PREDICTING DISABILITY PROGRESSION IN SECONDARY PROGRESSIVE MULTIPLE SCLEROSIS BY MACHINE LEARNING: A COMPARISON OF COMMON METHODS AND ANALYSIS OF DATA LIMITATIONS

by

Marco Law

B.Eng., Carleton University, 2016

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Biomedical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

July 2019

© Marco Law, 2019

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, a dissertation/thesis entitled:

Predicting Disability Progression In Secondary Progressive Multiple Sclerosis By Machine Learning: A Comparison Of Common Methods And Analysis Of Data Limitations

submitted by Marco Law in partial fulfillment of the requirements for

the degree of Master of Applied Science

in Biomedical Engineering

Examining Committee:

Dr. Tam, Roger
Supervisor

Dr. Traboulsee, Anthony
Supervisory Committee Member

Dr. Wang, Jane Z.
Supervisory Committee Member

Additional Examiner

Additional Supervisory Committee Members:

Supervisory Committee Member

Supervisory Committee Member

Abstract

Secondary progressive MS (SPMS) is a late stage neurological disease characterized by chronic worsening. Enhanced prediction of SPMS progression could improve clinical trial design and may inform patient/physician treatment decisions, but the task is difficult since MS is characterized by heterogeneity in terms of clinical features, genetics, pathogenesis, and treatment response. The Expanded Disability Status Scale (EDSS), is a nominal MS disability scale for describing physical disability that is often incorrectly treated as a continuous variable. Machine learning (ML) models identify relationships between features and outcome, while deep learning (DL) adds on automatic feature extraction from low-level data. Although both have been applied to MS classification and early-stage transition prediction, late-stage MS disability progression prediction is lacking. The contributions of this thesis are the design, implementation, and evaluation of 1) ML using user-defined features (UDF), 2) DL using automatically extracted brain lesion mask features (BLM) for predicting SPMS disability progression, and 3) an evaluation of the impact on performance when EDSS is misused as a continuous variable. SPMS participants (n=485) in a 2-year placebo-controlled (negative) trial of MBP8298 were labelled progressors if a 6-month-sustained increase in EDSS ($\geq$1.0 and $\geq$0.5 for a baseline of $\leq$5.5 and $\geq$6.0 respectively) was observed within 24 months. UDF included EDSS, Multiple Sclerosis Functional Composite component scores, T_2 lesion volume, brain parenchymal fraction, disease duration, age, and sex. Logistic regression (LR), ensemble support vector machines (enSVM), random forest (RF), and AdaBoost decision trees (AdBDT) were trained using UDF only. DL networks were trained to extract BLM features and predict progression with and without UDF. The

primary outcome was the area under the receiver operating characteristic curve (AUC). Of the 485 participants, 115 progressed. When using continuous EDSS, AdBDT and RF had a greater AUC (60.3% and 56.2%) than enSVM (52.1%) and LR (44.7%), and DL using only BLM features outperformed LR using UDF (55.0% vs. 45.0%). UDF did not improve DL. RF and AdBDT were robust to EDSS treatment. SPMS trial cohorts selected by ML, DL, or both, could identify those at highest risk for progression, enabling smaller, shorter studies.

Lay Summary

Secondary progressive MS (SPMS) is a late stage neurological disease characterized by chronic worsening. Unfortunately, accurate prognoses are difficult to obtain as past clinical scores and traditional magnetic resonance imaging (MRI) measurements have poor predictive value of future disability and individual disease courses vary greatly. Artificial intelligence (AI) has the ability to learn complex patterns from seemingly random data. This thesis presents two AI approaches, machine learning and deep learning, for predicting disability progression in secondary progressive MS, a late disease stage characterized by chronic worsening which results in lasting disability. The prediction task was approached by training several machine learning models to discover relationships between progression, clinical disease scores and imaging biomarkers, as well as a deep learning model to automatically extract predictive features from brain lesion masks. Additionally, this thesis presents the impact on machine and deep learning models that incorrectly processing one clinical disease scale can cause.

Preface

The research described in this thesis was performed under the supervision of Dr. Roger Tam. Development, implementation, and evaluation, unless otherwise stated, was performed by the author of this thesis, M. Law. All figures and images, unless stated to be sourced or adapted, was generated by the author.

This thesis is a post hoc data analysis of existing research data initially collected for a negative 2-year, randomized, double-blind, placebo-controlled phase III study that evaluated the efficacy and safety of MBP8298 in patients diagnosed with secondary progressive multiple sclerosis (SPMS). The author's lab, the MS/MRI Research Group, was responsible for the processing and analysis of MRI images. This results in this thesis have not been previously reported.

Chapter 3 was based on the accepted abstract titled "Machine learning outperforms linear regression for predicting disability progression in SPMS" which was presented as a poster at the European Commission for Treatment and Research in Multiple Sclerosis (ECTRIMS) 2018 Congress in Berlin. An abstract titled "Prediction of disability progression in SPMS is outperformed by EDSS analyzed as a categorical variable rather than a continuous variable" has been accepted for poster presentation at ECTRIMS 2019 in Stockholm. In addition to writing the abstract, the author was involved in the development, implementation, and evaluation of the methods described. The remaining co-authors, A. Traboulsee, D.K.B. Li, R. Carruthers, M.S. Freedman, S. Kolind, and R. Tam contributed to the design and interpretation of the results, as well as the editing of the abstract. A manuscript consisting of the work described in Chapter 3 is currently under revision for a journal.

Chapter 4 utilized brain lesion masks generated for the SPMS MBP8298 study mentioned above using the methodologies described in Section 2.1 by the original researchers. The author, M. Law, performed all image pre-processing described in Section 4.2. Design of the deep learning networks was inspired by Dr. Youngjin Yoo et al and their related publication titled “Deep learning of brain lesion patterns and user-defined clinical and MRI features for predicting conversion to multiple sclerosis from clinically-isolated syndrome.” The author of the thesis performed the development, implementation, and evaluation of the methodology described in Section 4.4. A manuscript describing parts of Chapter 4 has been prepared for submission to a journal.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ΔNPV	change in pre- to post-negative predictive value
ΔPPV	change in pre- to post-positive predictive value
10CV	10-fold cross validation
9HPT	9-hole peg test
AI	artificial intelligence
AUC	area under the curve
BOD	burden of disease
BPF	brain parenchymal fraction
CDMS	clinically definite multiple sclerosis
CDP	confirmed disability progression
CIS	clinically isolated syndrome
CNN	convolutional neural network
CNS	central nervous system
CSF	cerebral spinal fluid
DLN	deep learning network
DNN	dense neural network
DT	decision tree
EDSS	Expanded Disability Status Scale
EDT	Euclidean distance transform
GM	grey matter

LR	logistic regression
MR	magnetic resonance
MRI	magnetic resonance imaging
MS	multiple sclerosis
MSFC	Multiple Sclerosis Functional Composite
NPV	negative predictive value
PASAT	paced auditory serial addition test
PDw	proton density weighted
PPMS	primary progressive multiple sclerosis
PPV	positive predictive value
RF	radio frequency
RF	random forest
RRMS	relapsing-remitting multiple sclerosis
SGD	stochastic gradient descent
SPMS	secondary progressive multiple sclerosis
SVM	support vector machine
T1w	T_1-weighted
T25W	timed 25-foot walk
T2w	T_2-weighted
WM	white matter

Acknowledgements

Although only one author is named in this thesis, it would not have been possible without the unwavering support of all those who were directly or indirectly involved.

I want to thank my supervisor Dr. Roger Tam, and Dr. Anthony Traboulsee, for their guidance throughout my academic journey. I appreciate the trust they had in me, allowing me to pursue all of the avenues I wanted to explore with respect to my research and providing me with the resources and help I needed without hesitation. In my times of financial need, he connected me with opportunities for additional work.

To Ken Bigelow, thank you for going above and beyond by accommodating my running experiments while going about all of your other duties and responsibilities.

To Dr. Youngjin Yoo, Dr. Lisa Tang and Kevin Lam, their help and support, particularly at the beginning of my journey, enabled me to efficiently ramp up research productivity and decipher work by my predecessors.

I would like to thank my mom for letting me do what I want, as well as my friends and acquaintances who kept me sane throughout my journey. I'd like to also thank Terri Yip, Julianna Mar and Rachel Jin. They all took the time to read through this thesis to provide me with feedback – out of pure curiosity.

Finally, huge thanks go to the National Science and Engineering Research Council, The Faculty of Graduate and Postdoctoral Studies, and The Multiple Sclerosis Society of Canada for their financial support. Without them, I would have had to eat only instant ramen for sustenance and swim in even more debt than I have been. This thanks also extends to Dr. Shannon Kolind who provided me with additional research assistantships.

Dedication

I dedicate this thesis to the individuals who suffer from MS, in particular those in the later disease stages who are unable to take advantage of the disease control and management strategies that have only recently become available to the newly diagnosed.

Chapter 1:

Introduction

1.1 Multiple Sclerosis

Multiple sclerosis (MS) is a chronic autoimmune demyelinating disease of the central nervous system (CNS), characterized by the destruction of the myelin sheath that surrounds and insulates axons of nerve cells. Myelinated axons allow for saltatory conduction of a nerve impulse (the jumping of nerve impulses between gaps between consecutive myelin sheaths known as nodes of Ranvier), thereby negating the otherwise required sequential depolarization of the entire cell membrane (a much slower process). The demyelination of axons in CNS results in scarring, disruption of nerve impulses, nerve fiber damage, and ultimately axonal death, resulting in clinical presentations and disease progressions that may vary greatly between individuals. Some symptoms of MS include extreme fatigue, lack of coordination, weakness, tingling, impaired sensation, vision and bladder problems, cognitive impairment, and mood changes.

