

Machine learning for large-scale wearable sensor data in Parkinson disease: concepts, promises, pitfalls and futures

Ken J Kubota, BS, SEP¹, Jason A. Chen, BSE^{2,3} and Max A. Little, PhD^{4,5}

- 1: Department of Data Science, tranSMART Foundation, 401 Edgewater Place, Suite 600, Wakefield, MA 01880, US
- 2: Verge Genomics, 450 Mission Street, Suite 201, San Francisco, 94105, US
- 3: Interdepartmental Program in Bioinformatics, University of California at Los Angeles, 695 Charles E. Young Drive South, Los Angeles, CA 90095, US
- 4: Aston University, Aston Triangle, Birmingham, B4 7ET, UK
- 5: Media Lab, Massachusetts Institute of Technology, Cambridge, MA, US

Corresponding Author: Ken J Kubota, Department of Data Science, tranSMART Foundation, 401 Edgewater Place, Suite 600, Wakefield MA 01880, United States

Tel.: +1-415 235-2914, E-Mail: Ken.Kubota@transmartfoundation.org

Abstract

For the treatment and monitoring of Parkinson disease (PD) to be scientific, a key requirement is that measurements of disease stages and severity are quantitative, reliable and repeatable. The last 50 years in PD research have been dominated by qualitative, subjective ratings obtained by human interpretation of the presentation of disease features at clinical visits. More recently, “wearable”, sensor-based, quantitative, objective, and easy-to-use systems for quantifying PD signs for large numbers of participants over extended durations have been developed. This technology has the potential to significantly improve both clinical diagnosis and management in PD and the conduct of clinical studies. However, the large-scale, high-dimensional character of the data captured by these wearable sensors requires sophisticated signal processing and machine learning algorithms to transform it into scientifically and clinically meaningful information. Such algorithms that “learn” from data have shown remarkable success at making accurate predictions for complex problems where human skill has been required to-date, but they are challenging to evaluate and apply without a basic understanding of the underlying logic upon which they are based. This article contains a non-technical, tutorial review of relevant machine learning algorithms, also describing their limitations and how these can be overcome. It discusses implications of this technology and a practical roadmap for realizing the full potential of this technology in PD research and practice.

Introduction

Medical practice aspires to diagnose patients at the earliest of clinical signs; to monitor disease progression and rapidly find optimal treatment regimens. Clinical scientists and drug developers seek to enroll large numbers of participants into trials with minimal cost and effort, to maximize scientific validity of studies. Patients wish to increase their quality of life while reducing physical clinic visits, and patient care seeks to minimize reliance on the clinic and transition into patient’s homes. In this article we describe a combination of technologies – wearable devices and machine learning – which

may be key to these aspirations. These technologies promise to enable this vision by providing objective, high-frequency, sensitive and continuous data on the signs of PD.

Machine learning systems have become a pervasive part of modern life. As examples, mail delivery corporations use machine vision algorithms to automatically transcribe hand written addresses on envelopes to route them to their intended destination, and vehicle license plate reading algorithms automatically track vehicles around road networks based upon images captured using roadside digital cameras.¹ In both applications, without machine learning it would require human skill and constant attention to carry out the transcription, but machine learning software performs this task to almost human levels of accuracy, but with far superior speed and reliability.

Meanwhile, consumer devices such as cell phones, smartphones and more recently smartwatches are worn and used by nearly a third of the world's population on a daily basis.² These devices are fully-featured, Internet and telephone network connected wearable computers ("wearables") incorporating numerous digital sensors measuring physical quantities of the wearer and their environment. They put the vast combined utility of cell phones, laptops and desktop computers in the hands of the wearer, 24 hours a day, in almost any environment.

These twin technologies – wearables and machine learning – have developmental histories driven by rapid advances in hardware miniaturization, computing power, mass data storage, Internet-enabled information accessibility, and conceptual advances in computer science, software engineering and mathematics. These advances have reached the point at which wearable devices with machine learning, might allow cheap, reliable, validated disease process-relevant measurements, which would normally only be collected at infrequent, physical clinic or research center visits, if at all.

Wearable sensor data is complex because it is *high frequency* (rates of hundreds to thousands of observations per second) and often *high-dimensional* (many different sensors capturing multiple forms of data simultaneously) and so it is *high volume*. Machine learning methods are widely applicable to many kinds of data, but wearable sensor data is particularly difficult to visualize, understand and manage without specialized algorithms such as those used to process and analyze digital sensor data collected in other industrial applications such as mail sorting or speech recognition.

Since these technologies have only really matured in the last 10 years, they are unlikely to have formed part of the traditional training of clinicians and clinical researchers. The objective of this article is to present what machine learning is, the terminology used and its relationship to that used in traditional biomedical statistics (Table 2) and how it may be applied to wearable sensor data in clinical PD measurement, provide a short tutorial on the most relevant machine learning methods, and describe the major pitfalls and limitations of machine learning-based approaches to sensor data analysis, and how these limitations can be mitigated.

Defining wearables and statistical machine learning

Here we define a *wearable device/wearable* as an electronic device which is small, easily and comfortably worn, for extended periods of time, on some part(s) of the body. The device contains digital *sensors* measuring particular physical parameters such as acceleration, light flux, sound pressure, skin temperature or blood volume. We include 'quasi-wearable' devices such as smartphones within this definition. Wearables can be *consumer* or *medical*. Consumer refers to devices with no specific clinical function but which can be reprogrammed to perform clinical measurement (*clinimetric*) functions. Medical devices are designed and marketed for specific clinical purposes.

Machine learning is usually defined as the application of mathematical algorithms that can find arbitrary patterns or structure in data, and make predictions for new input data.³ Unlike traditional programming where complex algorithms are designed and programmed to produce exactly specified outputs, machine learning attempts to “self-program” from data alone, mimicking the human ability to synthesize rules from data.

The distinct character of wearable sensor data versus traditional clinical neurology data

PD wearable data is radically different from traditional clinical data, which is a ‘static’, small-scale, subjective snapshot of PD status. Traditional data is often comprised of demographic attributes (for example age and gender), and disease data (such as date of first symptom onset, diagnostic category, and ordinal-scale UPDRS questions).⁴ It takes substantial time and effort to collect such data, so even the most detailed studies collect at most 200 or so data items at each patient visit, with months between visits.

By contrast, wearable data comprises ‘continuous’ high-frequency digital sensor readings, potentially tens of thousands per second, reporting quantities such as acceleration, rate of rotation, blood volume, acoustic pressure, skin temperature, GPS coordinates, magnetic field strength and ambient light flux.

Of course, traditional data has merits: for example, a clinical rater, although inherently subjective, synthesizes vast amounts of sensory information about the subject, and can actively seek out confirmatory data from disparate sources in order to reach a holistic diagnosis or disease severity assessment.

Example large-scale wearables studies in PD research: detecting medication responsiveness

Here we pick out two large-scale consumer wearable studies providing an indication of what might be achievable in future.

