatment to be properly adjusted.

There are several challenging issues in building the predictive model. First, although the activity data X can be collected frequently through telemonitoring (e.g., daily), disease assessment Y is significantly less frequent (e.g., in months). As a result, the labeled samples of (X, Y) for each patient are quite limited, which is insufficient to train a robust model. Furthermore, although it could be helpful to include the abundant unlabeled samples (i.e., samples of X without matching Y) to facilitate the model training, it remains a challenge how to effectively leverage these unlabeled samples. In machine learning, the subfield of integrating labeled and unlabeled samples for building a predictive model is known as semi-supervised learning (**SSL**). However, the existing SSL models lack a simultaneous feature selection and instance selection (**S2**) capability.

The S2 capability of SSL is essential for our problem. Feature selection is important because the activity data collected via telemonitoring is high-dimensional and noisy. Instance selection refers to the selection of informative unlabeled samples to include in model training. As mentioned previously, it is easy to generate activity data, which results in many unlabeled samples from each patient. However, including all of them in modeling training is computationally inefficient. Also, because the activity is performed by patients without supervision, the data quality of unlabeled samples could vary depending on environmental disturbance and patient compliance. Thus, it is important to select informative unlabeled samples to include in model training. To our best knowledge, there is limited work in SSL that simultaneously addresses feature and instance selection in a unified framework.

In this paper, we propose a novel S2SSL model, and the contributions of this work are summarized as follows:

- New model formulation to integrate S2 into the SSL design: We propose a novel mathematical formulation of S2SSL based on manifold learning in the Reproducing Kernel Hilbert Space (RKHS). Both feature and instance selection are integrated within the same non-linear model framework that provides flexibility for modeling the complicated relationship between the activity data and MDS-UPDRS.
- Algorithm for parameter estimation of S2SSL: We develop an algorithm for estimating the parameters of the proposed S2SSL model by integrating integer

programming and bio-inspired swarm intelligence optimization. The former helps find the optimal instance subset efficiently and meanwhile preserves the manifold underlying the original instance set. The latter provides a flexible wrapper approach for feature selection, allowing for finding the near-optimal solution efficiently using parallel computing resources.

Contribution to telemonitoring of PD: we apply S2SSL to an application of smartphone-based telemonitoring of PD patients using activity data collected by mPower. S2SSL achieves high accuracy in predicting MDS-UPDRS. The selected features by our algorithm also shed some light on which aspects of the movement and speech functions of PD patients are mostly impaired by the disease. We discuss how the proposed method can potentially help improve health care automation in several aspects such as: enabling frequent, remote health monitoring for each patient and timely medical decisions; improving patient access to advanced care; and facilitating the design of efficient and effective patient triage systems. A general framework for the telemonitoring workflow to translate activity data collected from smartphones or smartwatch-based apps into a severity score of the disease is shown in Fig. 1).

The remainder of this paper is organized as follows: Section II reviews related works and points out gaps. Section III presents the mathematical formulation of S2SSL. Section IV presents the parameter estimation algorithm. Section V conducts simulation experiments. Section VI provides the application case study. Section VII concludes the paper.

# **II. RELATED WORKS**

The methodological development of this paper is related to SSL in machine learning. We first provide an overview of different SSL models (Sec. II.A). Then, we review the existing work of instance selection in SSL (Sec. II.B) and the existing work of feature selection in SSL (Sec. II.C). We found that there is limited work for integrating instance and feature selection in a unified framework (Sec. II.D). This gap drives the methodological development of this paper.

### A. Semi-Supervised Learning (SSL)

SSL models fall into several major categories including generative models [4], [5], [54], [76] self-training [6], [7], [55], [77] co-training [8], [9], [56], [78] low-density separation [10], [11], [57], [79] and manifold learning [12], [13], [58], [59], [80]. Among these methods, manifold learning has drawn much attention in recent years due to the rigorous mathematical formulation and demonstrated good performance in various applications. Specifically, graph-based manifold learning is considered an excellent choice with high-dimensional features. For example, Zhu and Lafferty [12] proposed a regularized generative mixture model with graph Laplacian and demonstrated its application on handwritten digit and teapot image datasets. Xie et al. [58] introduced a novel manifold regularization (MR) based distributed SSL algorithm on the "Concrete" and "Breast Cancer Wisconsin (Diagnostic)"

GAW et al.: NOVEL SEMI-SUPERVISED LEARNING MODEL FOR SMARTPHONE-BASED HEALTH TELEMONITORING



Fig. 1. Overview of telemonitoring workflow, data science challenges in predictive modeling, and proposed solution. Telemonitoring Workflow: First a patient performs exercises on a smartphone. The information from the exercises is then recorded by the smartphone and sent to a database where feature extraction is performed. Next, predictive model takes as input the extracted features and makes a prediction on the disease severity for a timely medical decision. In Predictive Modeling, there are three common issues in telemonitoring that are addressed by this work: (1) small sample size of labeled data (addressed by semi-supervised learning), (2) high-dimensional and noise features (addressed by feature selection), and (3) many (redundant) unlabeled instances (addressed by instance selection). The main contribution of this work is building a model that can perform all three tasks simultaneously to predict a continuous disease severity score.

datasets. Zhao et al. [59] developed a semi-supervised broad learning system using manifold regularization on the G50C, MNIST, and NORB datasets. Belkin et al. [13] proposed a graph-based regularization framework that relied on the properties of RKHS to enable efficient and accurate prediction. In this paper, graph-based manifold learning is chosen to form the base model of the proposed S2SSL.Research works that extended the model in [13] have mainly focused on transfer learning [84], [85], [86], [87], [88], domain adaptation [89], [90], [91], [92], [93], non-negative matrix factorization [94], [95], [96], graph convolutional neural networks [97], [98], [99], [100], and other deep learning applications [101], [102], [103], [104], [105], [106], [107]. There is also some work on feature selection [62], [75], [108], [109], [110], [111], [112] and instance selection [113], [114], [115]. However, none of these existing works have covered simultaneous instance and feature selection in semi-supervised regression. The reason why we considered the original model in [13] as our base model is because of the simplicity of its implementation for finding a solution via the Representer Theorem [35]. With a given set of features and instances, our proposed S2SSL reduces to the original Laplacian Regularized Least Squares in reference [13], allowing for ease in implementation by practitioners who are familiar with the method and need to augment it with feature and instance selection.

# B. Instance Selection in SSL

In graph-based manifold learning, the graph is constructed with nodes being labeled and unlabeled instances. The weight of an edge between two nodes is computed by a kernel in the feature space. Because the graph is used as an input to an SSL algorithm, the computational efficiency and accuracy of the algorithm are affected by the graph. To reduce the complexity of the graph, there are several ways: graph sparsification, graph embedding, and instance selection.

Graph sparsification reduces edges, not instances/nodes. Typical algorithms include KNN [14], b-matching [15], and minimum spanning tree [16]. Graph embedding reduces the complexity of the graph Laplacian matrix by sparse coding [17], [18] or low-rank approximation [19]. Instance selection reduces the nodes, which results in a smaller graph. Because an SSL algorithm typically needs to invert the graph Laplacian matrix, instance selection provides a more direct approach to help stabilize the matrix inversion and improve computational efficiency. Despite the advantage, instance selection is less studied in the SSL literature. We noted one work by Sun et al. [20], in which a manifold-preserving instance selection algorithm was proposed to maximize the total edge weight between the selected and unselected instances/nodes under a given number of selected instances.

#### C. Feature Selection in SSL

Feature selection has been primarily investigated for supervised learning methods. Less work is done for SSL [63]. Some of the methods developed for supervised learning can be applied to SSL. Feature selection methods can be divided into filter, embedded, and wrapper methods. Filter methods do not need the labels of samples/instances. Therefore, theoretically speaking, any filter method developed for supervised learning can be used for SSL. However, because feature filtering does not consider the predictive value of the features to the response variable, there is a high risk that relevant features may be filtered out. Embedded methods incorporate feature selection into the objective function of fitting a predictive model [21], [22], [23], [62], [66], [75], [81]. These methods enjoy the benefit of integrating feature selection and predictive modeling fitting. The limitation, however, is that the integration/embedding mechanism must be designed specifically for each type of predictive model, and therefore is not universally applicable.

In contrast, wrapper methods do not cling to any particular predictive model and can be wrapped around any base learner. This provides greater flexibility. In SSL, wrapper methods such as sequential feature selection have been integrated with self-training [24], [25], [65], [82]. Ensemble learning has been used to extend the capability of a single base learner to improve the robustness [26], [27], [28], [64], [83].