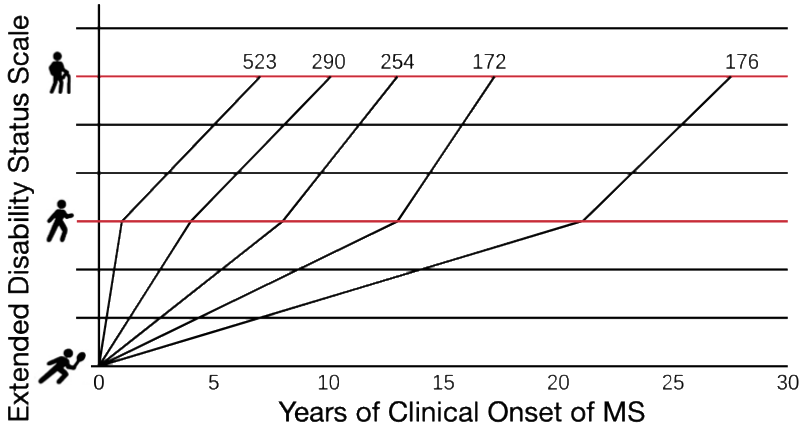

Figure 1-1 Heterogeneity of Multiple Sclerosis. An illustration of the high degree of variability in disease progression. Adapted from [1].

MS is a chronic disease and most patients experience varying rates and severity of eventual permanent disability (Figure 1-1). There are four clinical forms of MS outlined by the McDonald diagnostic criteria: clinically-isolated syndrome (CIS), primary progressive (PPMS), relapsing-remitting (RRMS), and secondary progressive (SPMS) [2]. While some MS patients experience uninterrupted disability progression from disease onset (PPMS), the majority of MS patients start with the relapsing-remitting phase (characterized by acute worsening from which patients may or may not fully recover and periods of remission) before advancing into the secondary progressive phase (SPMS) [3].

1.1.1 Secondary Progressive Multiple Sclerosis

Unlike PPMS where disability gradually worsens from disease onset, secondary progressive multiple sclerosis is a retrospective diagnosis based on a history of gradual worsening without acute disease worsening that follows a relapsing-remitting disease course, [2]. Figure 1-2 illustrates hypothetical PPMS, RRMS and SPMS disease courses for visualization of the different disability progressions.

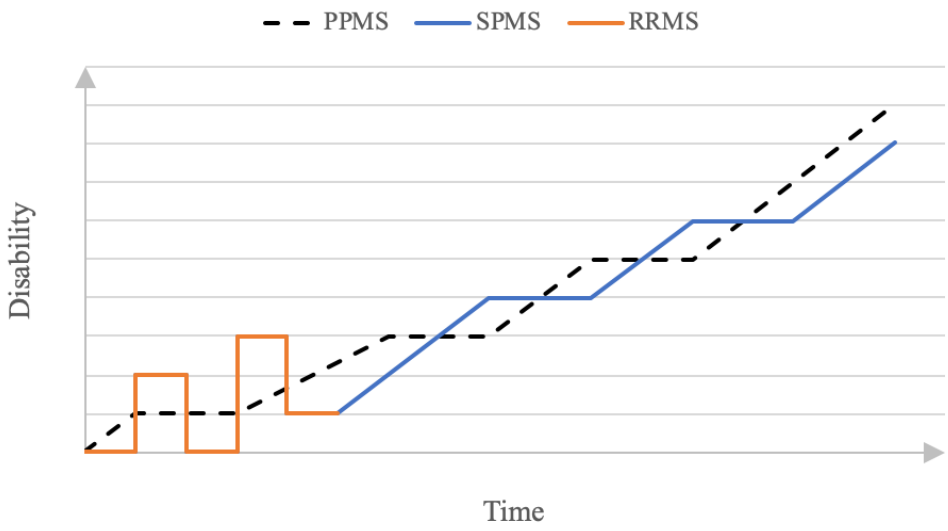

Figure 1-2 Example of primary progressive (PP) MS, relapsing-remitting (RR) MS and secondary progressive (SP) MS. PPMS (dashed) is characterized by chronic disease worsening from onset. RRMS (orange) is characterized by acute disability that may leave permanent deficits, and SPMS is characterized by chronic disease worsening following a history of RRMS

1.1.2 Clinical Scores for Multiple Sclerosis

The Expanded Disability Status Scale (EDSS) is the most commonly used clinical score for summarizing disability in MS. Although EDSS was designed as an ordinal (ordered categorical) variable, it is often treated as a continuous variable. EDSS ranges from 0 to 10 in 0.5 increments, signifying increasing disability from absence of neurological deficits to death caused by MS. An individual's EDSS score is a combination of the scores of eight functional systems – pyramidal, cerebellar, brain stem, sensory, bowel and bladder, visual, cerebral, and other [4]. While EDSS provides a simple overview of a patient's disability, it focuses heavily on physical disability and less on the highly variable and nuanced cognitive impacts of MS.

Unlike EDSS, the Multiple Sclerosis Functional Composite (MSFC) was developed as a composite measure for summarizing arm/hand, leg, and cognitive function assessed

through three neurological function tests – a timed 25-foot walk (T25W) for assessing leg function, a 9-hole peg test (9HPT) for evaluating arm function, and a paced auditory serial addition test (PASAT) for assessing cognitive function. To obtain an individual's MSFC score, the Z-score of the three tests (commonly obtained by standardizing results to the Task Force Dataset) are averaged [5].

1.2 Magnetic Resonance Imaging

Magnetic resonance (MR) imaging (MRI) is a non-radiating and non-invasive imaging method commonly used for visualizing human anatomy. Two-dimensional or three-dimensional images of the body are obtained by first aligning randomly oriented protons of hydrogen atoms in water molecules to an external magnetic field. The protons are then stimulated by radiofrequency (RF) pulses. As the atoms realign with the external magnetic field, RF signals are generated. These are detected by antennas and reconstructed into an image. Tissue is characterized by two relaxation time constants, T_1 and T_2. T_1 relaxation time constant determines the rate at which excited protons realign with the external magnetic field, while T_2 relaxation time constant determines the rate of RF signal decay following excitation. By altering the two parameters of the excitation RF pulse, repetition time and echo time, three image types with unique tissue contrast characteristics – T_1-weighted (T1w), T_2-weighted (T2w), and proton density weighted (PDw) – can be produced. Additional sequences (e.g. fluid-attenuated inversion recovery, diffusion weighted, flow sensitive, etc.) can also be produced by introducing new parameters which further manipulate the RF pulses. To detect specific pathologies, contrast agents may also be used. Figure 1-3 shows sample T1w, T2w, and PDw brain MR images.

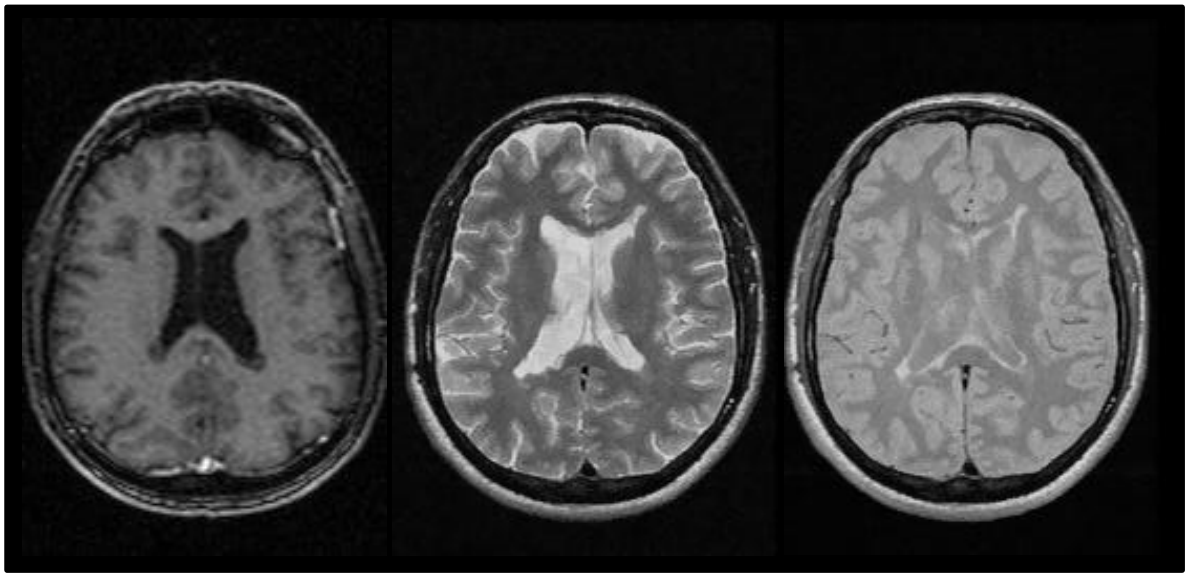

Figure 1-3 Examples of brain MR images. Left: T_1-weighted (T1w) with contrast MRI. Middle: T_2-weighted (T2w) MRI. Right: proton density weighted (PDw) MRI.

1.2.1 Magnetic Resonance Imaging in Multiple Sclerosis

MR images of the brain and spinal cord are used most commonly for the identification of brain and spinal cord lesions. Depending on the clinical presentation of MS, MRI evidence demonstrating one or both of lesion dissemination in space and in time, may be required for a diagnosis of clinically definite MS (CDMS). Dissemination in space refers to the spatial distribution of lesions within the CNS, while dissemination in time refers to evidence of active lesions across time [6].

MR imaging is also extremely valuable for the monitoring of MS disease progression. Individual or a combination of MR images may be used to extract imaging biomarkers including but not limited to white matter lesion counts, lesion volume, brain atrophy, and gadolinium-enhancing lesions indicating new disease activity [7].

1.3 Artificial Intelligence

Artificial intelligence (AI) refers to computer systems that are able to perform tasks that normally require human intelligence. A small subset of such tasks includes image recognition, image segmentation, object detection, language processing, and classification.