The “Smartphone-PD” global observational study recruits healthy and PD participants using software downloaded to participant’s own Android smartphones.⁵ As of 2015, 457 individuals were recruited and over 46,000 hours of raw sensor data captured for 6 months’ duration from each participant, including GPS location, accelerometry, gyroscope, magnetometer, proximity and ambient light flux. Sensor data from short (less than 5 minute) controlled, systematic tests of gait, voice, touchscreen tapping, postural and rest tremor conducted using the smartphone allowed a machine learning algorithm to discriminate drug-responsive from non-responsive participants,⁶ and to quantify the level of responsiveness to treatments against l-dopa equivalent dosage. Similar results were found in the US-wide “mPower” iPhone-based study of PD.⁷ Here, touchscreen tapping sensor data from 57 PD participants was analyzed using various machine learning algorithms. Individualized medication responsiveness was also quantifiable using this tapping data.

This evidence for the clinical utility of this approach is promising: drug titration and timing is a delicate problem in PD disease management which might be substantially improved by real-time, objective and validated assessment of potential new treatments for PD. Many other smaller-scale studies have used machine learning for wearable device data in PD (Table 1).

Machine learning: a brief tutorial

Statistical machine learning methods subsume most traditional statistical methods used in biomedicine, for example: parametric and non-parametric null hypothesis testing, linear and logistic regression, discriminant analysis, principal components, factor analysis and cluster analysis. Typically,

machine learning extends these methods to cope with high-dimensionality and nonlinearity which is of particular importance in wearable sensor data. It overlaps with artificial intelligence but the problems it seeks to solve are usually recognizable to traditional biomedical statistics (Table 2). Statistical machine learning is one of the fastest-growing scientific disciplines; we describe the primary conceptual ‘branches’ and their importance in PD wearable data analysis.

Signal processing: data conditioning and feature extraction

A prerequisite of most machine learning approaches to sensor data analysis is the preparation of the digital data by identifying useful regions to process (segmentation). This signal processing is usually essential because in practice, sensor data is not always meaningful, for example, if the wearable is not actually being worn.⁸ Once reliably segmented, these data can then be summarized into a small set of *features*, which are then input to the machine learning algorithm.¹ Feature extraction makes the machine learning problem tractable, because it substantially reduces the number of data dimensions.

Supervised machine learning

Supervised learning *trains* an algorithm to *mimic* some input-output relationship (Figure 1A). Such methods require *labeled* data: each training input must be associated with an output value. We have examples of input paired with output data, and the algorithm finds a rule which maps the input data onto the output. The pattern to be learned is this rule, which can be used to make predictions when given new inputs. Depending upon the kind of output data – whether it takes a continuous range of values or a finite discrete set – the mathematical mapping is known as *regression* or *classification*, respectively.

Regression: As an example, consider output data where the labels represent clinically-scored severity of a PD sign (say, on the scale 0-100), and some (simplified) features related to that sign from a wearable device. In this situation, regression can form an automatic way of predicting the clinical score using the device. This approach has been used to predict UPDRS ratings from digital voice recordings, accelerometers, and touch screens.⁹ For example, Stamatakis et al.¹⁰ demonstrate logistic regression to predict UPDRS from features of performance on a finger tapping test recorded by accelerometer.

In classical statistical techniques such as *linear regression* and *linear discriminant analysis* this mapping rule is *linear*: that is, the classification decision boundary or regression curve is a *hyperplane* (e.g. line or a plane and equivalent geometric object in higher dimensions). By contrast, many supervised machine learning algorithms are in principle capable of discovering *any* nonlinear relationship.¹ This makes them particularly suited to finding high-dimensional, complex rules. These kinds of relationships might be impossible to intuit, but they are abundant when dealing with biomedical data, so one is often forced to use machine learning algorithms.

Classification: This method finds significant applications separating (discriminating) between two or more discrete output values.¹ An archetypal application example of machine learning classification is diagnosis: discriminating between healthy and PD cases. By comparing the expert diagnosis (the labels) against the predicted diagnosis from the classifier, all the familiar metrics of diagnostic performance such as sensitivity, specificity, positive and negative predictive value, ROC and AUC values (for classifiers with probabilistic outputs) can be computed. However, machine learning classifiers can produce classifications from high-dimensional input data where they find a general *decision boundary* in the input space which separates the data into the separate classes (e.g. healthy versus PD). Example classifiers include the *support vector machine* (SVM; Figure 2A) which can in

principle find a boundary of any complexity; Patel et al.¹¹ use SVMs to identify patients with PD from accelerometer data.

Other examples of classifiers include the *artificial neural network* (Figure 2B) which mimics the gross hierarchical connectivity of biological neural circuitry¹; Jane et al.¹² used neural networks to detect PD gait disturbances from wearable accelerometers. More recent incarnations such as the *convolutional neural network* have shown state-of-the-art prediction accuracy in applications such as image and speech identification¹³, although their robustness to confounding factors is somewhat in doubt¹⁴.

Decision trees are another widely-used classification algorithm which finds a decomposition of the input space into high-dimensional, hierarchically nested rectangular regions, assigning an optimal class label per region, the decomposition represented as a tree-like diagram (hence the name). The technique is substantially enhanced in accuracy by randomization and resampling – using many different possible decompositions – leading to *random forest classification*. Arora et al.⁹ used random forests in discriminating patients with PD from controls from a broad variety of smartphone sensor data.

Unsupervised machine learning

Supervised methods require *labeled* data and this is very often a difficult situation to arrange, particularly so for large-scale wearable data. *Unsupervised* machine learning methods do not require labeled data as the method attempts to find the *parameters* of specific structure in the input data (Figure 1B).^{1,3}

(Unsupervised) *clustering* partitions the data into separate groups (*clusters*), each with representative characteristics. When making predictions, clustering techniques, much like classifiers, map an unknown input onto a discrete output value which is the identifier for the closest cluster to that input data. A classic example is *K-means* which attempts to group the data so as to minimize the total distance between the center of each cluster and each input data point. The learned parameters are the cluster centers, and the process creates clusters which are approximately spherical in the input space. *K-means* is fairly well established in PD studies which seek to identify sub-types of PD, such as those who are tremor-dominant versus those with rapid motor function decline and cognitive impairment.¹⁵

In clustering, each data point is assigned to only one cluster: clusters do not overlap. However, in many situations this is an unrealistic simplification; it is more accurate to presume that the clusters overlap. In techniques such as *mixture modeling* the clustering is assumed to be uncertain, so that each data point is given a *probability* of belonging to *every* cluster.³

Dimensionality reduction

Most PD wearable data comprises an enormous number of separate measurements so it is said to have a large *dimensionality*. However, it may be the case that the *intrinsic* dimensionality of the data is much smaller. Consider 3D data with (X,Y,Z) coordinates for each data point, but where the data actually all lies on a 2D plane embedded in this 3D space. Any point on that flat ‘object’ can be located by specifying just two coordinates (X, Y). *Dimensionality reduction* attempts to find such lower-dimensional data structures embedded in very high-dimensional input spaces.¹ Making predictions involves finding the corresponding coordinates of the higher-dimensional input data point on the lower-dimensional object.