# D. Gaps of the Existing Research

From our literature review of SSL papers, we found that although feature selection and instance selection have been separately explored for SSL models, there is limited work for integrating instance and feature selection in a unified framework. This drives our methodological development in this paper. Specifically, the proposed S2SSL uses graph-based manifold learning as the base model and augments its capacity by simultaneous feature and instance selection. For instance selection, we propose a novel integer programming optimization to find the minimum number of instances that are needed to preserve the manifold. This overcomes several limitations of existing algorithms, e.g., requiring a pre-determined number of selected instances and using a greedy search [20]. For feature selection, we propose a flexible wrapper approach based on bio-inspired swarm intelligence optimization, which allows for finding the near-optimal solution efficiently using parallel computing resources.

There has been limited work performed with SSL in the area of telemonitoring[60], [61]. Deshmukh et al. [60] proposed a semi-supervised transfer learning model that was tested on the Parkinson's Disease Telemonitoring dataset from the UCI machine learning repository for prediction accuracy. Gogna et al. [61] developed an autoencoder-based framework that simultaneously reconstructs electrocardiogram and electroencephalograms in a semi-supervised fashion. Neither of these works address feature or instance selection in semisupervised frameworks, and there is still much work to be performed in the area of SSL in telemonitoring.

# **III. MATHEMATICAL FORMULATION OF S2SSL**

Suppose there are *L* labeled samples/instances,  $\{\mathbf{x}_l, y_l\}_{l=1}^L$ . Note that we will use "samples" and "instances" interchangeably in this paper.  $\mathbf{x}_l$  is a set of features (e.g., the features extracted from smartphone-collected activity data).  $y_l$  is the response variable (e.g., the MDS-UPDRS summary score). Assume  $y_l$  is on a continuous scale, whereas the extension to other types of response is straightforward. For example, if one were interested in classification using a binary response variable  $y_l$  can be transformed by using a sigmoid function and thresholding. In addition, suppose there are *U* unlabeled instances.  $\{\mathbf{x}_l\}_{l=L+1}^{L+U}$ . The base model we will build S2SSL upon is a graph-based manifold learning model developed by Belkin et al. [13]: (1) as shown at the bottom of the next page, f is a predictive function on the RKHS, i.e.,  $f \in \mathcal{H}_K$ , with a Mercer Kernel K, which can model a non-linear relationship between the features and the response variable.  $||f||_{K}^{2}$  is a norm on  $H_{K}$ , which encourages stability and generalizability of the solution.  $\gamma_A$  and  $\gamma_I$  are tuning parameters. The 3rd term in (1) deserves more explanation:  $\mathbf{f} = (f(\mathbf{x}_1), \dots, f(\mathbf{x}_{L+U}))^T$  contains the predictions on all labeled and unlabeled instances.  $\Phi$  is a  $(L+U) \times (L+U)$ matrix that regularizes **f**. A common choice of  $\Phi$  is the Laplacian matrix of a graph G = (V, W). V contains labeled and unlabeled instances as nodes of the graph. W contains edge weights  $\{w_{ij}\}$  between each pair of nodes. The weight is higher if two nodes/instances are closer on the feature space. Under this definition  $\mathbf{\Phi}$ , the 3<sup>rd</sup> term is equivalent to:

$$\mathbf{f}^T \mathbf{\Phi} \mathbf{f} = \sum_{i,j=1,\dots,L+U,i < j} w_{ij} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2.$$

The role of this term is to encourage the predictions of two instances to be similar if these instances have a bigger weight in the graph. This is also known as manifold learning because the graph characterizes the manifold underlying the observed features.

The model in (1) has two major areas for improvement: First, it includes all the unlabeled instances. This creates problems in numerical stability and computational efficiency in solving the optimization. Second, it includes all the features, but some of them may be noisy.

To enable simultaneous instance and feature selection, we propose to modify the SSL model in (1) to the following form, namely the S2SSL model: (2) as shown at the bottom of the next page.

Comparing the new model in (2) with (1), two new notations are introduced: **R** and **S**.  $\mathbf{R}_{D\times D} = diag(r_1, \ldots, r_D)$  is a diagonal matrix.  $r_d = 1$  if the corresponding feature is selected and  $r_d = 0$  otherwise,  $d = 1, \ldots, D$ .  $\mathbf{S}_{(L+U)\times(L+U)} =$  $diag(s_1, \ldots, s_{L+U})$ .  $s_l = 1$  if the corresponding instance is selected and  $s_l = 0$  otherwise,  $l = 1, \ldots, L + U$ . Also, since the graph Laplacian is computed based on the selected features, we use  $\mathbf{\Phi}(\mathbf{R})$  to denote that  $\mathbf{\Phi}$  is a function of the selected features. The formulation in (2) has three parameters to be estimated: f, **R** and **S**, which is not easy. In the next section, we propose an algorithm to solve this formulation.

# IV. PARAMETER ESTIMATION OF S2SSL BY INTEGRATING INTEGER PROGRAMMING AND SWARM INTELLIGENCE OPTIMIZATION

Existing algorithms do not suffice for solving the S2SSL formulation in (2). Simultaneously solving all three parameters is impossible. Coordinate descent types of algorithms [29] that iteratively cycle through each parameter also do not work because the descending direction is hard to find. Furthermore, having the parameters complicatedly entangled with one another rules out other algorithms that tackle mixed-integer nonlinear optimization problems, such as alpha branch and bound ( $\alpha$ -BB) [30], [31] and factorable programming trees [32], [33].

GAW et al.: NOVEL SEMI-SUPERVISED LEARNING MODEL FOR SMARTPHONE-BASED HEALTH TELEMONITORING

#### A. Overview of the Proposed Algorithm

It is very difficult to solve the feature selection matrix  $\mathbf{R}$ in (2) directly, because it is embedded in the graph Laplacian matrix  $\Phi$ . To alleviate this challenge, we propose to use a wrapper approach to identify the optimal feature subset. The advantage of wrapper algorithms in feature selection is that the algorithm can be integrated with any base learner. In our case, the base learner, i.e., with a fixed feature subset, is (2) with only two parameters: the predictive function f and the instance selection matrix S. This base learner is much easier to solve than the original formulation with three parameters. Furthermore, note that in the base learner with two parameters, the instance selection matrix S only appears in the last term that regularizes the predictive functions on all labeled and unlabeled samples. This observation sheds some light on how to select instances. Specifically, our idea is that the instances should be selected as those that preserve the underlying manifold formed by all the labeled and unlabeled samples. In this way, the selected instances would impose a similar regularization effect as that by including all the instances, while at the same time greatly reducing the computational complexity.

Specifically, our proposed algorithm includes three key components:

- Instance selection: Given  $\mathbf{R}$  (a feature subset), a manifoldpreserving integer programming optimization is proposed to solve for  $\mathbf{S}(\mathbf{R})$ , which is the optimal instance subset under this specific feature subset. This optimization will be discussed in Sec. IV.B.
- Solving the predictive function: With the **R** and **S**(**R**), the predictive function *f* can be solved using the Representer Theorem. This will be discussed in Sec. IV.C.

# B. Manifold-Preserving Integer Programming Optimization for Instance Selection

Focus on the  $3^{rd}$  term of (2) that involves **S**.  $S\Phi(\mathbf{R})S$  produces a submatrix of  $\Phi(\mathbf{R})$  that only involves selected instances. For example, consider a simple case with five instances and the first three being selected. Then,

Because this submatrix is used to regularize the predictive function, the submatrix is considered adequate if it can exert a similar regularization effect as the full matrix. In other words, the selected instances in this submatrix can preserve the underlying manifold formed by all the instances. To find the adequate submatrix (i.e., to solve for S), we propose the following optimization: (3) as shown at the bottom of the next page.

The last constraint deserves some explanation:  $a_{li}$  is an 0/1 variable indicating if instance i is a neighbor of instance l.  $a_{li}$  is obtained by applying a K-Nearest Neighbors (KNN) algorithm to the graph  $G(\mathbf{R})$  and converting it into an unweighted graph (i.e., with 0/1 weights). Refer to Sec. III for the definition of G.  $G(\mathbf{R})$  is a graph constructed in a similar way but only using the selected features in **R**.  $\sum_{i=1}^{L+U} a_{li}s_i$ counts how many neighbors of instance l are selected. This number is required to be at least  $\lambda$ , which is a tuning parameter. The  $\lambda$  parameter can be determined by using a line search. The purpose of this constraint is to make sure a certain number of neighbors of each instance to be selected, so that in case that instance is not selected, it still has a good representation in the model by its neighbors. The optimization in (3) is an integer programming problem and can be solved by commonly used solvers such as branch and bound [34].