Machine learning is a branch of AI that uses computational algorithms to learn how to perform a specific task (i.e. classification) from a set of training data, after which it can perform the task with new data. Machine learning can be broken down into unsupervised or supervised learning. In unsupervised learning, algorithms learn hidden patterns in unlabeled training data. This is useful for discovering new relationships within a dataset and is more akin to data mining [8]. With supervised learning, the algorithm learns from a set of labelled training data; each example in the training data has a corresponding target output and the algorithm learns the relationships between features in the dataset and the desired output.

Deep learning is an evolution of artificial intelligence from machine learning wherein the learning algorithm is composed of multiple processing layers that enable the learning of various levels of abstraction that are used as features for classification. This approach breaks free from the limitation of learning from the data in their raw form that exists with conventional machine learning approaches [9].

The key difference between machine learning and deep learning is that with machine learning, the algorithm learns relationships between given features to accomplish a given task, while deep learning performs feature extraction as well.

1.3.1 Supervised Machine Learning for Binary Classification

Several machine learning approaches have seen wide-spread application for the classification task. These approaches are logistic regression, support vector machines, decision tree, and ensemble classifiers.

1.3.1.1 Logistic Regression

Logistic regression (LR) is the conventional statistical model for learning linear relationships between explanatory variables and categorical response variables (such as the presence or absence of disease) in many healthcare and clinical applications.

For a continuous response variable with one explanatory variable $x \coloneqq \{x_1\}$, the expected response variable Y given x is denoted by $E(Y|x)$ and has the form shown in Equation (1.1).

$$E(Y|x) = \beta_0 + \beta_1 x_1 \tag{1.1}$$

In the case of a binary variable, the conditional mean outcome is bound between zero and one, such that $0 \leq E(Y|x) \leq 1$, and is achievable with the logistic distribution. The resulting logistic regression model $\pi(x)$ is shown in Equation (1.2). The logit transformation of $\pi(x)$, $g(x)$, enables properties of the linear regression model such as linear parameters and continuous explanatory variables (Equation (1.3)). Given x, the outcome variable y is expressed as $y = \pi(x) + \epsilon$, where ϵ is the error from the conditional mean. In the binary case, as discussed, the probability is $\pi(x)$ when $y = 1$ and $1 - \pi(x)$ when y = 0.

$$\pi(x) = E(Y|x) = \frac{e^{\beta_0 + \beta_1 x_1}}{1 + e^{\beta_0 + \beta_1 x_1}} \tag{1.2}$$

$$g(x) = \ln\left[\frac{\pi(x)}{1 - \pi(x)}\right] = \beta_0 + \beta_1 x_1 \tag{1.3}$$

The parameters (i.e. β_0 and β_1) in logistic regression are estimated by optimization of the log-likelihood function to obtain the maximum likelihood estimates $\hat{\beta}$. The log-likelihood function $L(\beta)$ for n pairs of x and y, $\{(x_i, y_i), i \in 1 \dots n\}$ is:

$$L(\beta) = \sum_{i=1}^{n} \{y_i \ln[\pi(x_i)] + (1 - y_i) \ln[1 - \pi(x_i)]\} \tag{1.4}$$

Equation (1.4) can then be differentiated for β_0 and β_1 and $\hat{\beta}$ obtained by setting the derivatives to zero. The nonlinearity of the resulting derivatives requires iterative numerical methods for solving [10]. The importance or contribution of each explanatory variable to the model output can be directly assessed from the $|\hat{\beta}|$ for each variable, provided that the data has been scaled such that the range of each explanatory variable is similar (e.g. standardizing to a mean of zero and variance of one). Another method of data scaling robust to outliers is the removal of the median and scaling to the interquartile range of each explanatory variable.

1.3.1.2 Support Vector Machines

The support vector machine (SVM) is a machine learning technique for classification problems that aims to learn from input data, an optimal hyperplane with optimal class separation. This is achieved by the identification of support vectors that define the optimal margin – the largest separation between two classes. An example of the two-class separation by an optimal hyperplane is shown in Figure 1-4.

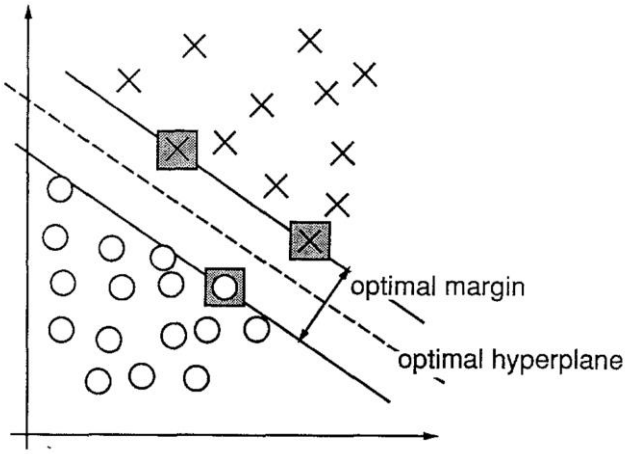

Figure 1-4 An example of an optimal hyperplane and margin in 2-dimensional space, defined by support vectors (gray boxes) that is learned by a support vector machine. Source: [11]

SVMs achieve this by first mapping the n-dimensional input vector x from its input space to a higher, N-dimensional feature space using N-dimensional vector functions ϕ. Classification of an input vector $\boldsymbol{x}$ is then done by applying a decision function (i.e. $sign$ function) on the decision surface function $f(x)$ (Equation (1.5), where $K(\boldsymbol{x}, \boldsymbol{x_i})$ is a kernel function applied to the input vectors $\boldsymbol{x}$ and support vectors $\boldsymbol{x_i}$. Support vectors $\boldsymbol{x_i}$ and weights α_i is found by solving the dual quadratic problem described in [11].

$$f(\boldsymbol{x}) = \sum_{i=1}^{l} y_i \alpha_i K(\boldsymbol{x}, \boldsymbol{x_i}) \tag{1.5}$$

The choice of K determines the type of decision surface that is used to perform classification. Two common choices of kernels are the linear kernel and the radial basis function (RBF) kernel. The linear kernel SVM (linSVM) is similar to that of logistic regression with the improved generalizability due to its fitting to a set of support vectors instead of the complete dataset. The RBF kernel ($K_{RBF} = \exp\left\{-\frac{|x-x_i|^2}{\sigma^2}\right\}$) produces SVMs

with a non-linear decision surface and is particularly useful for learning non-linear relationships but is more at-risk of overfitting.

1.3.1.3 Decision Trees

Decision trees (DT) learn simple decision rules from the input data to perform the classification task. Given a labelled dataset, the DT determines some criterion that splits the data (parent node) into two subsets (child nodes), each with decreased class impurity compared to its parent. This process can be repeated indefinitely until the DT is perfectly fit to the training data by allowing the tree to grow until there are no misclassifications of the training data. To classify new data, the tree simply follows the decision rules determined during training.

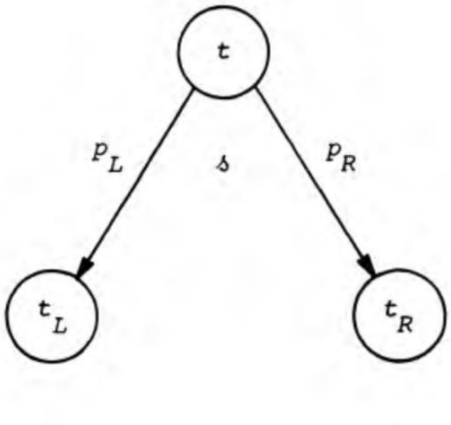

Figure 1-5 Example of node splitting *s* of node *t* into nodes t_L and t_R with proportions p_L and p_R. Source: [12]

Core to the construction of a DT is the calculation of node impurity, denoted by i. A common impurity measure that constructs class probability trees is Gini impurity. The tree is

constructed such that impurity decreases ($\Delta i < 0$) when splitting a parent node t into the child nodes t_L and t_R (Figure 1-5). Impurity change is calculated by Equation (1.6):

$$\Delta i(s,t) = i(t) - p_L i(t_L) - p_R i(t_R) \tag{1.6}$$

where p_A and p_B are the proportions of t that go into nodes t_A and t_B [12]. The generalizability of a DT is mainly governed by model parameters that define maximum tree depth or node splitting requirements; node splitting requirements may be impurity based (i.e. node impurity before splitting and the decrease in impurity resulting from a split) or based on properties of the resulting child nodes (i.e. number of samples in the child node). The decision rules governing each split, s, can be determined by identifying the split with the greatest decrease in impurity from: a) all possible decision rules or b) a random set of decision rules.

The two DT-based ensemble classifiers that are explored in this thesis are the random forest and AdaBoost-DT classifiers.

1.3.1.4 Ensemble Classifiers

Ensemble classifiers are a collection of classifiers whose individual class predictions are used to determine the final class prediction. To construct an ensemble classifier, N-classifiers are first trained individually. The prediction of each classifier is then aggregated to produce one final prediction for the ensemble classifier, commonly by majority-voting or averaging the N individual predictions. As the name implies, majority voting predicts a sample's class based on the class represented by the majority of individual classifiers. Averaging calculates the average of the probabilistic outputs.

Individual classifiers are typically trained on bootstrapped samples – this results in classifiers that are not identical. Unique classifiers can also be trained by introducing

randomness to each classifier (e.g. random forests), changing model parameters (e.g. AdaBoost-DT), or training classifiers on different subsamples of the original dataset. The main benefit of ensemble classifiers is a reduced likelihood of overfitting.

1.3.1.4.1 Random Forest

The random forest classifier (RF) is a collection of DT classifiers, each trained on a random subset of features from the input dataset with/without bootstrapped samples. Complexity, and therefore generalizability, is controlled by the number of DTs in the random forest, the complexity of the individual trees that make up the random forest, and the correlation between the trees [13]. While the original RF uses majority voting, probabilistic predictions can be averaged as well.