Assuming linear relationships in the data, the classical *principal components analysis* (PCA) finds *hyperplanes*, that is, lines, planes and their higher-dimensional analogs. Linearity is also assumed in

independent components analysis (ICA), which makes less restrictive mathematical assumptions about the data.¹ By contrast, a large variety of *manifold learning* methods can find arbitrary, nonlinear low-dimensional geometric objects.¹⁶ In PD research PCA and related techniques have been used to find a few intrinsic dimensions of PD signs within high-dimensional UPDRS data.^{17, 18}

Advanced machine learning methods

Advances in machine learning research have led to a wide variety of other algorithms, we describe a few important ones here. For *partially labeled* data where only some of the output training data is available, *semi-supervised* techniques such *manifold regularization* can provide direct constraints on the structure of the data which helps to locate the classification decision boundary in the input space (Figure 1C).¹⁹ Most supervised algorithms can be applied to partially labeled through *iterative self-training* by substituting for the unknown outputs with predictions from the method trained on the labeled data. *Non-parametric Bayesian* methods such as *Gaussian processes* allow for finding not only the optimal parameters from the training data, but also the appropriate *number* of parameters. Thus, non-parametric Bayesian methods can adapt to the complexity of the data as more data is encountered.²⁰ For example, *Dirichlet process mixture models* in PD subtyping discover both the number of subtypes and the representative clinical characteristics of each subtype.²¹

Pitfalls and remedies of machine learning

Wearable PD data analysis will usually require statistical machine learning algorithms of varying sophistication to make clinical sense of this data. There are many traps to be avoided if wearables are to find practical use in clinical applications. Here we discuss several of these issues and how they might be addressed.

Overfitting and underfitting: finding the right model

Science is not just about collecting data, it is concerned with formalizing general hypotheses about *all* the relevant data, which can make predictions to be tested. Prediction accuracy is crucial to the practical value of the hypothesis. Thus one goal of science is to come up with models (hypotheses based on assumptions) which are at exactly the right level of simplicity/complexity to predict the relevant observed and unobserved data. As a familiar example, classical null hypothesis testing can be considered as a model selection problem in which we select between a simple null and a more complex alternative hypothesis. This principle of tuning the complexity of a model, also known as *Occam's razor*, has a counterpart in algorithms for wearable data analysis embodied in the complementary concepts of *overfitting* and *underfitting*.¹

An *overfitted* model is more complex than can be justified by the data. An overfitted model might have too many free parameters and thus risks confusing random noise or other confounds in the training data for genuine disease-related structure. This is a pervasive problem in statistical machine learning because the complexity of the model can often be set as high as required to get arbitrarily high prediction accuracy. An example: with *K*-means it is possible to get the prediction error as low as desired by simply increasing the number of clusters, *K*, and if each data point has its own cluster, zero prediction error can be obtained. Of course, this would not be a meaningful clustering, but it illustrates what can go wrong with careless use of machine learning algorithms. The opposite situation, where the model is too simple, is known as *underfitting*: such a model will be insensitive to the actual disease-related structure in the data, producing predictions with poor accuracy.

The (inevitably imperfect) remedy is appropriate *complexity control*, a critically important and weighty topic. The more robust machine learning algorithms have some kind of complexity control parameter (and Bayesian algorithms naturally incorporate complexity control through the “spread”

of the prior distribution over the model's parameters) but many can be adapted to use *sampling-based* approaches which we discuss below. However, there are inherent strengths and weaknesses to all forms of complexity control and no technique is guaranteed to find the ideal solution under all circumstances.^{1, 22}

Hold-out testing is a popular sampling approach to complexity control. The data is split into *testing* and *training* subsets, and the algorithm parameters are found from the training subset. The prediction accuracy is computed on the test subset. Thus, no information from the training subset is used in testing the algorithm. Since the test subset will always be a different random subset of the overall *population* data to the training subset, the machine learning technique is appropriately penalized for making overly accurate predictions on the training data. The rationale is that an algorithm which has perfect predictions for the training data has almost zero probability of making good predictions on the test data. In *cross-validation*, hold-out testing is repeated by randomized train/test splitting, and this leads to an estimate of the variability of the prediction error. This variability establishes a statistically meaningful confidence interval for the prediction error when predictions are made on the population data (which is unavailable in principle).

The major weakness of cross-validation is that it requires every measurement in the data (grouping all dimensions together into one measurement) to be (1) statistically independent of all the rest, and yet (2) share the same statistical distribution as all the rest of the measurements, and (3) this distribution must match the population distribution.²² These three strict conditions make the problem of *distributional mismatch* acute: for example, if there is any internal heterogeneity in the data, then each training subset is unlikely to contain all the information required to learn the overall structure in the data. As a result, the predictions on the test set are unlikely to be accurate and hold-out testing will give misleadingly low prediction accuracy. Unfortunately, identifying departures from these strict assumptions is, in the final analysis, as difficult as having a complete statistical model for the data, and this is, of course, exactly what we do not know and are trying to find!

Another serious weakness with hold-out approaches is that of *data leakage*, where some unintended effect causes the training subset to inadvertently contain information about the test subset, which can dramatically undermine the independence between training and test subsets required to make this a valid procedure.²³ It can be extremely hard to identify sources of data leakage because data can be confounded by all manner of unknown effects. This difficulty is compounded by the fact that there are often unintended interactions between the machine learning algorithm and the kind of leakage: some algorithms are less sensitive to leakage than others.

Another test of prediction performance is *replication*, sometimes called *true hold-out* testing, which is analogous to prospective testing. Here, a completely different data set (ideally collected in another lab or in a different sample of the population) is used to test the prediction performance of the machine learning technique trained on the original data set. Replication helps to avoid problems such as data leakage, but it can sharpen the deleterious influence of distributional mismatch, because it is extremely hard to guarantee a perfect match in experimental conditions across different experimental settings.

For sampling methods, replication testing is generally held to be the most faithful representation of how the machine learning method would perform for the population in general, followed by hold-out testing such as cross-validation. There are many different forms of cross-validation including: leaving out one sample or subject from the training set, or 90%-10% train/test splitting, and they all have different statistical properties which can entail differing estimates of the prediction error and confidence in these estimates.

The above suggests that data and algorithms go hand-in-hand: it is not possible to rely on wearable data alone – we need to supply some assumptions in order to make the data collection effort scientific. But this is why the choice of machine learning algorithm, and the underlying assumptions that each entails, is critical for the success of any machine learning approach to wearable data analysis; in other words, data alone do not suffice. Similarly, since there are inherent assumptions to any complexity control method which may not hold entirely true for the data it is necessary to compare the results obtained using several methods in order to gain a more comprehensive understanding of the performance of any machine learning algorithm.

Data quality issues

As discussed above, wearable data vastly exceeds the amount of data collected in classical clinical neurology applications, and there are many advantages to this. But, it is generally true that “big data” is not automatically better than small. This section examines some of the reasons why.

One of the most pertinent issues is the perennial problem of *sampling/selection bias*, which is an unavoidable issue in just about every statistical study.²⁴ The obvious population biases are uneven technology adoption in particular sub-populations, for example, the more vulnerable older PD patients are unlikely to have familiarity with contemporary technologies and so tend to select themselves out of wearable studies. Since they are likely to have advanced PD, wearable studies will have to be carefully planned and executed to avoid omitting the more severe PD cases. Similarly, there are behavioral sampling biases: imagine a wearable being charged and not worn during the “wearing off” period for a particular PD participant, that period is thereby omitted for that participant. These are obvious sampling biases, but there could be much more subtle biases that only come to light during replication attempts. These issues can be mitigated by the careful use of well-known population sampling designs.