# C. Solving Predictive Function by Representer Theorem

With fixed **R** and **S**, the formulation in (2) has only the predictive function f to be solved, i.e.,

$$\underset{f \in \mathcal{H}_{K},}{\operatorname{argmin}} \frac{1}{L} \sum_{l=1}^{L} (y_{l} - f(\mathbf{x}_{l} \mathbf{R}))^{2} + \gamma_{A} \|f\|_{K}^{2} + \gamma_{I} \mathbf{f}^{T} \mathbf{S} \boldsymbol{\Phi}(\mathbf{R}) \mathbf{S} \mathbf{f}.$$
(4)

Using the Representer Theorem [35], we know that the solution of (4) takes the following form:

$$f(\mathbf{xR}) = \boldsymbol{\alpha}^T K(\mathbf{xR}, \cdot) \mathbf{S}, \qquad (5)$$

where **x** is any instance we want to predict,  $K(\mathbf{xR}, \cdot) = (K(\mathbf{xR}, \mathbf{x}_1\mathbf{R}), \dots, K(\mathbf{xR}, \mathbf{x}_{L+U}\mathbf{R}))^T$  contains the kernels computed between **x** and each instance in the training set using the selected features in **R**.  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_{L+U})^T$  contains the coefficients. To solve these coefficients, insert (5) into (4).

$$f^{*} = \underset{f \in \mathcal{H}_{K}}{\operatorname{argmin}} \underbrace{\frac{1}{L} \sum_{l=1}^{L} (y_{l} - f(\mathbf{x}_{l}))^{2}}_{l=1} + \underbrace{\gamma_{A} \| f \|_{K}^{2}}_{q_{A} \| f \|_{K}^{2}} + \underbrace{\gamma_{I} \mathbf{f}^{T} \Phi \mathbf{f}}_{regularization on predictions of labeled and unlabeled instances}$$
(1)

$$\underset{f \in \mathcal{H}_{K}, \mathbf{R}, \mathbf{S}}{\operatorname{argmin}} \frac{1}{L} \sum_{l=1}^{L} (y_{l} - f(\widetilde{\mathbf{x}_{l}\mathbf{R}}))^{2} + \gamma_{A} \|f\|_{K}^{2} + \gamma_{l} \widetilde{\mathbf{f}^{T}\mathbf{S}} \Phi(\mathbf{R})\mathbf{S}\mathbf{f}.$$
(2)

Then, we get

$$\operatorname{argmin}_{\alpha} \frac{1}{L} (\mathbf{y} - \mathbf{J} \mathbf{K} \boldsymbol{\alpha})^{T} (\mathbf{y} - \mathbf{J} \mathbf{K} \boldsymbol{\alpha}) + \gamma_{A} \boldsymbol{\alpha}^{T} \mathbf{S} \mathbf{K} \mathbf{S} \boldsymbol{\alpha} + \gamma_{I} \boldsymbol{\alpha}^{T} \mathbf{S} \mathbf{K} \boldsymbol{\Phi} (\mathbf{R}) \mathbf{K} \mathbf{S} \boldsymbol{\alpha}$$
(6)

where  $\mathbf{y} = (y_1, \dots, y_L, 0, \dots, 0)^T$  is a  $(L + U) \times 1$  vector that contains the responses of L labeled instances followed by U zeros. **K** is a  $(L + U) \times (L + U)$  kernel matrix over the labeled and unlabeled instances in the training set using the selected features in **R**. **J** is a diagonal matrix with the first L diagonal elements being ones and the next U elements being zeros. We can derive the minimizer to the optimization problem in (6) as:

$$\boldsymbol{\alpha}^* \triangleq \begin{pmatrix} \tilde{\boldsymbol{\alpha}} \\ \boldsymbol{0} \end{pmatrix},$$

where  $\mathbf{0}$  corresponds to the instances that are not selected in  $\mathbf{S}$ , and

$$\tilde{\boldsymbol{\alpha}} = \left(\tilde{\mathbf{J}}\tilde{\mathbf{K}} + \gamma_A \mathbf{L}\mathbf{I} + \gamma_I \mathbf{L}\tilde{\boldsymbol{\Phi}}(\mathbf{R})\tilde{\mathbf{K}}\right)^{-1}\tilde{\mathbf{y}}.$$

Here, the overhead " $\sim$ " is used to denote a sub-matrix or sub-vector after removing the rows/columns or elements corresponding to the instances that are not selected in **S**.

#### D. PSO-Based Wrapper for Feature Selection

The solution of the predictive function f, as described in Sec. C, assumes that a feature subset **R** and an instance subset **S** have been given. Because **S** can be solved with a given **R** (Sec. B), the performance of f is ultimately affected by what features are selected in **R**.

To find the optimal feature subset, we adopt a wrapper method due to its flexibility of being able to integrate with any type of predictive model. The basic idea of a wrapper method is to use an efficient algorithm to search through the solution space, which in our case is the space containing different subsets of the features. Since this space is typically very large, an exhaustive search is impossible. We propose to use PSO to search for the optimal solution due to its demonstrated efficiency and optimality in various of other applications [36], [37].

In what follows, we present the detailed design of using PSO to search for the optimal feature subset in our problem. The algorithm starts by putting together a collection of *m* particles. The initial position of each particle in the feature space is randomly assigned and represented by  $\mathbf{r}_i^0$ .  $\mathbf{r}_i^0$  is a  $D \times 1$  vector of 0/1 with one representing the corresponding feature being

selected and zero otherwise, i.e.,  $\mathbf{r}_i^0$  corresponds to a subset of features. Then, these particles will move with their velocities determined by the following equation: (7) as shown at the bottom of the next page. In (7),  $\mathbf{v}_i^t$  is the velocity of particle *i* at the *t*-th iteration. On the right-hand side,  $\mathbf{v}_i^{t-1}$  is the velocity at the previous iteration.  $\mathbf{p}_i^{0:t}$  is the historically best position of particle *i*.  $\mathbf{p}_g^t$  is the globally best position among all the particles. We will discuss how to find these best positions later in this section.  $\mathbf{r}_i^t$  is the current position of particle *i*.  $b_1$  and  $b_2$  are sampled from a uniform distribution U[0, 1] to add stochasticity.  $\omega^t$ ,  $c_1$  and  $c_2$  are weights to combine the three parts in determining the velocity of particle *i*. Appropriate values for  $\omega^t$ ,  $c_1$ , and  $c_2$  are discussed in [38].

After the velocity of each particle is computed by (7), the position of the particle in the next iteration is updated by

$$r_{id}^{t+1} = \begin{cases} 1, & \text{if } H\left(v_{id}^{t}\right) > 0.5\\ 0, & \text{otherwise} \end{cases}$$
(8)

In (7),  $r_{id}^{t+1}$  is the *d*-th element of the position vector  $\mathbf{r}_i^{t+1}$ .  $v_{id}^t$  is *d*-th element of the velocity vector  $\mathbf{v}_i^t$ .  $H(v_{id}^t) = \frac{1}{1+e^{-v_{id}^t}}$  is a sigmoid function that squashes  $v_{id}^t$  into the range of [0,1]. Fig. 3 shows a simple example to demonstrate the basic idea of the PSO algorithm.

• *Feature selection*: The predictive function corresponding to the optimal **R**<sup>\*</sup> and **S**<sup>\*</sup>(**R**<sup>\*</sup>), i.e., *f*<sup>\*</sup>, will be searched using a wrapper method based on Particle Swarm Optimization (PSO). This will be discussed in Sec. IV.D.

Sec. B-D will discuss each component respectively. Fig. 2 provides a schematic overview of the inter-relationship of these components that compose the algorithm.

Note that we have a remaining question yet to be addressed in the above description of the PSO algorithm, i.e., how to find the historically best position of a particle and the globally best position among all the particles. This question boils down to how to evaluate the goodness for the position of each particle *i* at each iteration *t*,  $\mathbf{r}_i^t$ . Recall that a "position" in our context corresponds to a feature subset. For a given position/feature subset, we can use the method in Sec. C to solve the predictive function. The prediction accuracy, either on a validation set or through cross validation, can be used to assess the goodness of the position.

1) Convergence and Repeatability of PSO: Limited theoretical work has been done to investigate the convergence of PSO due to the difficulty of fully understanding the dynamics of the algorithm. Practical recommendations have been given by a few researchers. Eberhart and Shi [39] stated that a pragmatic

$\min_{s} \sum_{l=1}^{L+U} s_l  \dots  \text{Minimize the number of selected instances}$	
s.t. $s_l \in \{0, 1\}, l = L + 1, \dots, L + U \dots$ Binary decision variable for each instance	
$s_l = 1, l = 1, \dots, L$ Labeled instances must be selected.	
$\sum_{i=1}^{L+U} a_{li}s_i \ge \lambda, l = 1, \dots, L+U \dots$ At least $\lambda$ neighbors of each instance must be selected.	(3)



Fig. 2. A schematic overview of the proposed algorithm to solve the S2SSL model formulation in (2).