1.3.1.4.2 AdaBoost

An AdaBoost classifier is an ensemble of N-classifiers c_i for $i = 1, \dots, N$ whose initial classifier c_0 is trained with uniform sample weights and additional classifiers c_i are trained sequentially using sample weights updated based on the misclassification error of the previous classifier c_{i-1} [14]. The final output of an AdaBoost classifier is a weighted majority-vote of the individual classifiers.

1.3.2 Supervised Convolutional Deep Learning Networks for Classification

Convolutional deep learning networks, referred to as deep learning networks (DLN) herein, consists of a convolutional neural network (CNN) for feature extraction connected to a dense neural network (DNN) for class output, and is commonly used for image recognition and classification tasks.

DLNs are commonly trained using stochastic gradient descent (SGD) and backpropagation [9]. SGD attempts to minimize an objective function by tweaking model

parameters with a fixed step-size in a direction that decreases the objective function. Another increasingly popular variant of SGD is Adam, which uses adaptive step sizes [15].

1.3.2.1 Convolutional Neural Network

The convolutional neural network is structurally similar to the ventral visual cortex pathway [16]. A typical CNN consists of an input (visible) layer and L-convolutional layers. Each convolutional layer $l \coloneqq \{1,2, \dots L\}$ learns abstract representations of the preceding layer's output X^{l-1}. This is achieved by first convoluting the layer input X^{l-1} with the current layer's learnable set of flipped filter kernels ($W^l \mapsto \widetilde{W}^l$) of k filters where $W^l \coloneqq \{W_1^l, W_2^l, \dots, W_K^l\}$, and then applying learnable biases $B^l \coloneqq \{B_1^l B_2^l, \dots, B_K^l\}$. The activations of layer l, $\boldsymbol{X^l}$, is then the element-wise transformation by some non-linear activation function $f(\cdot)$, where $\boldsymbol{X^l} \coloneqq \{X_1^l, X_2^l, \dots, X_K^l\}$. A single feature space (activation of layer l for filter k) is shown in Equation (1.7).

$$X_k^l = f\left(\widetilde{W}_k^l * \boldsymbol{X^{l-1}} + B_k^l\right) \tag{1.7}$$

Convolution introduces translational invariance, and as the weights of the filters are shared by the convolution operation, the number of parameters to be tuned is reduced.

Pooling layers are typically placed between convolutional layers to reduce the spatial dimensionality of individual feature spaces. This is done by subsampling of the feature space through the aggregation of neighboring activations into a single activation with an aggregating function (e.g. max, min, mean). By controlling the size of the neighborhood, varying degrees of invariance to shift and perturbances can be introduced to the feature spaces at the cost of reduced spatial resolution. An example of a convolutional layer followed by a pooling layer is shown in Figure 1-6.

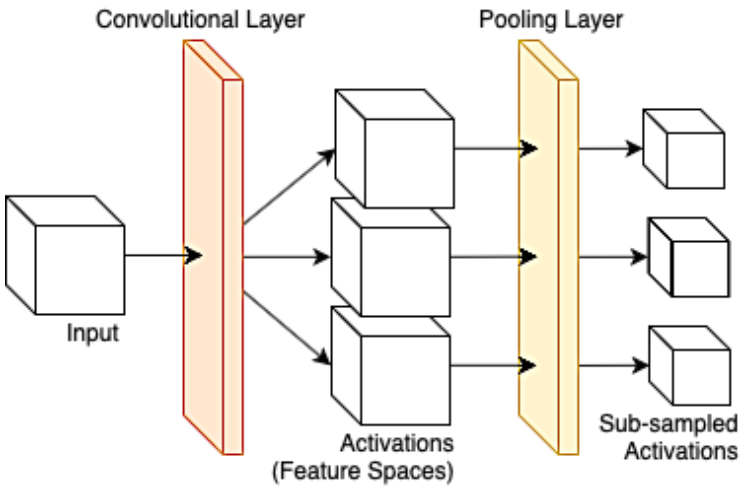

Figure 1-6 Example of a 3D convolutional layer with 3 filters and an arbitrary pooling layer for reducing data dimensionality that would be found in a CNN

In a DLN, CNNs are used to extract features that are used for classification. This is commonly achieved by flattening the last set of feature spaces into a 1-dimensional feature vector that is then used as the input to a dense neural network.

1.3.2.2 Dense Neural Network

A dense neural network (DNN) for classification tasks consists of one or more hidden dense layers sandwiched between an input (visible) layer and an output layer of class predictions. Figure 1-7 illustrates an example of a 2 hidden layer DNN with 3 input features and one output.

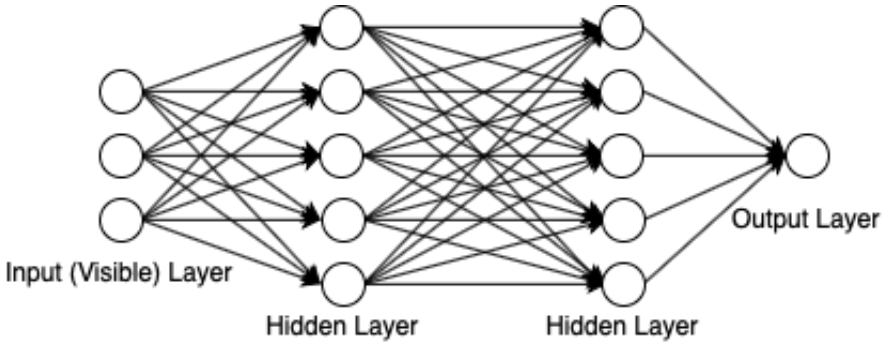

Figure 1-7 Example of a DNN with 2 hidden layers with 5 nodes, an input layer with 3 nodes, and an output layer with a single node

Dense layers are also called fully-connected layers, as all nodes within a layer are connected to all of the nodes both preceding and succeeding it. For the lth layer consisting of N_l nodes and an input vector $\boldsymbol{x}$ (which may be the input layer or the activations of a preceding layer), the activation a_n of node $n \coloneqq \{1,2,\dots,N\}$, is calculated by Equation (1.8), where $\boldsymbol{W}$ and b_n are the learnable set of weights for each element of $\boldsymbol{x}$ and a node specific bias respectively. To allow for the learning of non-linear relationships, a non-linear activation function $f(\cdot)$ is applied to the otherwise linear combination of inputs.

$$a_n = f(\boldsymbol{W_n^T x} + b_n) \tag{1.8}$$

To train a DNN, a labelled dataset D with T samples, $D \coloneqq \{\boldsymbol{x_i}, \boldsymbol{y}_i\}_{i=1}^{T}$, where $\boldsymbol{x}_i$ is the input vector for one sample and y_i is the corresponding true class label, is passed through the DNN to obtain the predicted class $\hat{y}$. Weights and biases are iteratively updated such that the average of a loss function over all samples is minimized. For classification, the cross-entropy loss is optimized by SGD (or a variant, such as Adam) and backpropagation [9].

In DLNs, the connection between the CNN and DNN allows for backpropagation of loss gradients from the output layer of the DNN to the first convolutional layer of the CNN.

This enables the CNN to update weights in its convolutional filters to extract the most useful features for classification by the DNN.

1.4 Literature Review of AI Applications in Multiple Sclerosis

Supervised learning has enabled the development of disease-specific decision support machines for classification and prediction, but the use of machine and deep learning in multiple sclerosis lags behind that of other neurological disorders. One literature survey of publications using AI with neuroimaging in neurological disorders resulted in 209 papers, of which only 8 papers (3.8%) were in MS, compared to 61 (29.1%) in Alzheimer's, 21 (10.0%) in schizophrenia, and 20 (9.6%) in depression [17].

1.4.1 Machine Learning in Multiple Sclerosis

Most applications of AI in MS are for detection, disease course classification, or differential diagnosis of MS from other neurological disorders. In [18], RF was used to classify MS patient disease course using clinical and lesion MR metabolic features, and was able to obtain F1-scores, the harmonic average of precision and recall, of up to 87%. SVM was used in [19] to differentiate RRMS patients from healthy volunteers with 89% accuracy using fractional anisotropy maps, structural and functional connectivity extracted from MR images, and in [20] to differentiate between the MS disease courses using grey matter measures and functional connectivity patterns extracted from MR images.

Predictive applications of AI in MS have mostly been focused on the prediction of conversion from CIS to MS since time matters in the management of MS – earlier diagnosis allows for earlier treatment, resulting in longer life expectancies [21]. SVMs have been used to predict CIS to MS conversion within 1 and 3 years with 71.4% and 68% accuracy

respectively from lesion features and clinical/demographic characteristics in [22]. A random forest used in [23] was able to predict CIS-MS conversion within 3 years with 84.5% accuracy using shape and intensity features extracted from computer-assisted manual lesion segmentations. In [24], SVMs were used to predict 2-year CIS-MS conversion from image-based lesion geometric features and clinical/demographical features with 70.4% accuracy. Only one study has evaluated the use of machine learning for predicting binary disability progression with an ensemble of linear SVM, using longitudinal clinical, demographical, and MRI data [25]. While they achieved an overall prediction sensitivity up to 86%, this was only observed in individuals with low disability scores.

1.4.2 Deep Learning in Multiple Sclerosis

While deep learning has been used for unsupervised feature learning from MR images that correlate with clinical scores [26] and for segmentation tasks [27][28], clinical deep learning applications for MS detection are fairly limited. Yoo et al. used a DLN in [29] to learn spatial features from multimodal MR images for differentiating between MS patients and healthy volunteers with an accuracy of 87.9%, and in [30] for the differential diagnosis of MS from Neuromyelitis Optica spectrum disorders with 81.3% accuracy. For prediction, Yoo et al. developed a DLN that extracted predictive features from brain lesion patterns [31]. These features were then used in conjunction with user-defined clinical and MRI features to predict CIS-MS conversion with 75.0% accuracy.