One of the main advantages wearables is the possibility of continuous monitoring which can be scaled up to study sizes of thousands of individuals or larger, and to conduct studies in the “home setting” rather than the “lab”. The freedom to collect more realistic, unconstrained ambulatory data across large sections of the population is valuable but we lose a substantial amount of experimental control over the data collection. Studies conducted this way are fundamentally different to most classical clinical studies, in that participant behavior is largely unconstrained. So, we have little information from moment to moment about what participants are doing, with far greater opportunity for confounding effects to undermine any subsequent analysis. Analyzing behavioral data from wearables in PD is a nascent science, and there is still much that is unknown about how to establish vital contextual information from sensor data which allows us to take into account the effects of confounders. To mitigate this issue, it is suggested that multiple sources of sensor data should be collected so that context can be more easily established.

Another major problem with data collection is potential lack of accuracy of labels. For example, where participants are prompted to state when they have taken their medications using a smartphone app, they may fail to report, or they may enter incorrect times and dosages. Of course, this problem is not unique to wearable studies, but the problem may be more acute than traditional studies because identifying a few bad labels among a huge number may be very difficult indeed. Machine learning algorithms which are explicitly designed to handle errors in labels have more recently been developed and may be useful in this situation.

Data drift and non-stationarity

The vast majority of machine learning algorithms make the implicit assumption that the distribution of the training data is static and unchanging over time. But, one of the unavoidable facts of human

behavior is that it is adaptable to the circumstances, flexible and changeable. When machine learning algorithms are trained on a particular data set, they are trained to discover structure which is specific to that data as if it is a sample from the population distribution. But in practice, it is rare for the population distribution to remain static, and it is very likely to drift over time in response to changing environmental or personal conditions of the wearer. Therefore, by the time the machine learning system is deployed the distribution of the observed data may no longer match that of the training data, causing the machine learning predictions to be invalid.

There are several responses to this: simpler machine learning algorithms are less likely than more complex algorithms to be ‘brittle’ in the face of changing distributions, so that simpler methods can be preferred when data drift is expected.²⁵ The problem with this approach is that simpler methods often produce less accurate predictions, because they are underfitted. An alternative is to re-train the algorithm on the data as it is accumulated in deployment, but this may require new labeled data which is difficult to obtain, and/or increased computational and implementation complexity which may rule out immediate user feedback applications.

Lack of model interpretability

Machine learning algorithms have unprecedented *predictive power*. They can make high-accuracy predictions and discover structure and relationships in data which would be well beyond the reach of any unaided human analyst to intuit. However, these techniques do not generally supply *explanatory power*. In other words, they are “black boxes” which ingest data to produce predictive outputs.²⁶ There is a major disadvantage to such opacity: if the algorithm goes wrong, it is difficult to find out why. Also, the only guaranteed way to probe the full behavior of these black boxes is to explore all possible combinations of the inputs, which, for high-dimensional wearable data, is intractable. In other words, a method such as a convolutional neural network is a powerful prediction machine, but is of severely limited utility for facilitating *understanding* of the structure in the data.

This is a serious shortcoming because one of the main principles of science is explain structure in data, which can link up to existing knowledge and therefore corroborate, confirm, revise and extend our knowledge about reality. We are also missing a vital check on the validity of machine learning predictions which will have life-changing implications for patients if used to make clinical treatment decisions. Mitigating this problem requires careful attention to establishing the logical connections between the physical properties measured by the sensors, any features extracted and their relationship to the underlying physiology, and the mathematical structure of the machine learning algorithm used to make predictions.

Technical implementation complexity

One of the guiding principles of engineering is *modularity*: building a system out of parts which can be tested independently. Modularity allows us to isolate effects to particular components of the system and thus prevent failure in one part from causing catastrophic, system-wide failure. Most machine learning algorithms are, unfortunately, not modular at all. Small changes in one parameter can have substantial and largely unpredictable effects on all the rest of the model. All the input data must be in exactly the same form as it was when used to train the model.²⁷

These kind of strong dependencies and general ‘brittleness’ to even slight changes mean that the engineering complexity, particularly in system maintenance, is very high, and premature system design choices can have large knock-on consequences which cannot be addressed without considerable engineering effort. What this means in practice is that for wearable data, the route from data capture to algorithm development and into clinical deployment is far from easy, and it is

not clear at this stage whether the promising early results obtained in academic studies will actually translate to deployment.

Of course, this is something that occurs with any nascent technology, but with machine learning, the gap between academic studies on the intellectual frontier and final technical implementation is wide.²⁷ This is systemic to the academic enterprise because it is generally uninterested in the technical implementation details. Perhaps a useful analog is that machine learning algorithm research and development is like *basic research*, and technical implementation issues are like *translational research*, and as it stands at the moment, there is a disproportionate amount of effort and funding being put into the basic science, and little, if any, research into the translational aspects.

Options to address this issue include careful design and testing of each component algorithm in the machine learning system to ensure that each component is robust and can function effectively when not part of the overall system.

Technological difficulties arising

As this field is nascent there is still much to understand about how to maximize the effectiveness of wearable data. For example, the raw digital data from a single sensor can, by itself, be misleading – in the Fox Insight study²⁸, patients report that mowing the lawn or exercising can mimic a tremor, and derived smartphone assessments of gait parameters from this raw sensor data have also been inaccurate for participants living in a cluttered environment.

So, while continuous wearable data is objective, it needs effective statistical signal processing for data quality control and high-accuracy prediction algorithms to be scientifically valuable. The transformation of raw data into a form that accurately reflects PD disease severity and progression requires continued research, even as large-scale studies such as Fox Insight²⁸, Smartphone-PD⁵ and mPower⁷ continue to amass data.

Conclusions and Future Directions

From where we stand now with this technology, how will we get to a state where it fulfills its full potential in PD? We can identify several key steps: *clinimetric algorithm validation*, *open platform standardization and dissemination*, and *data sharing*.²⁸

Clinimetric validation is required for regulatory approval, much like the measurement of e.g. blood glycosylated hemoglobin or blood pressure. Over and above demonstrating ease of repetitive use, approval requires that (a) it provides an accurate parameter of a clinically relevant feature of the disease, (b) there is confirmed evidence that this parameter has an ecologically relevant response within some specific clinical application, and (c) a target numerical range exists where the parameter measures adequate treatment response. The specific clinimetric validation approach taken needs some care: although it is possible to use machine learning algorithms to map wearable sensor data onto existing clinical scales such as UPDRS⁹, this is a poor approach because wearable sensors can be far more sensitive and precise than subjective ratings. This suggests that we should look to other reliability measures such as test-retest, parallel forms or internal consistency testing.²⁹

Platform standardization requires that the software and algorithms are made publicly available for general and widespread use, because any technology lives or dies by the scale of its adoption and dissemination in the wider community. Since PD measurement is a niche technology application, it is unlikely to be profitably funded over the long term through private investment. However *open*

source communities can be committed to maintaining and improving on public interest projects such as this, substituting the need for investment with community-sourced ‘intellectual capital’.

Finally, science demands continual replication and confirmation on new data, for which data sharing is key. At the same time as learning more about PD, by sharing data we can further refine algorithms and provide increasingly extensive and rigorous clinimetric validation evidence. This will require careful planning at the initial stages of clinical trial design using this technology, since participants need to be consented to share the data widely. There are many other considerations that arise from this requirement, for example, dealing with data privacy issues. These must also be worked through in detail.