Fig. 3. PSO iterations to find the optimal feature subset (for this toy example assume we have three candidate features in a search space represented by a 3-dimensional unit cube): At Iteration 0, particles are chosen such that a random set of candidate features are used to solve a particular objective function (note positions of particles can only be at corners of the cube, (1,0,0), (0,1,0), ...(1,1,1), where 1 indicates that a feature has been selected, and 0 otherwise); based on model performance, particles velocities are updated (eq. (7)) and new candidate features are chosen (eq. (7)). This process continues until the particles' candidate feature sets converge or when PSO reaches Iteration N (the predefined maximum number of iterations).

approach is to have minimum and maximum values that each particle's velocities can take,  $V_{min}$  and  $V_{max}$ , respectively. In particular for binary PSO, it has been suggested to set  $V_{min} = -6$  and  $V_{max} = 6$  to avoid oversaturation in the sigmoid function. Additionally, it has practically been found that using an inertia weight  $\omega$  set to decrease linearly from 0.9 to 0.4 across the chosen iterations allows PSO to explore a large area at the start of the iterations and to refine the search

later by gradually decreasing $\omega$  [40]. Additionally, we have found PSO to be more repeatable when more particles are used in the search space. Having more particles increases computation, but with PSO it is easy to implement distributed computing to achieve a solution that is near-globally optimal. Since each particle that contains a potential solution in the swarm is independently trained and tested for a given iteration, parallel computing can be implemented to handle a swarm with many particles to search the solution space for the global optimum. These strategies were used in our experiments, and we have not seen convergence to be an issue. Finally, we summarize the steps of the S2SSL algorithm by integrating Sec. B-C (see Algorithm on the next page).

2) Tuning Parameter Selection: There are three major tuning parameters including the  $\gamma_A$  and  $\gamma_I$  in (2) and the neighborhood size  $\lambda$  in (3). We use a grid search on these parameters to minimize the prediction error on a validation set. Please see Sec. V and VI for more details.

# V. SIMULATION STUDY

# A. Comparison Between S2SSL and Supervised Learning

We adopt the commonly used S-shape manifold from scikitlearn [41] to generate simulation data (Fig. 4(a)). The training set includes 150 instances (6 labeled and the others unlabeled). Response variables of the labeled instances are indicated by different colors. A separate validation set of 25 instances is generated for evaluating model accuracy.

In training the S2SSL model, edge weights of the graph over labeled and unlabeled instances were computed using a Gaussian kernel (when compared to other kernels, such as linear or polynomial, using a Gaussian kernel resulted in the most accurate predictions). In order to input the graph into the instance selection algorithm (Eq. (3)), the graph must be binary and indicate each instance's nearest neighbors (with 1 indicating a nearest neighbor, and 0 otherwise). Thus, the graph was converted to an unweighted one using a KNN algorithm. The best performance was achieved when K=6. The  $\lambda$  parameter in the instance selection algorithm (eq. (3)) was determined by a line search over [1,7].  $\gamma_A$  and  $\gamma_I$  were searched over a grid of  $[1 \times 10^{-3}, \dots, 1 \times 10^{1}] \times [1 \times 10^{-3}, \dots, 1 \times 10^{1}].$ For comparison, a supervised learning (SL) model was also trained using only the labeled instances, which is equivalent to setting  $\gamma_I$  to be zero. Fig. 4 (b) shows the mean absolute error (MAE) between predicted and true responses on the validation set for the two models. S2SSL significantly outperforms SL (p < 0.001).

# B. Utility of Feature Selection in S2SSL

We added noise features to the dataset from the Sec. A by sampling each feature from N(0, 5). Experiments were

$$\mathbf{v}_{i}^{t} = \omega^{t} \mathbf{v}_{i}^{t-1} + b_{1}c_{1} \qquad \overbrace{(\mathbf{p}_{i}^{0:t} - \mathbf{r}_{i}^{t})}^{move to historical} + b_{2}c_{2} \qquad \overbrace{(\mathbf{p}_{g}^{t} - \mathbf{r}_{i}^{t})}^{move to global}$$

$$(7)$$

IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING



Fig. 4. (a) 150 training instances (6 labeled – in red circles; others unlabeled – grey dots) on S-shape manifold. (b) MAE (s) denotes Mean Absolute Error (MAE)  $\pm$  standard deviation (s) of prediction on the validation set by supervised learning and semi-supervised learning capability provided by S2SSL. (c) MAE of S2SSL without and with the feature selection capacity under different numbers of noise features.

performed with 1, 5, 10, 50, and 100 noise features. S2SSL models were trained with feature selection using PSO and without feature selection. For the former model, the number of particles used in PSO was 10 times the number of noise features. Fig. 4 (c) shows the result. Without feature selection, the model performance deteriorates as more noise features are included. With feature selection, the MAE is significantly lower (p < 0.001).

# C. Utility of Instance Selection in S2SSL

We increased the number of instances in Sec. A to 2000 (6 labeled and the rest unlabeled) (Fig. 5(a)). No noise features were added. S2SSL models were trained with instance selection by the proposed integer programming optimization and without instance selection. For the former model, the best performance was achieved at  $\lambda = 2$  (see Table I). With instance selection, the MAE is smaller but the difference is not statistically significant. However, there is a 70.4% reduction of model training time.

# D. Performance of Simultaneous Feature and Instance Selection in S2SSL

This final experiment is based on a simulation dataset with noise features (1, 5, 10, 50, 100) and a large number of instances (2000). Depending on the number of noise features, 3-100 particles were used in the PSO algorithm for feature selection.  $\lambda = 2$ , the best setting found in Sec. C, is used for the instance selection algorithm. Fig 5(b) shows the result. It is clear that the MAE with feature and instance selection is significantly lower (p < 0.001).

# VI. APPLICATION TO SMARTPHONE-BASED TELEMONITORING OF PD

In this section, we present an application of S2SSL for predicting the severity of PD using data collected from patients' smartphones.

## A. Data Collection

The data collection was guided by a patient's smartphone running an app mPower, which was developed by Sage Bionetworks and released in March 2015. Sage Bionetworks organized a nationwide study that enrolled patients with PD to use mPower to collect their activity data [2]. To participate in the mPower study, each participant needed to selfnavigate through eligibility criteria and submit e-consent to the conditions. Once the consent process is finished, users were presented with the option of performing several activities guided by the mPower app, such as "tapping", "speaking", etc., which were intended to measure PD-related symptoms.

For the tapping activity, the app instructs the user to use two fingers on the same hand to tap alternately between two fixed points on the screen for a period of 20 seconds (Fig. 6). Time series data of the tapping process is recorded. For the speaking activity, the user is instructed to say 'Aaaaah' into the microphone at a steady volume for at most 10s (Fig. 6). The voice signal is recorded.

Each participant of the mPower study not only has the activity data collected through their smartphones at least once per day, but also has data of the clinical instrument/questionnaire, MDS-UPDRS, collected at a much less frequent basis (usually on a monthly basis). The summary score of MDS-UPDRS is used as the response variable for representing PD severity. A score of 0 denotes no disability, while a score of 64 indicates the worst possible disability. For additional information and demonstration of specific case examples, we refer the reader to the mPower Public Researcher Portal (https://www.synapse.org/mpower).

# B. Feature Extraction From Smartphone-Collected Data

43 features were extracted from the tapping data based on previous studies [42], [43], [44]. These features measure tapping speed, inter-tap interval, position, fatigue, etc. 339 features were extracted from the speaking time series data,



Fig. 5. (a) 2000 training instances on S-shape manifold. (b) MAE (s) denotes MAE  $\pm$  standard deviation (s) of S2SSL without and with the simultaneous instance & feature selection capacity under different numbers of noise features.

TABLE I

Comparison of S2SSL at different levels of  $\lambda$ . Validation Errors are in Terms of Mean Absolute Error (MAE  $\pm$  Standard Deviation) and Pearson Correlation; Time Refers to the Time to Sample + Train + Test; Number Sampled is the Number of Instances Sampled From the Dataset. The Best Sampling Result is  $\lambda = 2$  (in bold)

	λ	Mean Absolute Error (MAE)	Pearson Correlation	Time (s)	Number Sampled
No Sampling	N/A	$0.495\pm0.343$	0.984	41.9	N/A
	1	$0.506\pm0.375$	0.986	8.5	65
	2	$0.363\pm0.315$	0.990	12.4	106
	3	$0.460\pm0.331$	0.987	6.6	169
Sampling	4	$0.568\pm0.516$	0.972	8.9	235
	5	$0.594\pm0.438$	0.976	8.9	236
	6	$0.611\pm0.458$	0.974	9.7	285
	7	$0.611 \pm 0.467$	0.972	7.7	309



Fig. 6. Smartphone-collected activities used in our application and feature extraction.

based on previous studies [45], [46]. These features characterize amplitude (shimmer variants), frequency (jitter variants), increased noise (signal-to-noise measures), etc (Fig. 6). Modeling by S2SSL.