1.5 Motivation

Although studies of clinical machine learning and deep learning applications in multiple sclerosis exist, they are heavily skewed towards MS detection and disease course

classification, differential diagnosis, and prediction of CIS to MS conversion. In regard to the prediction of disability progression, there has only been one study that evaluated machine learning on a population skewed towards low disability.

Early diagnosis and treatment of MS is important, and understandably, more research focus has been on the prediction of conversion from CIS to MS, but it is also important not to neglect individuals that are in the later stages of their disease course and/or have higher disability than newly diagnosed individuals. Alas, there exists a knowledge gap with respect to the use of artificial intelligence for the prediction of disability progression in individuals with moderate disability (i.e. PPMS and SPMS). Both PPMS and SPMS are characterized by increasing disability over time - their unpredictability, in addition to the research gap, makes the task of predicting disability progression in SPMS enticing and valuable.

Existing research on applications of machine learning in multiple sclerosis then raises two simple questions. Firstly, is there, if any, added prognostic value to using conventional machine learning techniques for predicting disability progression in SPMS? And secondly, can DLNs learn features from 3-dimensional imaging data, as it does for predicting CIS-MS conversion in [31] and differential diagnosis of neuromyelitis optical spectrum disorders from MS in [30], that have prognostic value for disability progression prediction in SPMS?

1.6 Thesis Contributions

This thesis presents three main contributions:

1. **Short-term binary confirmed disability progression prediction in SPMS from user-defined features using non-parametric machine learning approaches:** SVM and RF have been shown to perform well for MS disease course classification and

prediction of CIS-MS conversion using user-defined features. We implemented and evaluated four conventional ML classifiers: LR, and three ensemble classifiers (linear SVM, RF, and AdaBoost-DT), for predicting 18-month confirmed disability progression in SPMS using only baseline clinical, demographical, and pre-defined MRI features. We show that non-parametric ML (RF and AdaBoost-DT) has higher predictive performance for predicting short-term disability progression than parametric approaches and prevalence-based prediction when the EDSS predictor was preprocessed as a continuous input variable.

2. **Short-term binary confirmed disability progression prediction in SPMS using deep learned features from brain lesion masks:** Deep learning has been shown to automatically extract features from brain lesion masks for predicting CIS to MS conversion. We explored whether it can learn features from brain lesion masks to predict disability progression in SPMS. A DLN was developed and trained to automatically extract features from brain lesion masks. Predictive performance of deep-learned features was evaluated with and without the use of user-defined features against LR using only user-defined features. We show that the DLN is able to learn lesion mask features with greater predictive value than user-defined features for predicting disability progression when EDSS was analyzed as a continuous variable.
3. **Impact of continuous vs. categorical analysis of EDSS on conventional machine learning and deep learning performance in predicting SPMS disability progression:** We evaluated the performance of ML and DL models for predicting SPMS disability progression when EDSS was used as a categorical variable in addition to its use as a continuous variable and showed that linear parametric ML

models, LR and enSVM, performed better when EDSS was treated as a categorical variable as opposed to a continuous variable. The non-parametric ML models, RF and AdBDT, had similar performance regardless of how EDSS was used. Non-parametric ML models were less affected by how EDSS was analyzed with respect to feature contributions to model training. DLNs were also robust to the treatment of EDSS – features were extracted from brain lesion masks independent of EDSS. We showed that non-parametric ML models are more robust to data handling and are likely the models of choice when using data without domain specific knowledge or information regarding proper data preprocessing.

Chapter 2:

Materials and Generation of User-Defined Features

The BioMS dataset is comprised of clinical, demographical, and MRI data from a negative 2-year randomized, double-blind, placebo-controlled phase III study with participation from 47 centers across 10 countries that evaluated the efficacy and safety of MBP8298 in patients diagnosed with SPMS. The detailed study design can be found in [32].

2.1 Brain Lesion Masks

Binary brain lesion masks were generated using a semi-automatic 2-D region growing technique used in [33] from T2 and PDw MR images with dimensions 256 x 256 x 50 and voxel dimensions of 1mm x 1mm x 3mm. Seed points initially placed on lesions by radiologists were interactively grown by trained technicians, constrained by automatically generated sample points of white matter (WM), grey matter (GM), and cerebral spinal fluid (CSF) closest to the selected seed for lesion growing. The methods used for automated sampling of WM, GM and CSF are also detailed in [33]. Figure 2-1 illustrates an example of the semi-automatic method for brain lesion mask generation.

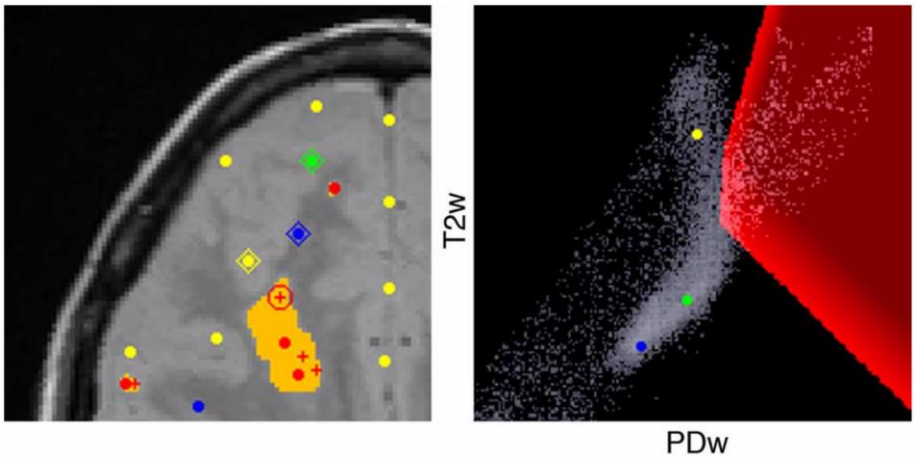

Figure 2-1 Semi-automatic method used for generating brain lesion masks. Left: A PDw scan with automatically generated sample points (blue = WM, green = GM, yellow = CSF) and radiologist-planted lesion seed points (red dots). Lesions are first grown from the seed points, but additional supporting points can be added (red +) if the grown lesion is not adequate. Lesions are grown from a selected red dot or + (circled in red), and is constrained by the closest WM, GM, and CSF dots (enclosed in diamonds). The grown lesion is the orange area. Right: a T2w/PDw histogram with WM, GM and CSF illustrated. The red area is the intensity space that the region can grow towards. Source: [33]

2.2 User-Defined Features

Clinical features were comprised of baseline EDSS score, MSFC, and the MSFC component Z-scores (9HPT, T25W, PASAT). Demographical features included disease duration and age in years at baseline, as well as biological sex. MRI features included baseline T2 lesion volume (burden of disease, BOD) and brain parenchymal fraction (BPF). BOD was calculated by multiplying the voxel volume of a brain lesion mask with the voxel dimensions. For example, a brain lesion mask with 100 lesion voxels with voxel dimensions of 2mm by 2mm by 2mm would have a BOD of 800 mm^3. BPF was calculated using

Equation (2.1) from the volume of the intradural space, $V_{intradural,}$ and CSF volume, V_{CSF}, calculated from intradural and CSF masks [34].

$$BPF = \frac{V_{intradural} - V_{CSF}}{V_{intradural}} \tag{2.1}$$

2.3 Confirmed Disability Progression Definition

Time to confirmed disability progression t_{CDP} was determined as the time from baseline until an EDSS increase greater than or equal to 1.0 was observed in individuals with a baseline EDSS less than or equal to 5.5, or an increase greater or equal to 0.5 was observed in individuals with a baseline EDSS greater than 5.5.

Subjects were labelled as positive (CDP+), for confirmed disability progression (CDP) if t_{CDP} was within 24 months of baseline. Those whose initial increase occurred after 18 months of baseline were labelled negative (CDP-) since individuals with t_{CDP} > 18 months were unable to have their EDSS confirmed 6 months later (their confirmation date surpasses the study end date).

2.4 Filtering

539 of 612 randomized subjects (88%) completed the study. Data from both control and treatment arms of the MBP8298 study was filtered to remove participants with multiple missing visits or data entries at any given visit. This included participants that did not have a complete set of baseline clinical scores (EDSS, MSFC, 9HP, T25W, PASAT) or missing baseline BOD or BPF. Imputation was not performed for participants missing multiple data entries for multiple reasons. Imputation would require assumptions be made regarding the

underlying population distribution. Additionally, within a short time-frame, consecutive clinical and MRI measurements are known to be noisy. Imputing missing temporal values with interpolation or extrapolation is unlikely to accurately approximate the true value. Only one missing disease duration (time since first MS diagnosis) was replaced with the mean diagnosis duration of the study cohort. One missing disease duration entry was replaced with the mean disease duration of the study sample. A total of 485 subjects were retained. Data breakdown is illustrated in Figure 2-2.

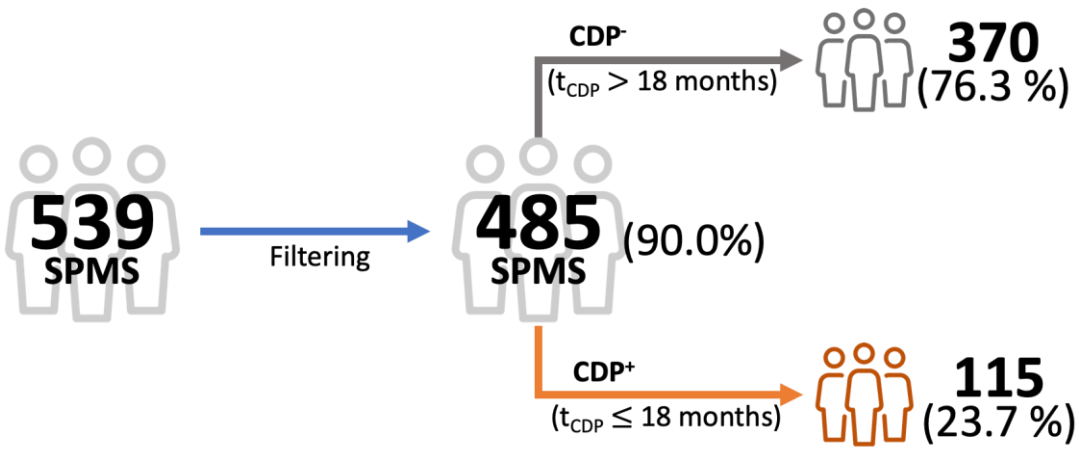

Figure 2-2 Dataset Breakdown. Of the whole dataset, 485 of 539 (90%) was used, and only 23.7% progressed within 18 months.