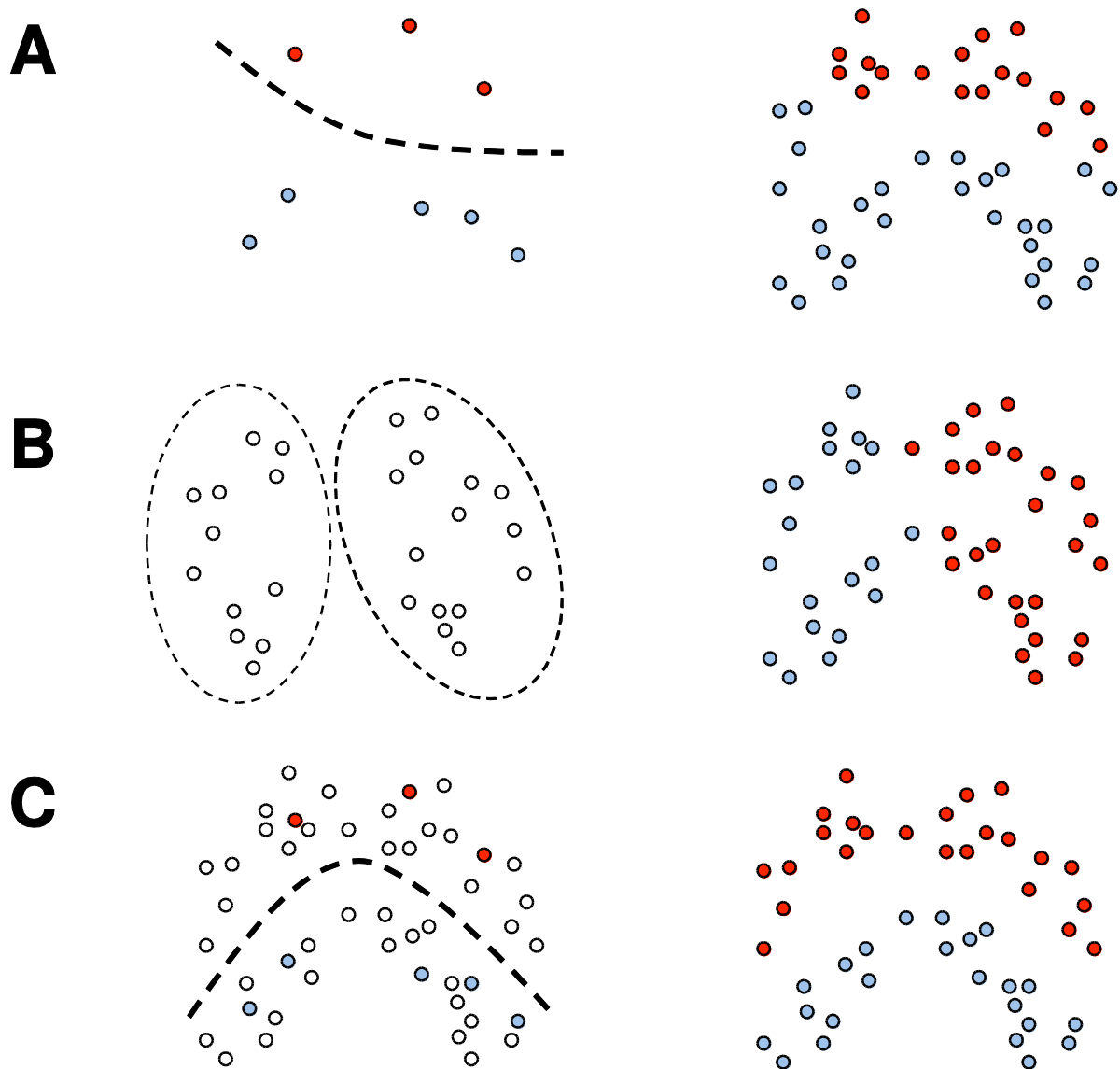


Figure 1: A graphical depiction of how machine learning algorithms can process input data from wearables in PD. In each panel the horizontal and vertical dimensions on the page represent values of “features” (see main text) of the wearable sensor data. For example, in a two-dimensional space as depicted here, each point is associated with two features, one for the horizontal and one for the vertical dimension. (A) Supervised methods use labeled training data (left, labels correspond to red and blue coloring) to make predictions about new data. Here, a supervised classifier learned on the training data is used to categorize new data in either the blue or red class (right). The learned class

decision boundary is depicted by the dashed line. (B) In PD wearable applications, labeled training data can be difficult to acquire but unlabeled training data is often available in large quantities. Unsupervised learning methods can use this unlabeled data (left) to learn the parameters of the relationship between data points. Here, a clustering algorithm identifies subgroups of the data (right, dashed ovals). This grouping implicitly separates the entire space of the data into two non-overlapping regions assigned to each class, which can then be used to classify new input data, as with supervised classification. (C) Semi-supervised learning is a “blend” of supervised and unsupervised learning methods, using both labeled and unlabeled training data (left) to strengthen its class predictions (right). This allows full exploitation of datasets with both labeled and unlabeled data, a common occurrence with PD wearables.