A subset of 37 PD patients from the mPower study was included in our application. These patients were selected on the basis of having monthly MDS-UPDRS scores for at least three months as well as complete daily tapping and speaking information. S2SSL was trained on three different datasets: (1) tapping, (2) speaking, (3) tapping + speaking combined. The reason why we decided to test on combined datasets of tapping and speaking is because there is significant variability in the presentation and progression of PD symptoms [2] across patients, and we hypothesize that having a model trained on different types of PD symptoms will result in significantly improved results. For each dataset, two labeled instances (i.e., instances with MDS-UPDRS available) were randomly selected from each patient and included in the training set, together with unlabeled instances. The remaining labeled instances from each patient were included in the validation set. The training set contained a total of 563 unlabeled instances and 74 labeled instances, while the validation set contained 70 instances. There was a total of 144 labeled instances (with

IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING

				res selected by S2SSL	Definition	
	MAE	Correlation between predicted		kurTapInter	Kurtosis of inter-tap interval	
	(3)		Tapping	madDriftRight	Median absolute deviation of drift of finger po between consecutive taps in the right button	
Tapping 3.580 (0.328)	3.580	30 18)		Shimmer- >F0_PQ5_classical_Schoentgen	Classical Schoentgen of the shimmer signal u cycle samples.	
	(0.520)			Shimmer->F0_FM	Frequency modulation of the shimmer signal	
				mean_MFCC_1st	Mean of 1st Mel Frequency Cepstral Coeffici	
Speaking (0.33	0.000	r=0.759		mean_MFCC_6th	Mean of 6th Mel Frequency Cepstral Coeffici	
	(0.337)		Speaking	std_8th_delta_delta	Standard deviation of the 8th delta delta (2nd derivative) Mel Frequency Cepstral Coefficie	
				det_TKEO_mean_10_coef	Mean Teager-Kaiser energy operator 10th det coefficient	
2.600 + (0.311)	r = 0.828		app_LT_entropy_log_1_coef	Log entropy of 1st approximation coefficient prior F0 transform)		
	(0.511)			app_LT_entropy_log_5_coef	Log entropy of 5th approximation coefficient prior F0 transform)	
		(a)		(	Ъ)	

Fig. 7. (a) Prediction accuracy of S2SSL trained on tapping, speaking, and tapping + speaking datasets (where MAE (s) denotes MAE  $\pm$  standard deviation (s) and Pearson correlation is used). (b) Selected features and their definitions from the model trained on the tapping + speaking dataset.

#### TABLE II

Comparison of Proposed S2SSL With Benchmarks Across All Simulation and Application Tests. S2SSL Performed Significantly Better Than the Competing Methods for Both Simulation Datasets (p < 0.001) and the PD Telemonitoring Application Dataset (p < 0.005). MAE (s) Denotes MAE ± Standard Deviation (s). SSL-RF=Semi-Supervised Learning Random Forest Tree Ensembles; MSSRA=Multi-Scheme Semi-Supervised Regression Approach; GS<sup>3</sup>FS=Graph-Based Semi-Supervised Sparse Feature Selection

	Simulation MAE (s)		Application: PD Telemonitoring MAE (s)		
Model	150 Instance S-Curve	2000 Instance S-Curve	Tapping	Speaking	Tapping + Speaking
SSL-RF	2.348 (1.351)	2.358 (1.345)	4.594 (3.300)	4.640 (3.089)	4.586 (3.100)
MSRRA	0.964 (0.624)	1.103 (0.638)	4.717 (3.717)	5.185 (4.292)	5.131 (4.148)
GS <sup>3</sup> FS	1.460 (1.034)	1.485 (1.124)	11.096 (6.106)	10.934 (5.835)	10.388 (6.440)
S2SSL	0.293 (0.249)	0.363 (0.315)	3.580 (0.328)	2.823 (0.337)	2.600 (0.311)

both tapping/speaking features and MDS-UPDRS available) from the 37 patients. The 144 instances were approximately equally split between the training set (74) and the validation set (70). In the training set, two instances were included from each patient to avoid the potential risk that the model may be biased by some patients (if more data from these patients were included than others).

To expedite the training process of S2SSL, computing was performed using two Intel Xeon E5-2680 v4 CPUs running at 2.40 GHz, which provide 28 CPU cores to perform calculations for each particle in the PSO in parallel. To further limit the computational time, we constrained the maximum number of ones in the position vector of each particle to be 20.

### C. Results and Interpretation

Fig. 7 (a) provides a summary of the results for S2SSL trained on tapping, speaking, and tapping + speaking features. Inclusion of both tapping and speaking features has a clear benefit. The reduction of MAE is significant (p = 0.01). Also, we plotted the predicted versus actual MDS-UPDRS, which shows better correlation of using combined features.

Since S2SSL trained on tapping + speaking features achieved the highest accuracy, we examine this result more closely. 10 features were selected in the combined dataset (2

from tapping and 8 from speaking). Fig. 7(b) provides the definition for each feature chosen.

The tapping features chosen were the *kurTapInter* and *madDriftRight*. *kurTapInter* is the kurtosis of the inter-tap interval. *madDriftRight* is the median absolute deviation of drift of finger position between consecutive taps in the right button. PD patients have been found to have a higher intra-individual variability of finger tapping due to a lack of control in fine motor capabilities [47].

The speaking features chosen fit under three categories:

- Shimmer (Shimmer->F0\_PQ5\_classical\_Schoentgen and Shimmer->F0\_FM),
- 2) Mel Frequency Cepstral Coefficients (MFCCs) (mean\_MFCC\_1st, mean\_MFCC\_6th, and std\_8th\_delta\_delta), and
- 3) Wavelet measures (*det\_TKEO\_mean\_10\_coef*, *app\_LT\_entropy\_log\_1\_coef*, and *app\_LT\_entropy\_log\_5\_coef*).

The shimmer (cycle-to-cycle variation in amplitude) of voice signal is known to be higher in PD patients than healthy controls [48], [49], [50]. Shimmer has frequently been used as a measure of voice signal for PD. Mel Frequency Cepstral Coefficients (MFCCs) capture variation in both vocal folds

**Input**: a training set and a validation set; maximum number of PSO iterations;  $\gamma_A$  and  $\gamma_I$ 

**Initialization**: initialize the positions of *m* particles,  $\mathbf{r}_i^0$ , i = 1, ..., m

**Iterate**: At the *t*-th iteration and for each particle *i*, do the following:

- 1. Construct the feature subset corresponding to the position of the particle,  $\mathbf{R}_{i}^{t} = diag(\mathbf{r}_{i}^{t})$ ;
- 2. Construct a graph of all the labeled and unlabeled instances based on the selected features,  $G(\mathbf{R}_i^t)$ , and compute the graph Laplacian  $\Phi(\mathbf{R}_i^t)$ ;
- 3. Convert  $G(\mathbf{R}_i^t)$  into an unweighted graph using KNN;
- Solve the integer programming optimization in (3) to get the optimal instance subset under this feature subset, S(R<sup>t</sup><sub>i</sub>);
- 5. Solve the predictive function under this instance subset and feature subset using (6), denoted by  $f(\mathbf{R}_i^t, \mathbf{S}(\mathbf{R}_i^t))$ ;
- 6. Apply the predictive function on the validation set and compute the prediction accuracy;
- 7. Update the velocity of this particle, i.e.,  $\mathbf{v}_i^t$ , using (7);
- 8. Compute the position of the particle in the next iteration,  $\mathbf{r}_{i}^{t+1}$ , using (7), and go to the next iteration.

**Output:**  $f^*$ ,  $\mathbf{R}^*$ , and  $\mathbf{S}^*(\mathbf{R}^*)$  corresponding to the highest validation accuracy—found when the error is no longer decreasing after a predefined number of iterations or when the maximum available number of iterations has been reached (t = N).

and the vocal tract (i.e., tongue, lips, jaw, etc.). PD research has demonstrated that, in addition to the vocal folds that traditional measures capture, articulators of the vocal tract (i.e., tongue, lips, jaw, etc.) are affected by the disease [51]. Wavelet measures are derived from the discrete wavelet transform (DWT), which can quantify both regularity effects (scale aspects) and transient processes (time aspects) [52]. DWT decomposes the wavelet signal into detail information (detail coefficients) and course approximation (approximation coefficients). The main rationale for wavelet measures is that people with pathological voices cannot sustain a vowel with minimum deviation from exact periodicity, while healthy controls can [53].

Table II additionally shows the performance of S2SSL relative to three semi-supervised learning benchmarks in the recent literature—namely, Semi-Supervised Learning Random Forest Tree Ensembles (SSL-RF) [116], Multi-scheme Semi-Supervised Regression Approach (MSRRA) [117], and Graph-based Semi-Supervised Sparse Feature Selection (GS<sup>3</sup>FS) [75]. Results reported include both simulation datasets and the PD Telemonitoring dataset.