The characteristics of the baseline features of the 485 patients included in the study sample can be found in Table 2-1.

Table 2-1 Characteristics of user-defined demographical, clinical, and MRI features

	CDP+ (n = 115)	CDP- (n = 370)	Overall (n = 485)
Demographical Features			
# of Females	74 (64.3%)	237 (64.1%)	311 (64.1%)
Mean age [years] (SD)	50.3 (8.2)	51.1 (7.9)	50.9 (8.0)
Mean duration[a] [years] (SD)	9.1 (4.4)	9.3 (5.1)	9.3 (5.0)
Clinical Features			
Median EDSS (25^{th}, 75^{th} %tile)	6.0 (4.5, 6.0)	6.0 (4.5, 6.5)	6.0 (4.5, 6.5)
Mean T25W[b] [Z] (SD)	0.08 (1.52)	0.05 (1.54)	0.06 (1.54)
Mean 9HP[b] [Z] (SD)	-0.02 (0.93)	0.07 (0.95)	0.05 (0.95)
Mean PASAT[b] [Z] (SD)	0.05 (1.02)	0.01 (1.00)	0.02 (1.01)
Magnetic Resonance Imaging Biomarkers			
Median T2 BOD [mm^3] (25^{th}, 75^{th} %tile)	10403.9 (3392.5, 19796.4)	9012.0 (3730.3, 19889.3)	9321.4 (3621.6, 19872.8)
Mean BPF (SD)	0.7559 (0.0473)	0.7520 (0.0474)	0.7530 (0.0476)

[a] Disease duration (time since first MS diagnosis), [b] Standardized to the Task Force Dataset [5]

Chapter 3:

Machine Learning for Predicting Short-term Confirmed Secondary Progressive Multiple Sclerosis Progression

3.1 Overview

An ensemble of linSVM (enSVM) as suggested by [25], a random forest, and AdaBoost-DT (AdBDT), an AdaBoost classifier constructed with decision trees, were evaluated against the logistic regression classifier for predicting 18-month binary confirmed disability progression using user-defined clinical, demographic, and MRI features only. Generalizability was estimated using 10-fold stratified cross validations (10CV).

Data analysis and experiments were performed in Python 3.6. All classifiers were built and trained using Scikit-learn 0.21 with default parameters [35]. Statistical analyses were performed using Pandas 0.23.4 [36] and SciPy 1.1.0 [37].

3.2 Classifier Training and Evaluation with 10CV

Classifiers were trained and evaluated for generalizability using 10-fold stratified cross validations (10CV). The 485 subjects were shuffled and split into 10 non-overlapping groups with approximately the same class frequencies as the whole sample; this allowed for ten cycles (folds) of training and validation (Figure 3-1). Each fold used one unique group (containing 10% of the subjects) for validation while the remaining groups (90% of subjects) were used for training each classifier.

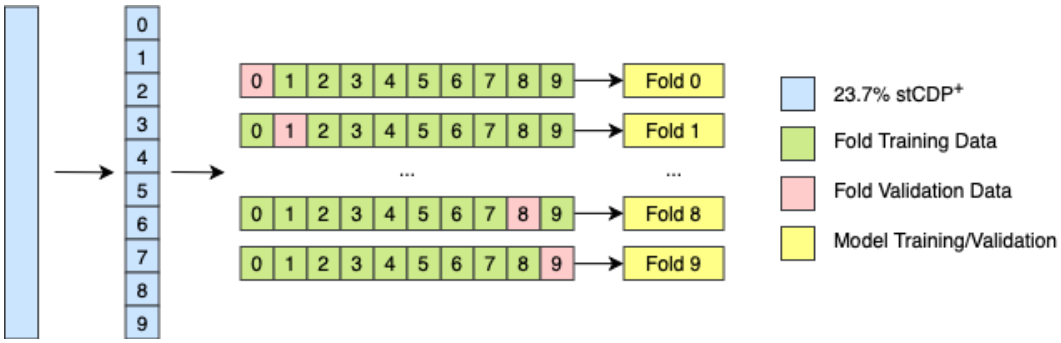

Figure 3-1 Example of 10-fold stratified cross validation where training and validation data for each fold have same class proportions as the whole sample

3.2.1 Data Preprocessing

Classifiers were trained to predict 18-month confirmed disability progression using the user-defined features discussed in Section 2.2. Each user-defined feature (with the exception of sex) in the training data of each 10CV fold were transformed by removal of median values and data scaled according to the interquartile range. Statistics calculated from the training data were then used to scale the validation data.

3.2.2 Class Imbalance

As can be seen in Figure 2-2, the dataset has slightly over three times more CDP- than CDP+ individuals. To prevent classifiers from biasing learning and predictions for CDP-, random under-sampling was applied on the training data for each fold of 10x10CV prior to being used by classifiers for training. Random under-sampling randomly selects CDP- patients to omit from classifier training so that data presented to classifiers have equal class representation.

3.2.3 Model Parameters

enSVM is a 10-classifier ensemble of linSVM. Each individual linSVM was trained on a randomly under-sampled subset of the training data. The enSVM class output is the average probabilistic output of the ten individual linSVMs.

The random forest classifier was constructed with 100 decision trees, each trained using two randomly chosen user-defined features from a bootstrapped sample from randomly under-sampled training data.

AdaBoost-DT is an AdaBoost classifier constructed from 50 decision tree stumps (max tree depth of 1) each trained on the same class-balanced dataset following the AdaBoost training algorithm described in Section 1.3.1.4.2.

The logistic regression classifier fit a logistic regression model on the class-balanced dataset using L2 regularization.

3.2.4 Performance Evaluations

The overall performance of each model was estimated by their ability to separate classes (CDP+ and CDP-) and to predict progression (CDP+) or non-progression (CDP-), by averaging the performance on the validation datasets in each 10-CV cycle.

The area under the receiver-operator characteristic curve (AUC) was used as the primary outcome. AUC summarizes each models' ability to separate the two classes. An AUC of 50% indicates no better than random separation, AUC of 0% indicates inversed class separation (i.e., all CDP+ classified as CDP-, and vice versa), while an AUC of 100% indicates perfectly separated classes.

To assess performance on predicting progression, precision/positive predictive value (PPV), change in pre- to post-positive predictive value (ΔPPV), and recall were used, and are defined as in Equations (3.1), (3.2), and (3.3) respectively.

$$Precision/PPV = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{3.1}$$

$$\Delta PPV = PPV - Prevalence_{CDP+} \tag{3.2}$$

$$Recall/Sensitivity = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{3.3}$$

Precision, or positive predictive value (PPV) is the proportion of predicted progressors that progressed. Change in pre- to post-positive predictive value (ΔPPV) shows the change in probability that an individual predicted to progress will progress compared to the baseline likelihood defined by the prevalence of progression.

Model performance in predicting non-progression was evaluated using the following negative predictive value (NPV), change in pre- to post-negative predictive value (ΔNPV), and specificity, and are defined in Equations (3.4), (3.5), and (3.6) respectively.

$$NPV = \frac{True\ Negatives}{True\ Negatives + False\ Negatives} \tag{3.4}$$

$$\Delta NPV = NPV - Prevalence_{CDP-} \tag{3.5}$$

$$Specificity = \frac{True\ Negatives}{True\ Negatives + False\ Positives} \tag{3.6}$$

Like PPV, NPV is the proportion of predicted non-progressors that did not progress. ΔNPV is the change in probability that an individual predicted to be CDP- does not progress compared to the baseline likelihood of non-progression defined by the prevalence of non-progression. Specificity is the percentage of CDP- that were correctly classified as CDP-.

3.2.5 EDSS analysis as categorical variable

Despite the Kurtzke Expanded Disability Status Scale (EDSS) commonly used as a continuous variable due to its characterization as a range from 0 to 10 in 0.5 increments, it is in fact an ordered categorical MS clinical disability scale. To evaluate the impact of EDSS treatment on classifier performance, categorical EDSS was assessed in addition to the primary analysis of EDSS as a continuous variable for predicting disability progression in SPMS.

3.2.6 Feature Importance to Classifier Training

As each classifier learns differently (e.g. parametric versus non-parametric, linear versus non-linear, etc.), we examined the importance of the user-defined features for training each classifier. The contribution, C, of each feature x in the logistic regression and ensemble SVM classifier was calculated from the classifier coefficients c and represented as a percentage using Equation (3.7).

$$C(x) = \frac{|c_x|}{\sum_{i=0}^{8}|c_i|} \times 100\% \tag{3.7}$$

RF and AdaBoost predictor importance were determined by their individual impact on decreasing impurity at a tree/forest node (see Section 1.3.1.3) and was extracted from the classifier at the end of its training.

3.2.7 Statistical Analysis

Paired t-tests with a significance threshold of $P < .05$ were performed on all evaluated performance metrics to compare classifier generalizability.

3.3 Experimental Results

Classifiers were evaluated based on their classification performance on the validation data, as well as the importance of each feature on the training of the classifiers, for each fold of the 10 repeated 10-fold cross validations.

3.3.1 Classifier Performance

Classifier performance was evaluated when EDSS was treated as a continuous and categorical variable separately.