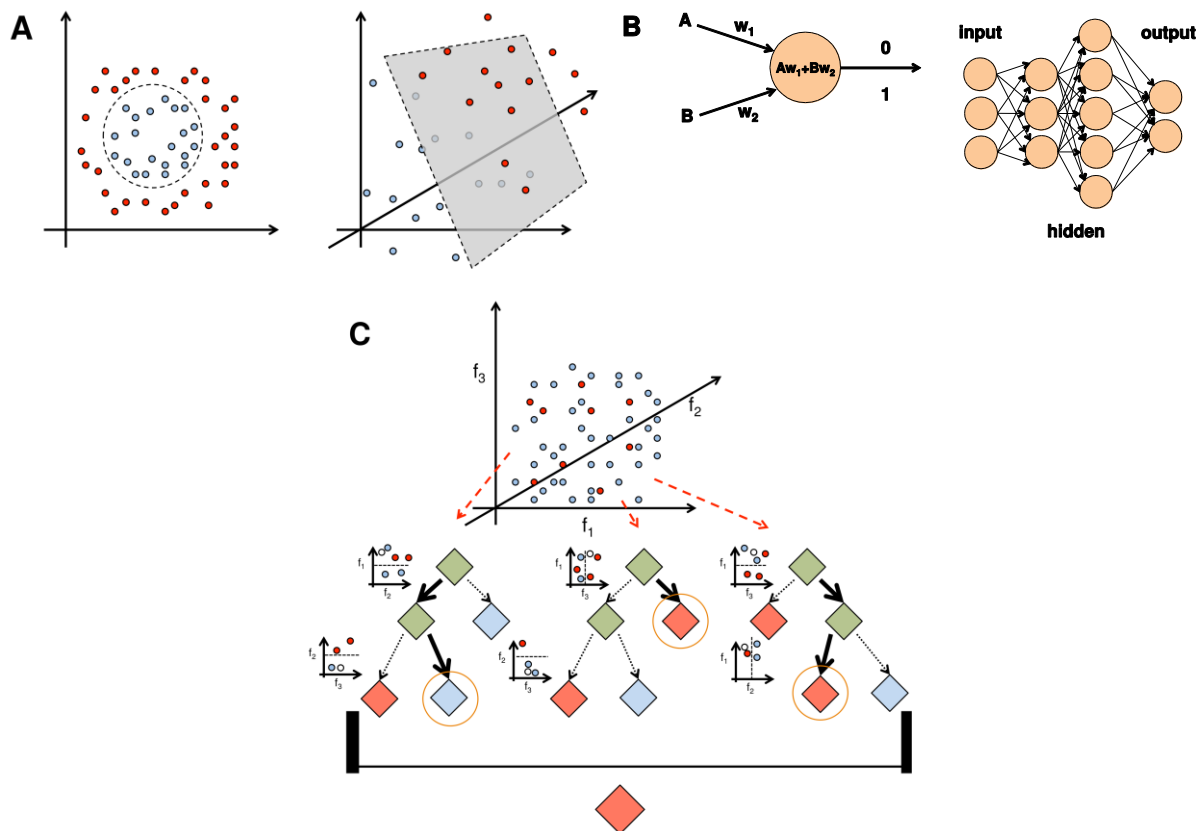


Figure 2: Common supervised learning algorithms. (A) Support vector machines (SVM) separate input data into classes using decision boundaries which are high-dimensional analogs of flat lines and planes in a higher-dimensional mathematical representation of the input data (right). The actual input data (left) has only two dimensions, so, each axes corresponds to values of “features” (see main text) of the individual data points, one for horizontal and one for vertical. In its own natural two dimensions (left), the input data is inseparable by a line, but can be separated by a high-dimensional plane after transformation into the higher dimensional space (right) by SVM. The dashed decision boundary (left panel) is the class boundary which results from projecting the high-dimensional plane down onto the input space. (B) A graphical illustration of the computational algorithms underlying artificial neural networks. A simple artificial neuron (left) with two input feature values A and B. Each feature value is multiplied by a weight (w_1 , w_2) and the result is added together, and forms the input to a mathematical function which is used to determine the predicted class output of the neuron (0 or 1). A training algorithm applied to the training data determines the parameters of the algorithm which are the weight values. Simple artificial neurons such as these are connected together, input to output, into networks to make powerful classification algorithms, for example, a “two-layer feed-

forward” network taking three feature inputs and producing two outputs (right). (C) Depicting decision trees and random forests (RF) for three-dimensional input feature data (top graph axes with f_1 horizontal, f_2 depth and f_3 vertical, red/blue indicates training data class). A decision tree represents hierarchical splitting of the input data into rectangular regions which are “nested” inside each other, successively “zooming in” to a small region of the input space which contains only a few data points which are expected to belong to only one class. Each split occurs at specific values of the input features, giving rise to the decision boundaries (horizontal and vertical dashed black lines on small inset feature axes). To make a decision tree prediction, the feature values of an input data point determine which route down the tree to follow; obtaining a predicted class (red/blue diamonds) at the bottom of the tree. Random forests create many “randomized” decision trees by training each tree on a randomly chosen subset of both the training data and the feature axes. For a new input (white circles, inset feature axes), each decision tree is followed (thick black arrows) to obtain multiple class predictions, one from each tree (circled in orange). By majority vote over these predictions a final random forest class prediction is obtained (in this case red).

Table 1: Uses of machine learning algorithms for PD wearable sensor data.

Study	Year	Reference	Sample Size	Tasks	Sensor(s)	Machine Learning Algorithm	Type of Algorithm	Prediction	Reported Results
Hoff et al.	2001	[33]	23 PD	specific activities of daily living	accelerometers	linear regression	supervised	dyskinesia	correlation coefficient of 0.83, 0.87, 0.82 while sitting, counting, and spelling, respectively
Keijsers et al.	2003	[34]	13 PD	specific activities of daily living	accelerometers	neural network classifier	supervised	dyskinesia	classification accuracy of 93.7%, 99.7%, and 97.0% in the arm, trunk, and leg, respectively
Little et al.	2009	[35]	23 PD, 8 controls	voice	voice recording	support vector machine	supervised	Parkinson's disease	overall classification accuracy of 91.4%
Patel et al.	2009	[11]	12 PD	specific motor tasks	accelerometers	support vector machine	supervised	tremor, bradykinesia, dyskinesia	percent estimation error of 2.8% for tremor, 1.7% for bradykinesia, and 1.2% for dyskinesia
Cancela et al.	2010	[36]	20 PD	free movement	accelerometers, gyroscopes	k-nearest neighbor, decision tree, neural network, support vector machine	unsupervised and supervised	bradykinesia	classification accuracy of 70-86%, depending on the algorithm
Tsipouras et al.	2010	[37]	7 PD, 3 controls	specific activities of daily living, posture	accelerometers, gyroscopes	naïve Bayes, 5-nearest neighbor, decision trees, random forest	supervised	dyskinesia	classification accuracy of 93.73% for levodopa-induced dyskinesia detection
Zwartjes et al.	2010	[38]	6 PD, 7 controls	specific activities of daily living	accelerometers, gyroscopes	decision tree	supervised	tremor, bradykinesia	classification accuracy of 99.3% for activity classifier; correlation of 0.84 with UPDRS tremor score
Memedi et al.	2011	[39]	65 PD	fine motor tests	smartphone touchscreen	linear regression	supervised	total UPDRS	correlation coefficient of 0.6 with total UPDRS
Roy et al.	2011	[40]	19 PD, 4 controls	free movement	accelerometers, surface EMG, video	neural network	supervised	tremor, dyskinesia	sensitivity of 88.6-93.8%, specificity of 91.9-94.6%, global error rate of 6.3-8.4% for tremor; sensitivity of 90%, specificity of 91.3-95.5%, global error rate of 4.5-9.4% for dyskinesia
Mazilu et al.	2012	[28]	10 PD	walking	accelerometers, gyroscopes	random trees, random forest, decision trees, naïve Bayes classifier, neural network classifier, k-nearest neighbor	supervised	freezing of gait	sensitivity of 99.69% and specificity of 99.96% for detecting freezing of gait events
Rigas et al.	2012	[41]	18 PD, 5 controls	posture stability	accelerometers	hidden Markov model classifier	supervised	tremor classification	classification accuracy of 87% for tremor severity
Albert et al.	2012	[42]	8 PD, 18 controls	sitting, standing, holding phone, walking	accelerometers	support vector machine, sparse multinomial logistic regression	supervised	type of activity (e.g. sitting, standing, walking)	cross-validation accuracy of 86% for SVM and 85.2% for SMLR
Tripoliti et al.	2013	[29]	11 PD, 5 controls	specific activities of daily living	accelerometers, gyroscopes	naïve Bayes classifier, random forest, decision trees, random trees	supervised	freezing of gait	sensitivity of 81.94%, specificity of 98.74%, classification accuracy of 96.11% and AUC of 98.6% for freezing of gait events