# D. Discussion on Utilities of the Results for Health Care Automation

There are multi-fold utilities: 1) For a patient with PD, a predicted MDS-UPDRS score that reflects his/her disease severity can be generated as soon as the patient performs activities (e.g., tapping, speaking) guided by the smartphone app. This prediction can be remotely shared with his/her physician to make timely medical decisions. Since the patient can almost perform the activities "anytime anywhere", his/her disease progression can be closely monitored and thus better controlled. 2) With the aid of the proposed telemonitoring method, patients and medical specialists do not have to be in physical proximity for the patients to receive medical advice. This will greatly improve patient access to advanced health care resources, especially for individuals living in resourcepoor regions or counties. 3) Using the predicted MDS-UPDRS as a surrogate, hospitals and clinics can design more effective and efficient patient triage systems. This will facilitate better decision making regarding which patients should come to a physical clinic and at what time in order to receive further evaluation and treatment. This will not only benefit patients but also prevent overloading the health care system and staff.

#### VII. CONCLUSION

We proposed a novel semi-supervised learning model, S2SSL, that allows for simultaneous feature and instance selection to improve model building from datasets with few labeled instances, many available features, and an abundance of unlabeled instances. Because there is no analytical solution for S2SSL, a parameter estimation method using particle swarm optimization and integer programming was also developed.

S2SSL was applied to the activity data of patients with PD collected by their smartphones and demonstrated excellent accuracy in predicting their disease severity (0.828 Pearson correlation on tapping and voice features). Clinically relevant features were also selected and provided more information about which features are more effective at predicting the MDS-UPDRS Parkinson's Disease severity score.

Future research includes the methodological extension to other types of response variables and alternative algorithms for solving the S2SSL formulation, as well as applications to the telemonitoring of other diseases, such as Alzheimer's Disease and post-traumatic headache.

#### REFERENCES

- Parkinson's News Today. (2020). Parkinson's Disease Statistics. [Online]. Available: https://parkinsonsnewstoday.com/parkinsonsdisease-statistics/
- [2] B. M. Bot et al., "The mPower study, Parkinson disease mobile data collected using ResearchKit," Sci. Data, vol. 3, no. 1, pp. 1–9, 2016.
- [3] C. G. Goetz et al., "Movement disorder society-sponsored revision of the unified Parkinson's disease rating scale (mds-updrs): Scale presentation and clinimetric testing results," *Movement Disorders, Off. J. Movement Disorder Soc.*, vol. 23, no. 15, pp. 2129–2170, 2008.
- [4] A. Holub, M. Welling, and P. Perona, "Exploiting unlabelled data for hybrid object classification," in *Proc. Neural Inf. Process. Syst.*, *Workshop Inter-Class Transf.*, vol. 7, 2005, p. 2.
- [5] A. Fujino, N. Ueda, and K. Saito, "A hybrid generative/discriminative approach to semi-supervised classifier design," in *Proc. AAAI*, 2005, pp. 764–769.
- [6] Y. Li, C. Guan, H. Li, and Z. Chin, "A self-training semi-supervised SVM algorithm and its application in an EEG-based brain computer interface speller system," *Pattern Recognit. Lett.*, vol. 29, no. 9, pp. 1285–1294, Jul. 2008.
- [7] J. Tanha, M. van Someren, and H. Afsarmanesh, "Semi-supervised self-training for decision tree classifiers," *Int. J. Mach. Learn. Cybern.*, vol. 8, no. 1, pp. 355–370, Feb. 2017.
- [8] X. Wan, "Co-training for cross-lingual sentiment classification," in Proc. 4th Int. Joint Conf. Natural Lang. Process., vol. 1, Aug. 2009, pp. 235–243.

12

IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING

- [9] Z.-H. Zhou, D.-C. Zhan, and Q. Yang, "Semi-supervised learning with very few labeled training examples," in *Proc. AAAI*, 2007, pp. 675–680.
- [10] X. Zhu and J. Lafferty, "Harmonic mixtures: Combining mixture models and graph-based methods for inductive and scalable semisupervised learning," in *Proc. 22nd Int. Conf. Mach. Learn.*, 2005, pp. 1052–1059.
- [11] H. Li, Y. Li, and H. Lu, "Semi-supervised learning with Gaussian processes," in *Proc. Chin. Conf. Pattern Recognit.*, Oct. 2008, pp. 753–760.
- [12] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, 2003, pp. 912–919.
- [13] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Jan. 2006.
- [14] J. Wang, T. Jebara, and S.-F. Chang, "Semi-supervised learning using greedy max-cut," J. Mach. Learn. Res., vol. 14, no. 1, pp. 771–800, Mar. 2013.
- [15] T. Jebara, J. Wang, and S.-F. Chang, "Graph construction and bmatching for semi-supervised learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 441–448.
- [16] S. Pettie and V. Ramachandran, "An optimal minimum spanning tree algorithm," J. Assoc. Comput. Machinery, vol. 49, no. 1, pp. 16–34, 2002.
- [17] Z. Lu, X. Gao, L. Wang, J.-R. Wen, and S. Huang, "Noise-robust semisupervised learning by large-scale sparse coding," in *Proc. AAAI Conf. Artif. Intell.*, 2015, vol. 29, no. 1, doi: 10.1609/aaai.v29i1.9551.
- [18] Z. Lu and L. Wang, "Noise-robust semi-supervised learning via fast sparse coding," *Pattern Recognit.*, vol. 48, no. 2, pp. 605–612, Feb. 2015.
- [19] M. Zhao, L. Jiao, J. Feng, and T. Liu, "A simplified low rank and sparse graph for semi-supervised learning," *Neurocomputing*, vol. 140, pp. 84–96, Sep. 2014.
- [20] S. Sun, Z. Hussain, and J. Shawe-Taylor, "Manifold-preserving graph reduction for sparse semi-supervised learning," *Neurocomputing*, vol. 124, pp. 13–21, Jan. 2014.
- [21] J. C. Ang, H. Haron, and H. N. A. Hamed, "Semi-supervised SVMbased feature selection for cancer classification using microarray gene expression data," in *Proc. Int. Conf. Ind., Eng. Appl. Appl. Intell. Syst.* Cham, Switzerland: Springer, 2015, pp. 468–477.
- [22] X. Song, J. Zhang, Y. Han, and J. Jiang, "Semi-supervised feature selection via hierarchical regression for web image classification," *Multimedia Syst.*, vol. 22, no. 1, pp. 41–49, Feb. 2016.
- [23] Z. Ma, F. Nie, Y. Yang, J. R. R. Uijlings, N. Sebe, and A. G. Hauptmann, "Discriminating joint feature analysis for multimedia data understanding," *IEEE Trans. Multimedia*, vol. 14, no. 6, pp. 1662–1672, Dec. 2012.
- [24] J. Ren, Z. Qiu, W. Fan, H. Cheng, and S. Y. Philip, "Forward semi-supervised feature selection," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining.* Cham, Switzerland: Springer, 2008, pp. 970–976.
- [25] B. Wang, Y. Jia, and S. Yang, "Forward semi-supervised feature selection based on relevant set correlation," in *Proc. Int. Conf. Comput. Sci. Softw. Eng.*, 2008, pp. 210–213.
- [26] F. Bellal, H. Elghazel, and A. Aussem, "A semi-supervised feature ranking method with ensemble learning," *Pattern Recognit. Lett.*, vol. 33, no. 10, pp. 1426–1433, 2012.
- [27] Y. Han, K. Park, and Y.-K. Lee, "Confident wrapper-type semisupervised feature selection using an ensemble classifier," in *Proc. 2nd Int. Conf. Artif. Intell., Manage. Sci. Electron. Commerce (AIMSEC)*, Aug. 2011, pp. 4581–4586.
- [28] H. Barkia, H. Elghazel, and A. Aussem, "Semi-supervised feature importance evaluation with ensemble learning," in *Proc. IEEE 11th Int. Conf. Data Mining*, Dec. 2011, pp. 31–40.
- [29] S. J. Wright, "Coordinate descent algorithms," Math. Program., vol. 151, no. 1, pp. 3–34, Jun. 2015.
- [30] C. S. Adjiman, S. Dallwig, C. A. Floudas, and A. Neumaier, "A global optimization method, αBB, for general twice-differentiable constrained NLPs—I. Theoretical advances," *Comput. Chem. Eng.*, vol. 22, no. 9, pp. 1137–1158, Aug. 1998.
- [31] C. D. Maranas and C. A. Floudas, "Finding all solutions of nonlinearly constrained systems of equations," *J. Global Optim.*, vol. 7, no. 2, pp. 143–182, Sep. 1995.