3.3.1.1 EDSS as a Continuous Variable

A summary of model AUC performance can be seen in Table 3-1. When the LR model was applied to the validation data, the model assigned the wrong class more often than the correct class (AUC = 44.7%) which indicates an inability to identify a generalizable decision boundary. In contrast, the remaining models performed better than random guessing. AdaBoost produced the highest AUC, achieving a 15.5% improvement compared to LR, and 8.1% compared to enSVM. No significant difference was observed between AdaBoost and RF AUC.

Table 3-1 Summary of area under the curve validation performance for logistic regression (LR), ensemble of linear support vector machines (enSVM), random forest (RF) and AdaBoost-DT (AdBDT) when EDSS was treated as a continuous variable

Ref. Model	% AUC n = 10			Mean % AUC Difference [a] n = 10, *df* = 9					
				enSVM-Ref.		RF-Ref.		AdBDT-Ref.	
	Mean	**SD**	**Error[b]**	95% CI	*P*	95% CI	*P*	95% CI	*P*
LR	44.7	6.3	14.3	(-1.5, 16.3)	0.09	**(3.3, 19.7)**	**<.01**	**(9.4, 21.6)**	**<.001**
enSVM	52.1	7.3	16.4			(-3.5, 11.7)	0.26	**(2.6, 13.6)**	**<.01**
RF	56.2	9.6	21.8					(-2.0, 10.0)	0.17
AdBDT	60.3	4.3	9.6						

[a] paired t-test, [b] 95% margin of error

AdaBoost outperformed enSVM and LR in terms of precision by 5.3%, and 6.3% respectively. No significant ΔPPV was observed in logistic regression and SVM, while random forest and AdaBoost both performed better than prevalence-based random classification with ΔPPVs of 3.6% ($P < .05$) and 5.3% ($P < .0001$) respectively. These findings are summarized in Table 3-2.

Table 3-2 Summary of validation precision and change from pre- to post-positive predictive value of logistic regression (LR), ensemble of linear support vector machines (enSVM), random forest (RF) and AdaBoost-DT (AdBDT) when EDSS was treated as a continuous variable

Ref. Model	% Precision n = 10			Mean % Precision Difference [a] n = 10, *df* = 9						Mean % ΔPPV[c] n = 10
				enSVM-Ref.		RF-Ref.		AdBDT-Ref.		
	Mean	**SD**	**Error[b]**	95% CI	*P*	95% CI	*P*	95% CI	*P*	
LR	22.7	4.5	10.2	(-5.5, 7.5)	0.73	**(0.6, 8.6)**	**0.03**	**(2.4, 10.2)**	**<.01**	-1.0
enSVM	23.7	6.0	13.7			(-1.7, 8.9)	0.16	**(1.0, 9.6)**	**0.02**	-0.0
RF	27.3	4.2	9.4					(-2.1, 5.5)	0.35	3.6*
AdBDT	29.0	2.6	5.8							5.3*

[a] paired t-test, [b] 95% margin of error, [c] compared to progression prevalence of 23.7%, * statistically significant ΔPPV at $P < .05$

When assessing each model's sensitivity (its ability to correctly identify CDP+ from all CDP+), logistic regression and enSVM only identified 49.0% and 50.5% of CDP+, whereas RF and AdBDT were able to sensitivity 54.9% and 60.9% of CDP+. No significant differences were observed between model sensitivity. A summary of model sensitivity is shown in Table 3-3.

Table 3-3 Summary of validation sensitivity of logistic regression (LR), ensemble of linear support vector machines (enSVM), random forest (RF) and AdaBoost-DT (AdBDT) when EDSS was treated as a continuous variable

Ref. Model	% Sensitivity n = 10			Mean % Sensitivity Difference [a] n = 10, *df* = 9					
				enSVM-Ref.		RF-Ref.		AdBDT-Ref.	
	Mean	SD	Error[b]	95% CI	*P*	95% CI	*P*	95% CI	*P*
LR	49.0	15.2	34.3	(-18.9, 21.7)	0.88	(-4.5, 16.3)	0.23	(-0.6, 24.4)	0.06
enSVM	50.5	17.9	40.6			(-11.3, 20.3)	0.54	(-3.0, 24.0)	0.11
RF	54.9	11.0	24.9					(-3.6, 15.6)	0.19
AdBDT	60.9	11.7	26.5						

[a] paired t-test, [b] 95% margin of error

We also considered model performance on detecting the larger proportion of CDP- by assessing their negative predictive values (Table 3-4) and specificity (Table 3-5).

Logistic regression correctly identified CDP- 75.7% of the time while enSVM correctly identified 77.3% of CDP-. Both RF and AdBDT outperformed LR with mean NPVs of 79.6% and 82.0% respectively. AdBDT was able to increase CDP- accuracy over prevalence-based random prediction with a ΔNPV of 5.7% ($P < .001$).

Table 3-4 Summary of validation negative predictive value and change from pre- to post-negative predictive value of logistic regression (LR), ensemble of linear support vector machines (enSVM), random forest (RF) and AdaBoost-DT (AdBDT) when EDSS was treated as a continuous variable

Ref. Model	% NPV n = 10			Mean NPV Difference [a] n = 10, *df* = 9						Mean % ΔNPV[c] n = 100
				enSVM-Ref.		RF-Ref.		AdBDT-Ref.		
	Mean	**SD**	**Error[b]**	95% CI	*P*	95% CI	*P*	95% CI	*P*	
LR	75.7	5.0	11.2	(-4.7, 7.9)	0.58	**(0.1, 7.5)**	**0.04**	**(2.3, 10.1)**	**<.01**	-0.6
enSVM	77.3	5.5	12.4			(-3.0, 7.6)	0.36	**(0.3, 9.1)**	**0.04**	1.0
RF	79.6	4.9	11.0					(-1.4, 6.2)	0.19	3.3
AdBDT	82.0	3.5	8.0							5.7*

[a] paired t-test, [b] 95% margin of error, [c] compared to non-progression prevalence of 76.3%, * statistically significant ΔNPV at $P < .05$

Logistic regression identified less than half of the individuals without progression. enSVM, random forest and AdBDT were identified more than half (50.8%, 54.6% and 54.1% respectively) of the non-progressors. No statistically significant differences were observed between the various machine learning models with respect to specificity.

Table 3-5 Summary of validation specificity of logistic regression (LR), ensemble of linear support vector machines (enSVM), random forest (RF) and AdaBoost-DT (AdBDT) when EDSS was treated as a continuous variable

Ref. Model	% Specificity n = 10			Mean % Specificity Difference [a] n = 10, *df* = 9					
				enSVM-Ref.		RF-Ref.		AdBDT-Ref.	
	Mean	**SD**	**Error[b]**	95% CI	*P*	95% CI	*P*	95% CI	*P*
LR	48.9	9.5	20.1	(-6.8, 10.5)	0.63	(-1.7, 13.1)	0.12	(-0.9, 11.1)	0.09
enSVM	50.8	7.0	15.7			(-2.0, 9.6)	0.17	(-2.8, 9.2)	0.25
RF	54.6	5.5	12.5					(-5.3, 4.3)	0.80
AdBDT	54.1	5.1	11.5						

[a] paired t-test, [b] 95% margin of error

3.3.1.2 EDSS as a Categorical Variable

While RF and AdBDT had greater AUCs with continuous EDSS, the analysis of EDSS as a categorical variable resulted in enSVM achieving the greatest AUC of 67.6%. enSVM outperformed LR, RF, and AdBDT by 8.0%, 7.1% and 9.7% respectively. Results are summarized in Table 3-6.

Table 3-6 Summary of area under the curve validation performance for logistic regression (LR), ensemble of linear support vector machines (enSVM), random forest (RF) and AdaBoost-DT (AdBDT) when EDSS was treated as a categorical variable

Ref. Model	% AUC n = 10			Mean % AUC Difference [a] n = 100, *df* = 99					
				enSVM-Ref.		RF-Ref.		AdBDT-Ref.	
	Mean	**SD**	**Error[b]**	95% CI	*P*	95% CI	*P*	95% CI	*P*
LR	59.6	11.1	25.1	**(4.2, 11.7)**	**<.01**	(-3.8, 5.6)	0.67	(-7.9, 4.3)	0.53
enSVM	67.6	9.3	21.1			**(-11.4, -2.8)**	**<.01**	**(-13.9, -5.6)**	**<.01**
RF	60.5	12.5	28.4					(-8.6, 3.3)	0.34
AdBDT	57.9	7.3	16.6						

[a] paired t-test, [b] 95% margin of error

No significant differences in precision were observed between classification models when categorical EDSS was used. All models performed better than prevalence-based random classification (Table 3-7).

Table 3-7 Summary of validation precision and change from pre- to post-positive predictive value of logistic regression (LR), ensemble of linear support vector machines (enSVM), random forest (RF) and AdaBoost-DT (AdBDT) when EDSS was treated as a categorical variable

Ref. Model	% Precision n = 10			Mean % Precision Difference [a] n = 10, *df* = 9						Mean % ΔPPV[c] n = 10
				enSVM-Ref.		RF-Ref.		AdBDT-Ref.		
	Mean	SD	Error[b]	95% CI	*P*	95% CI	*P*	95% CI	*P*	
LR	31.5	6.2	14.1	(-1.7, 4.6)	0.32	(-6.1, 3.9)	0.64	(-6.1, 2.7)	0.41	7.8*
enSVM	33.0	7.4	16.8			(-6.8, 1.7)	0.20	(-7.9, 1.5)	0.16	9.3*
RF	30.4	7.6	17.3					(-4.3, 3.1)	0.72	6.7*
AdBDT	29.8	4.1	9.2							6.1*

[a] paired t-test, [b] 95% margin of error, [c] compared to progression prevalence of 23.7%, * statistically significant ΔPPV at $P < .05$

No significant differences were observed in model sensitivity performance when EDSS was treated as a categorical variable (Table 3-8).