Arora et al.	2015	[9]	10 PD, 10 controls	voice, posture, gait, tapping	voice recording, smartphone touchscreen	random forest	supervised	UPDRS motor activities	sensitivity of 96.2% and specificity of 96.9% in PD diagnosis; average mean absolute error in modified motor UPDRS score of 1.26 points
Wahid et al.	2015	[43]	23 PD, 26 controls	self-selected walking	cameras	kernel Fisher discriminant, naïve Bayes classifier, k-nearest neighbor, support vector machine, random forest	supervised	gait disturbances, Parkinson's disease	classification accuracy of 92.6% using random forests and multiple regression normalization
Cook et al.	2015	[44]	50 PD, 68 controls	specific activities of daily living; Timed Up and Go test	infrared motion detectors, magnetic door sensors, ambient light sensors, vibration sensors	decision tree classifier, naïve Bayes classifier, random forest, support vector machine, adaptive boosting, principal component analysis	unsupervised and supervised	Parkinson's disease with or without mild cognitive impairment	classification accuracy of 97% and AUC of 0.97 in classifying healthy aged controls from PD without MCI using adaptive boosting with decision tree and random resampling
Jane et al.	2016	[12]	93 PD, 73 controls (from database)	walking	force transducer in shoe	Q-back propagated time-delay neural network	supervised	gait disturbances, Parkinson's disease	classification accuracy of 90.91-92.19% for PD, depending on the input dataset

Table 1 citations:

Hoff³⁰, Keijsers³¹, Little³², Patel¹¹, Cancela³³, Tsipouras³⁴, Zwartjes³⁵, Memedi³⁶, Roy³⁷, Mazilu³⁸, Rigas³⁹, Albert⁴⁰, Tripoliti⁴¹, Arora⁹, Wahid⁴², Cook⁴³, Jane¹²

Table 2: Glossary of main machine learning terminology.

Bayesian	There are two different interpretations of probability in statistics. The <i>frequentist</i> interpretation defines probability as an intrinsic property of observations, for example, the observed prevalence probability of any one individual having Parkinson's being estimated at around 150/100,000 from historical diagnosis data. The <i>Bayesian</i> interpretation defines probability as a <i>prior degree of belief</i> (about e.g. a diagnosis) that is updated as more data arrives, for example, as the individuals symptoms are observed.
Classification	Assignment of input observation data to a category (e.g. diagnostic class: Parkinson's disease vs. healthy).
Clustering	Grouping observations in a dataset together under the same category/class where they are similar to each other according to their input data values.
Cross-validation	A method to estimate the prediction accuracy/performance of a machine learning model which would be expected to generalize beyond any particular dataset. It is a procedure whereby (1) the dataset is partitioned into a subset used to train the model and a subset to test model performance (hold-out testing), and (2) the procedure is repeated with different partitions of the dataset.
Dimension	The number of features in a dataset. For example, a dataset which includes height, weight, and blood pressure would have a dimension of three; a dataset with measurements from multiple sensors in a wearable device will usually have far higher dimension.
Dimensionality reduction	Decreasing the number of total features needed to capture all the useful information in the observations, e.g. by removing features that are highly correlated with other features (feature selection) or by deriving a smaller set of new features from combinations of the existing features (for example by using principal or independent components analysis).
Feature	Summarized input data that can be used to predict a corresponding output. For example, a summary of the patient's accelerometer data from a wearable device or a clinical observation are both features that may be used to predict Parkinson's disease symptoms or diagnostic status.
Hold-out testing	A method to measure the accuracy/performance of a machine learning model in which a subset of the dataset is used to train the model and the remainder is used to test the accuracy of the model.
Input/output	A machine learning method is a mathematical algorithm or model that acts on <i>input</i> data to produce <i>output</i> data. For example, in classification, the input to the machine learning model is a high-dimensional data item or a smaller set of features, and the output is a predicted class label for a particular diagnosis.
Label	The "ground-truth" outputs (e.g. diagnostic classes/categories in disease classification or symptom severities in symptom severity regression) for a training subset.
Learning/training	The process of optimizing a machine learning algorithm, using training input data, to make accurate predictions.
Mathematical model/ algorithm/ method	A model is a set of mathematical equations which describes a (statistical/non-statistical) relationship between variables, and implemented in software as a series of programming steps (algorithm). In the context of this article these models represent machine learning methods.
Model complexity	Usually characterized by either the number of input features required to make predictions, the number of assumptions, or the number of "degrees of freedom" implicit in a mathematical model. Overly complex models overfit the data by producing outputs that accurately predict the training data outputs but not the test data; overly simple models may not be sensitive enough to the input data to

	make accurate predictions.
Parametric/non-parametric	Types of machine learning models based upon assumptions about the input space. Linear regression is a parametric model, because it is specified by a finite set of parameters (the regression coefficients). Parametric models are typically straightforward to calculate and easily interpreted, but require assumptions about the input data which may be unrealistic. A non-parametric model (e.g. a kernel support vector machine) does not make simple assumptions such as linearity and so can discover arbitrary input-output relationships. While it can produce accurate predictions for complex input-output relationships, it is more difficult to calculate and often yields non-interpretable models.
Prediction	Typically, a prediction is an estimate of an output data quantity (such as a diagnostic category or a symptom severity measure) produced by a trained machine learning algorithm, using input data (such as sensor readings).
Probabilistic graphical model	A general mathematical framework for statistical modeling of complex phenomena by representing the relationships between all the variables (e.g. input features and outputs) in the problem as a graph of statistical dependencies between these variables.
Regression	Predicting the value of an output variable by the value of associated input variables and a known relationship between them.
Regularization	A method to control against overfitting by penalizing model complexity (for example additional input features in a regression or classification model), therefore allowing for a proper trade-off between training accuracy/performance and generalizability to other data.
Replication testing	Similar to hold-out testing but where the test subset is obtained from an entirely different experimental setting. This is analogous to prospective testing.
Supervised/unsupervised/semi-supervised learning	Machine learning algorithms that create a mathematical model to produce output data, trained on either labelled data (<i>supervised</i>), unlabelled data (<i>unsupervised</i>), or a mix of labelled and unlabelled data (<i>semi-supervised</i>).
Test/train subset	Machine learning algorithms create a predictive mathematical model by identifying patterns from observations in a <i>training subset</i> of a dataset. However, evaluation of the accuracy/performance of such a model is biased towards unrealistic accuracy if computed using the same training subset, and typically an independent <i>test set</i> is used to evaluate the performance of the model.

References

1. Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed. New York, NY: Springer, 2009.
2. eMarketer. Smartphone User Penetration as Percentage of Total Global Population from 2011 to 2018*: Statista - The Statistics Portal; 2014.
3. Bishop C. Pattern Recognition and Machine Learning. New York, NY: Springer, 2006.
4. Goetz CG, Fahn S, Martinez-Martin P, et al. Movement Disorder Society-sponsored revision of the Unified Parkinson's Disease Rating Scale (MDS-UPDRS): Process, format, and clinimetric testing plan. *Mov Disord* 2007;22(1):41-47.
5. Abiola S, Biglan K, Dorsey E, et al. Smartphone-PD: Preliminary results of an mHealth application to track and quantify characteristics of Parkinson's disease in real-time (Abstracts of the Nineteenth International Congress of Parkinson's Disease and Movement Disorders). *Mov Disord* 2015;30:S568-S633.
6. Zhan A, Little MA, Harris DA, et al. High Frequency Remote Monitoring of Parkinson's Disease via Smartphone: Platform Overview and Medication Response Detection. *arXiv preprint* 2016:1601.00960.