- [32] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter, "Branching and bounds tighteningtechniques for non-convex MINLP," *Optim. Methods Softw.*, vol. 24, nos. 4–5, pp. 597–634, Oct. 2009.
- [33] S. Vigerske, "Decomposition in multistage stochastic programming and a constraint integer programming approach to mixed-integer nonlinear programming," Ph.D. dissertation, 2013. [Online]. Available: https://edoc.hu-berlin.de/bitstream/handle/18452/17356/vigerske. pdf?sequence=1
- [34] A. Land and A. Doig, "An automatic method of solving discrete programming problems," *Econometrica*, vol. 28, no. 3, pp. 497–520, Jul. 1960.
- [35] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Proc. Int. Conf. Comput. Learn. Theory.* Cham, Switzerland: Springer, 2001, pp. 416–426.
- [36] X. Chu, Q. Lu, B. Niu, and T. Wu, "Solving the distribution center location problem based on multi-swarm cooperative particle swarm optimizer," in *Proc. Int. Conf. Intell. Comput.* Cham, Switzerland: Springer, 2012, pp. 626–633.
- [37] B. Jarboui, N. Damak, P. Siarry, and A. Rebai, "A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems," *Appl. Math. Comput.*, vol. 195, no. 1, pp. 299–308, Jan. 2008.
- [38] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intell.*, vol. 1, no. 1, pp. 33–57, Jun. 2007.
- [39] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proc. Congr. Evol. Comput.*, vol. 1, Jul. 2000, pp. 84–88.
- [40] F. V. D. Bergh et al., "An analysis of particle swarm optimizers," Doctoral dissertation, Univ. Pretoria, Pretoria, South Africa, 2001.
- [41] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, Oct. 2011.
- [42] A. L. T. Tavares et al., "Quantitative measurements of alternating finger tapping in Parkinson's disease correlate with updrs motor disability and reveal the improvement in fine motor control from medication and deep brain stimulation," *Movement Disorders, Off. J. Movement Disorder Soc.*, vol. 20, no. 10, pp. 1286–1298, 2005.
- [43] S. Arora et al., "Detecting and monitoring the symptoms of Parkinson's disease using smartphones: A pilot study," *Parkinsonism Rel. Disorders*, vol. 21, no. 6, pp. 650–653, 2015.
- [44] P. Kassavetis et al., "Developing a tool for remote digital assessment of Parkinson's disease," *Movement Disorders Clin. Pract.*, vol. 3, no. 1, pp. 59–64, 2016.
- [45] A. Tsanas, M. A. Little, P. E. McSharry, and L. O. Ramig, "Nonlinear speech analysis algorithms mapped to a standard metric achieve clinically useful quantification of average Parkinson's disease symptom severity," *J. Roy. Soc. Interface*, vol. 8, no. 59, pp. 842–855, Jun. 2011.
- [46] H. Yoon and J. Li, "A novel positive transfer learning approach for telemonitoring of Parkinson's disease," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 1, pp. 180–191, Jan. 2019.
- [47] D. R. Roalf et al., "Quantitative assessment of finger tapping characteristics in mild cognitive impairment, Alzheimer's disease, and Parkinson's disease," *J. Neurol.*, vol. 265, no. 6, pp. 1365–1375, 2018.
- [48] L. A. Ramig, I. R. Titze, R. C. Scherer, and S. P. Ringel, "Acoustic analysis of voices of patients with neurologic disease: Rationale and preliminary data," *Ann. Otol., Rhinol. Laryngol.*, vol. 97, no. 2, pp. 164–172, Mar. 1988.
- [49] I. Hertrich and H. Ackermann, "Gender-specific vocal dysfunctions in Parkinson's disease: Electroglottographic and acoustic analyses," Ann. Otol., Rhinol. Laryngol., vol. 104, pp. 197–202, Mar. 1995.
- [50] J. Jiang, E. Lin, J. Wang, and D. G. Hanson, "Glottographic measures before and after levodopa treatment in Parkinson's disease," *Laryngoscope*, vol. 109, no. 8, pp. 1287–1294, Aug. 1999.
- [51] A. K. Ho, R. Iansek, C. Marigliani, J. L. Bradshaw, and S. Gates, "Speech impairment in a large sample of patients with Parkinson's disease," *Behavioural Neurol.*, vol. 11, no. 3, pp. 131–137, 1999.
- [52] A. Tsanas, "Accurate telemonitoring of Parkinson's disease symptom severity using nonlinear speech signal processing and statistical machine learning," Ph.D. dissertation, Oxford Univ., Oxford, U.K., 2012.
- [53] I. R. Titze and D. W. Martin, "Principles of voice production," J. Acoust. Soc. Amer., vol. 104, p. 1148, 1998, doi: 10.1121/1.424266.

- [54] J. Duan, B. Luo, and J. Zeng, "Semi-supervised learning with generative model for sentiment classification of stock messages," *Expert Syst. Appl.*, vol. 158, Nov. 2020, Art. no. 113540.
- [55] Y. Zou, Z. Yu, X. Liu, B. V. K. V. Kumar, and J. Wang, "Confidence regularized self-training," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.* (*ICCV*), Oct. 2019, pp. 5982–5991.
- [56] D. Kim, D. Seo, S. Cho, and P. Kang, "Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2 Vec," *Inf. Sci.*, vol. 477, pp. 15–29, Mar. 2019.
- [57] V. Verma et al., "Interpolation consistency training for semi-supervised learning," 2019, arXiv:1903.03825.
- [58] J. Xie, S. Liu, and H. Dai, "A distributed semi-supervised learning algorithm based on manifold regularization using wavelet neural network," *Neural Netw.*, vol. 118, pp. 300–309, Oct. 2019.
- [59] H. Zhao, J. Zheng, W. Deng, and Y. Song, "Semi-supervised broad learning system based on manifold regularization and broad network," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 3, pp. 983–994, Mar. 2020.
- [60] A. A. Deshmukh and E. Laftchiev, "Semi-supervised transfer learning using marginal predictors," in *Proc. IEEE Data Sci. Workshop (DSW)*, Jun. 2018, pp. 160–164.
- [61] A. Gogna, A. Majumdar, and R. Ward, "Semi-supervised stacked label consistent autoencoder for reconstruction and analysis of biomedical signals," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 9, pp. 2196–2205, Sep. 2017.
- [62] B. Jiang, X. Wu, K. Yu, and H. Chen, "Joint semi-supervised feature selection and classification through Bayesian approach," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 3983–3990.
- [63] R. Sheikhpour, M. A. Sarram, S. Gharaghani, and M. A. Z. Chahooki, "A survey on semi-supervised feature selection methods," *Pattern Recognit.*, vol. 64, pp. 141–158, Apr. 2017.
- [64] K. Liu, X. Yang, H. Yu, J. Mi, P. Wang, and X. Chen, "Rough set based semi-supervised feature selection via ensemble selector," *Knowl.-Based Syst.*, vol. 165, pp. 282–296, Feb. 2019.
- [65] Y. Gu, K. Li, Z. Guo, and Y. Wang, "Semi-supervised K-means DDoS detection method using hybrid feature selection algorithm," *IEEE Access*, vol. 7, pp. 64351–64365, 2019.
- [66] C. Shi, C. Duan, Z. Gu, Q. Tian, G. An, and R. Zhao, "Semi-supervised feature selection analysis with structured multiview sparse regularization," *Neurocomputing*, vol. 330, pp. 412–424, Feb. 2019.
- [67] Pew Research Center. (2019). Mobile Fact Sheet. [Online]. Available: https://www.pewresearch.org/internet/fact-sheet/mobile
- [68] G. Wahba, Spline Models for Observational Data. Philadelphia, PA, USA: SIAM, 1990.
- [69] T. J. Hastie and R. J. Tibshirani, *Generalized Additive Models*, vol. 43. Boca Raton, FL, USA: CRC Press, 1990.
- [70] M. Culp and G. Michailidis, "An iterative algorithm for extending learners to a semi-supervised setting," *J. Comput. Graph. Statist.*, vol. 17, no. 3, pp. 545–571, Sep. 2008.
  [71] M. Culp, "On propagated scoring for semisupervised additive
- [71] M. Culp, "On propagated scoring for semisupervised additive models," J. Amer. Stat. Assoc., vol. 106, no. 493, pp. 248–259, Mar. 2011.
- [72] J. J. Elm et al., "Feasibility and utility of a clinician dashboard from wearable and mobile application Parkinson's disease data," *NPJ Digit. Med.*, vol. 2, no. 1, pp. 1–6, 2019.
- [73] H. Zhang et al., "PDMove: Towards passive medication adherence monitoring of Parkinson's Disease using smartphone-based gait assessment," in *Proc. ACM Interact. Mobile, Wearable Ubiquitous Technol.*, 2019, vol. 3, no. 3, pp. 1–23.
- [74] C. Lo et al., "Predicting motor, cognitive & functional impairment in Parkinson's," Ann. Clin. Neurol., vol. 6, no. 8, pp. 1498–1509, 2019.
- [75] R. Sheikhpour, M. A. Sarram, S. Gharaghani, and M. A. Z. Chahooki, "A robust graph-based semi-supervised sparse feature selection method," *Inf. Sci.*, vol. 531, pp. 13–30, Aug. 2020.
- [76] X. Zhang and M. Kschischo, "MFmap: A semi-supervised generative model matching cell lines to tumours and cancer subtypes," *PLoS ONE*, vol. 16, no. 12, Dec. 2021, Art. no. e0261183.
- [77] M. M. Karim, R. Qin, G. Chen, and Z. Yin, "A semi-supervised self-training method to develop assistive intelligence for segmenting multiclass bridge elements from inspection videos," *Struct. Health Monitor.*, vol. 21, no. 3, pp. 835–852, May 2022.
- [78] Y. Liang, Z. Liu, and W. Liu, "A co-training style semi-supervised artificial neural network modeling and its application in thermal conductivity prediction of polymeric composites filled with BN sheets," *Energy AI*, vol. 4, Jun. 2021, Art. no. 100052.