7. Neto EC, Bot BM, Perumal T, et al. Personalized hypothesis tests for detecting medication response in parkinson disease patients using iPhone sensor data. *Pac Symp Biocomput* 2016;21:273-284.
8. Proakis JG, Manolakis DK. *Digital Signal Processing: Principles, Algorithms, and Applications*. 4th ed. Upper Saddle River, NJ: Prentice-Hall, 2006.
9. Arora S, Venkataraman V, Zhan A, et al. Detecting and monitoring the symptoms of Parkinson's disease using smartphones: A pilot study. *Parkinsonism Relat Disord* 2015;21(6):650-653.
10. Stamatakis J, Ambroise J, Crémers J, et al. Finger Tapping Clinimetric Score Prediction in Parkinson's Disease Using Low-Cost Accelerometers. *Comput Intell Neurosci* 2013;2013:13.
11. Patel S, Lorincz K, Hughes R, et al. Monitoring Motor Fluctuations in Patients With Parkinson's Disease Using Wearable Sensors. *IEEE Trans Inf Technol Biomed* 2009;13(6):864-873.
12. Nancy Jane Y, Khanna Nehemiah H, Arputharaj K. A Q-back propagated time delay neural network for diagnosing severity of gait disturbances in Parkinson's disease. *J Biomed Inform* 2016;(in press).
13. Goodfellow I, Bengio Y, Courville A. *Deep Learning*: MIT Press (in preparation), 2016.
14. Nguyen A, Yosinski J, Clune J. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2015:427-436.
15. Lawton M, Baig F, Rolinski M, et al. Parkinson's Disease Subtypes in the Oxford Parkinson Disease Centre (OPDC) Discovery Cohort. *J Parkinsons Dis* 2015;5(2):269-279.
16. van der Maaten LJP, Postma EO, van den Herik HJ. Dimensionality reduction: a comparative review. *J Mach Learn Res* 2009;10:66-71.
17. Brakedal B, Tysnes O-B, Skeie GO, Larsen JP, Müller B. The factor structure of the UPDRS motor scores changes during early Parkinson's disease. *Parkinsonism Relat Disord* 2014;20(6):617-621.
18. Vassar SD, Bordelon YM, Hays RD, et al. Confirmatory Factor Analysis of the Motor Unified Parkinson's Disease Rating Scale. *Parkinsons Dis* 2012;2012:10.
19. Belkin M, Niyogi P, Sindhvani V. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. *J Mach Learn Res* 2006;7:2399-2434.
20. Ghahramani Z. Probabilistic machine learning and artificial intelligence. *Nature* 2015;521(7553):452-459.
21. White N, Johnson H, Silburn P, Mengersen K. Dirichlet process mixture models for unsupervised clustering of symptoms in Parkinson's disease. *J Appl Stat* 2012;39(11):2363-2377.
22. Arlot S, Celisse A. A survey of cross-validation procedures for model selection. *Stat Surv* 2010:40-79.
23. Kaufman S, Rosset S, Perlich C. Leakage in data mining: formulation, detection, and avoidance. *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Diego, California, USA: ACM; 2011. p. 556-563.
24. Cortes C, Mohri M, Riley M, Rostamizadeh A. Sample Selection Bias Correction Theory. In: Freund Y, Györfi L, Turán G, Zeugmann T, eds. *Algorithmic Learning Theory: 19th International Conference, ALT 2008, Budapest, Hungary*. Berlin, Heidelberg: Springer, 2008:38-53.
25. Hand DJ. Classifier Technology and the Illusion of Progress. *Statist Sci* 2006:1-14.
26. Breiman L. Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author). *Statist Sci* 2001:199-231.
27. Sculley D, Holt G, Golovin D, et al. Hidden Technical Debt in Machine Learning Systems. *Adv Neural Inf Process Syst*; 2015. p. 2494-2502.
28. Michael J. Fox Foundation for Parkinson's Research. Fox Insight Clinical Trials NCT02668835 and NCT02474329. <https://foxinsight.michaeljfox.org/>. 2016.
29. Trochim W. *Research Methods Knowledge Base*. 2nd ed. Mason, OH: Atomic Dog Publishing, 2001.
30. Hoff JI, van den Plas AA, Wagemans EAH, van Hilten JJ. Accelerometric assessment of levodopa-induced dyskinesias in Parkinson's disease. *Mov Disord* 2001;16(1):58-61.

31. Keijsers NLW, Horstink MWIM, Gielen SCAM. Automatic assessment of levodopa-induced dyskinesias in daily life by neural networks. *Mov Disord* 2003;18(1):70-80.
32. Little MA, McSharry PE, Hunter EJ, Spielman J, Ramig LO. Suitability of Dysphonia Measurements for Telemonitoring of Parkinson's Disease. *Biomedical Engineering, IEEE Transactions on* 2009;56(4):1015-1022.
33. Cancela J, Pansera M, Arredondo MT, et al. A comprehensive motor symptom monitoring and management system: The bradykinesia case. *Engineering in Medicine and Biology Society (EMBC); 2010*. p. 1008-1011.
34. Tsipouras MG, Tzallas AT, Rigas G, Bougia P, Fotiadis DI, Konitsiotis S. Automated Levodopa-induced dyskinesia assessment. *Engineering in Medicine and Biology Society (EMBC); 2010*. p. 2411-2414.
35. Zwartjes DG, Heida T, Van Vugt JP, Geelen JA, Veltink PH. Ambulatory monitoring of activities and motor symptoms in Parkinson's disease. *IEEE Trans Biomed Eng* 2010;57(11):2778-2786.
36. Memedi M, Westin J, Nyholm D, Dougherty M, Groth T. A web application for follow-up of results from a mobile device test battery for Parkinson's disease patients. *Comput Methods Programs Biomed* 2011;104(2):219-226.
37. Roy SH, Cole BT, Gilmore LD, De Luca CJ, Nawab SH. Resolving signal complexities for ambulatory monitoring of motor function in Parkinson's disease. *Engineering in Medicine and Biology Society (EMBC); 2011: IEEE*. p. 4832-4835.
38. Mazilu S, Hardegger M, Zhu Z, et al. Online detection of freezing of gait with smartphones and machine learning techniques. *6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth); 2012*. p. 123-130.
39. Rigas G, Tzallas AT, Tsipouras MG, et al. Assessment of Tremor Activity in the Parkinson's Disease Using a Set of Wearable Sensors. *IEEE Trans Inf Technol Biomed* 2012;16(3):478-487.
40. Albert MV, Toledo S, Shapiro M, Koerding K. Using mobile phones for activity recognition in Parkinson's patients. *Front Neurol* 2012;3:158.
41. Tripoliti EE, Tzallas AT, Tsipouras MG, et al. Automatic detection of freezing of gait events in patients with Parkinson's disease. *Comput Methods Programs Biomed* 2013;110(1):12-26.
42. Wahid F, Begg RK, Hass CJ, Halgamuge S, Ackland DC. Classification of Parkinson's Disease Gait Using Spatial-Temporal Gait Features. *IEEE J Biomed Health Inform* 2015;19(6):1794-1802.
43. Cook DJ, Schmitter-Edgecombe M, Dawadi P. Analyzing Activity Behavior and Movement in a Naturalistic Environment Using Smart Home Techniques. *IEEE J Biomed Health Inform* 2015;19(6):1882-1892.