- [79] L. Xu et al., "Semi-supervised multi-layer convolution kernel learning in credit evaluation," *Pattern Recognit.*, vol. 120, Dec. 2021, Art. no. 108125.
- [80] Z. Song, X. Yang, Z. Xu, and I. King, "Graph-based semi-supervised learning: A comprehensive review," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 18, 2022, doi: 10.1109/TNNLS.2022.3155478.
- [81] W. Zhong, X. Chen, F. Nie, and J. Z. Huang, "Adaptive discriminant analysis for semi-supervised feature selection," *Inf. Sci.*, vol. 566, pp. 178–194, Aug. 2021.
- [82] V. Feofanov, E. Devijver, and M.-R. Amini, "Wrapper feature selection with partially labeled data," in *Applied Intelligence*. Berlin, Germany: Springer-Verlag, 2022, doi: 10.1007/s10489-021-03076-w.
- [83] J. M. Moreira de Lima and F. M. Ugulino de Araujo, "Ensemble deep relevant learning framework for semi-supervised soft sensor modeling of industrial processes," *Neurocomputing*, vol. 462, pp. 154–168, Oct. 2021.
- [84] M. Long, J. Wang, G. Ding, S. J. Pan, and P. S. Yu, "Adaptation regularization: A general framework for transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1076–1089, May 2014.
- [85] S. J. Pan, J. T. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction," in *Proc. AAAI*, vol. 8, Jul. 2008, pp. 677–682.
- [86] S. Pang et al., "Direct automated quantitative measurement of spine by cascade amplifier regression network with manifold regularization," *Med. Image Anal.*, vol. 55, pp. 103–115, Jul. 2019.
- [87] S. Si, D. Tao, and B. Geng, "Bregman divergence-based regularization for transfer subspace learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 7, pp. 929–942, Jul. 2010.
- [88] B. Tan, E. Zhong, E. W. Xiang, and Q. Yang, "Multi-transfer: Transfer learning with multiple views and multiple sources," in *Proc. SIAM Int. Conf. Data Mining*, May 2013, pp. 243–251.
- [89] P. Gardner, X. Liu, and K. Worden, "On the application of domain adaptation in structural health monitoring," *Mech. Syst. Signal Process.*, vol. 138, Apr. 2020, Art. no. 106550.
- [90] J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang, and P. S. Yu, "Visual domain adaptation with manifold embedded distribution alignment," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 402–410.
- [91] C. Wang and S. Mahadevan, "Heterogeneous domain adaptation using manifold alignment," in *Proc. IJCAI*, 2011, vol. 22, no. 1, p. 1541.
- [92] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.
- [93] L. Duan, I. W. Tsang, D. Xu, and T.-S. Chua, "Domain adaptation from multiple sources via auxiliary classifiers," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 289–296.
- [94] H. Liu, Z. Wu, X. Li, D. Cai, and T. S. Huang, "Constrained nonnegative matrix factorization for image representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1299–1311, Jul. 2012.
- [95] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.
- [96] D. Cai, X. He, X. Wu, and J. Han, "Non-negative matrix factorization on manifold," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 63–72.
- [97] S. Abu-El-Haija et al., "MixHop: Higher-order graph convolutional architectures via sparsified neighborhood mixing," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 21–29.
- [98] J. Ma, P. Cui, K. Kuang, X. Wang, and W. Zhu, "Disentangled graph convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4212–4221.
- [99] Q. Li, Z. Han, and X. M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 3538–3545, Art. no. 433.
- [100] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, arXiv:1609.02907.
- [101] P. Rodröguez, I. Laradji, A. Drouin, and A. Lacoste, "Embedding propagation: Smoother manifold for few-shot classification," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 121–138.
- [102] X. Zhao, M. Jia, and M. Lin, "Deep Laplacian auto-encoder and its application into imbalanced fault diagnosis of rotating machinery," *Measurement*, vol. 152, Feb. 2020, Art. no. 107320.

14

- IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING
- [103] F. Feng, X. He, J. Tang, and T.-S. Chua, "Graph adversarial training: Dynamically regularizing based on graph structure," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 6, pp. 2493–2504, Jun. 2021.
- [104] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5115–5124.
- [105] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, arXiv:1710.10903.
- [106] X. Li et al., "DeepSaliency: Multi-task deep neural network model for salient object detection," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3919–3930, Aug. 2016.
- [107] J. Weston, F. Ratle, and R. Collobert, "Deep learning via semisupervised embedding," in *Proc. 25th Int. Conf. Mach. Learn.* Berlin, Germany: Springer, 2008, pp. 639–655.
- [108] J. Hu, Y. Li, W. Gao, and P. Zhang, "Robust multi-label feature selection with dual-graph regularization," *Knowl.-Based Syst.*, vol. 203, Sep. 2020, Art. no. 106126.
- [109] J. Zhang, Z. Luo, C. Li, C. Zhou, and S. Li, "Manifold regularized discriminative feature selection for multi-label learning," *Pattern Recognit.*, vol. 95, pp. 136–150, Nov. 2019.
- [110] X. Zhu et al., "A novel relational regularization feature selection method for joint regression and classification in AD diagnosis," *Med. Image Anal.*, vol. 38, pp. 205–214, May 2017.
- [111] X. He, M. Ji, C. Zhang, and H. Bao, "A variance minimization criterion to feature selection using Laplacian regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 2013–2025, Oct. 2011.
- [112] M. Slawski, W. Zu Castell, and G. Tutz, "Feature selection guided by structural information," *Ann. Appl. Statist.*, vol. 4, no. 2, pp. 1056–1080, 2010.
- [113] N. G. Trillos, Z. Kaplan, T. Samakhoana, and D. Sanz-Alonso, "On the consistency of graph-based Bayesian semi-supervised learning and the scalability of sampling algorithms," *J. Mach. Learn. Res.*, vol. 21, no. 28, pp. 1–47, 2020.
- [114] L. Liu, W. Huang, B. Liu, L. Shen, and C. Wang, "Semisupervised hyperspectral image classification via Laplacian least squares support vector machine in sum space and random sampling," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 4086–4100, Nov. 2018.
- [115] G. Puy, N. Tremblay, R. Gribonval, and P. Vandergheynst, "Random sampling of bandlimited signals on graphs," *Appl. Comput. Harmon. Anal.*, vol. 44, no. 2, pp. 446–475, Mar. 2018.
- [116] J. Levatić, M. Ceci, T. Stepišnik, S. Džeroski, and D. Kocev, "Semisupervised regression trees with application to QSAR modelling," *Expert Syst. Appl.*, vol. 158, Nov. 2020, Art. no. 113569.
- [117] N. Fazakis, S. Karlos, S. Kotsiantis, and K. Sgarbas, "A multi-scheme semi-supervised regression approach," *Pattern Recognit. Lett.*, vol. 125, pp. 758–765, Jul. 2019.



Nathan Gaw (Member, IEEE) received the B.S.E. and M.S. degrees in biomedical engineering and the Ph.D. degree in industrial engineering from Arizona State University (ASU), Tempe, AZ, USA, in 2013, 2014, and 2019, respectively. He is currently an Assistant Professor with the Department of Operational Sciences, Air Force Institute of Technology, Wright-Patterson AFB, OH, USA. His research interests include multimodality fusion and semi-supervised learning in telemonitoring, military applications, and healthcare imaging.

He is a member of IISE and INFORMS.



Jing Li (Member, IEEE) received the Ph.D. degree in industrial and operations engineering from the University of Michigan, Ann Arbor, MI, USA. She is currently a Professor with the H. Milton Stewart School of Industrial and Systems Engineering (ISyE), Georgia Institute of Technology, Atlanta, GA, USA. Her research interests include statistical modeling and machine learning for health care applications. She was a recipient of the NSF Career Award. She is a member of IISE and INFORMS.



**Hyunsoo Yoon** (Member, IEEE) received the B.S. and M.S. degrees in industrial engineering from Korea University in 2010 and 2012, respectively, the M.S. degree in statistics from the Georgia Institute of Technology in 2013, and the Ph.D. degree in industrial engineering from Arizona State University in 2018. He is currently an Assistant Professor with the Department of Industrial Engineering, Yonsei University, Seoul, South Korea. His research interests include practice, methodology, and theory of statistical machine learning and deep learning. He is

a member of IISE and INFORMS.