Table 3-8 Summary of validation sensitivity of logistic regression (LR), ensemble of linear support vector machines (enSVM), random forest (RF) and AdaBoost-DT (AdBDT) when EDSS was treated as a categorical variable

Ref. Model	% Sensitivity n = 10			Mean % Sensitivity Difference [a] n = 10, *df* = 9					
				enSVM-Ref.		RF-Ref.		AdBDT-Ref.	
	Mean	SD	Error[b]	95% CI	*P*	95% CI	*P*	95% CI	*P*
LR	63.6	16.8	38.0	(-3.6, 5.4)	0.66	(-15.5, 0.1)	0.05	(-13.1, 7.8)	0.58
enSVM	64.5	17.3	39.2			(-18.0, 0.8)	0.07	(-14.4, 7.2)	0.47
RF	54.2	15.2	34.3					(-2.8, 12.9)	0.18
AdBDT	58.3	12.6	28.5						

[a] paired t-test, [b] 95% margin of error

With respect to negative predictive value, the treatment of EDSS as a categorical variable resulted in enSVM outperforming RF by 3.5%. All models performed better than prevalence-based prediction of non-progression. Results are summarized in Table 3-9.

Table 3-9 Summary of validation negative predictive value and change from pre- to post-positive predictive value of logistic regression (LR), ensemble of linear support vector machines (enSVM), random forest (RF) and AdaBoost-DT (AdBDT) when EDSS was treated as a categorical variable

Ref. Model	% NPV n = 10			Mean NPV Difference [a] n = 10, *df* = 9						Mean % ΔNPV[c] n = 10
				enSVM-Ref.		RF-Ref.		AdBDT-Ref.		
	Mean	**SD**	**Error[b]**	95% CI	*P*	95% CI	*P*	95% CI	*P*	
LR	84.0	5.8	13.2	(0.8, 2.4)	0.31	(-6.3, 0.7)	0.10	(-5.8, 2.4)	0.37	7.7*
enSVM	84.7	5.9	13.3			**(-6.7, -0.5)**	**0.03**	(-6.0, 1.1)	0.16	8.4*
RF	81.2	6.6	14.9					(-2.1, 4.3)	0.45	4.9*
AdBDT	82.3	5.0	11.3							6.0*

[a] paired t-test, [b] 95% margin of error, [c] compared to non-progression prevalence of 76.3%, * statistically significant ΔNPV at $P < .05$

No significant differences were observed between model specificity when EDSS was analyzed as a categorical variable. Findings are summarized in Table 3-10.

Table 3-10 Summary of validation specificity of logistic regression (LR), ensemble of linear support vector machines (enSVM), random forest (RF) and AdaBoost-DT (AdBDT) when EDSS was treated as a categorical variable

Ref. Model	% Specificity n = 10			Mean % Specificity Difference [a] n = 10, *df* = 9					
				enSVM-Ref.		RF-Ref.		AdBDT-Ref.	
	Mean	**SD**	**Error[b]**	95% CI	*P*	95% CI	*P*	95% CI	*P*
LR	57.6	6.9	15.5	(-2.3, 6.6)	0.30	(-4.9, 9.2)	0.51	(-6.2, 2.4)	0.34
enSVM	59.7	7.1	16.2			(-5.5, 5.5)	1.00	(-9,2, 1.1)	0.11
RF	59.7	9.0	20.5					(-10.7, 2.6)	0.20
AdBDT	55.7	5.1	11.6						

[a] paired t-test, [b] 95% margin of error

3.3.2 Feature Importance on Classifier Training

Feature importance on classifier training was assessed for when EDSS was treated as a continuous variable, and separately for when it was treated as a categorical variable. To

examine the influence of each predictor on model output, we looked at how much each predictor contributed to each model and noticed qualitative differences in predictor importance for each linear model (LR and RF), and each non-linear model (RF and AdBDT).

3.3.2.1 EDSS as a Continuous Variable

Continuous EDSS played a larger role in prediction (composing 22.0% and 30.2% of LR and enSVM respectively), while T25W played the smallest role (2.5% and 1.4% of LR and enSVM respectively). Sex contributed more to LR (11.6%) and enSVM (7.6%) than it did to the better performing non-linear models – only contributing to 1.8% of the random forest model and 0.3% with the AdBDT. Table 3-11 summarizes the findings.

Table 3-11 Feature importance on classifier training of logistic regression (LR), ensemble of linear support vector machines (enSVM), random forest (RF) and AdaBoost with decision trees (AdBDT) when EDSS was treated as a continuous variable

Predictor	LR		enSVM		RF		AdBDT	
	Mean (SD)	*Error*[a]	*Mean (SD)*	*Error*[a]	*Mean (SD)*	*Error*[a]	*Mean (SD)*	*Error*[a]
Age	10.5 (2.6)	9.6	7.4 (5.1)	11.6	11.0 (0.5)	1.2	7.8 (4.2)	9.4
Sex	12.8 (7.4)	16.7	6.7 (6.3)	11.7	1.8 (0.2)	0.5	0.8 (1.0)	2.3
Dur. [b]	7.3 (4.2)	9.4	4.3 (3.9)	8.9	10.7 (0.8)	1.9	4.4 (2.8)	6.3
Cont. EDSS	23.5 (8.6)	19.3	32.6 (6.7)	15.2	10.3 (0.9)	1.9	11.4 (2.8)	6.4
T25W	3.5 (4.1)	9.2	1.4 (1.0)	2.2	16.0 (1.3)	2.9	19.4 (5.6)	12.6
9HP	15.2 (8.8)	19.9	21.3 (3.4)	7.6	14.8 (1.0)	2.2	25.0 (4.2)	9.6
PASAT	9.5 (9.4)	21.3	6.5 (5.2)	11.9	10.2 (1.0)	2.3	6.0 (4.7)	10.7
T2 BOD	7.3 (2.6)	6.0	2.8 (2.6)	6.0	12.7 (0.7)	1.7	16.0 (6.5)	14.8
BPF	10.5 (6.2)	14.1	17.1 (4.6)	10.4	12.6 (0.8)	1.8	9.2 (5.1)	11.5

[a] 95% margin of error, [b] Disease Duration

In regard to the distribution of predictor contribution, while all predictors (with the exception of sex) contributed fairly equally in random forest classification, enSVM relied more on continuous EDSS, 9HP, and brain parenchymal fraction. LR and AdaBoost were

intermediate of the enSVM and RF. A plot of the feature contributions to each model is shown in Figure 3-2.

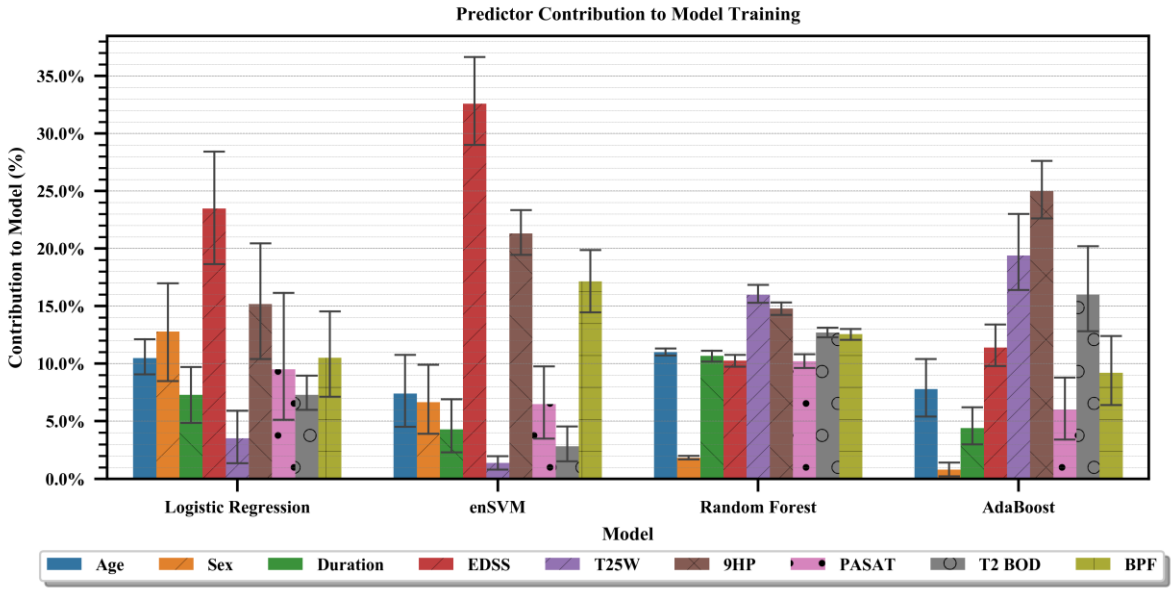

Figure 3-2 Feature importance to classifier training and predictions when EDSS was treated as a continuous variable, where EDSS = continuous EDSS

3.3.2.2 EDSS as a Categorical Variable

When EDSS was treated as a categorical variable, it became much more important than all other predictors in LR and enSVM, contributing to 78.5% and 98.7% of the model's training. Findings are summarized in Table 3-12.

Table 3-12 Feature importance on classifier training of logistic regression (LR), ensemble of linear support vector machines (enSVM), random forest (RF) and AdaBoost with decision trees (AdBDT) when EDSS was treated as a categorical variable

Predictor	LR		enSVM		RF		AdBDT	
	Mean (SD)	*Error*[a]	*Mean (SD)*	*Error*[a]	*Mean (SD)*	*Error*[a]	*Mean (SD)*	*Error*[a]
Age	4.8 (1.7)	3.9	0.1 (0.1)	0.2	10.6 (0.8)	1.7	8.2 (4.2)	9.4
Sex	3.8 (2.0)	4.5	0.1 (0.1)	0.1	1.9 (0.3)	0.6	0.8 (1.0)	2.3
Dur. [b]	1.5 (1.1)	2.4	0.1 (0.1)	0.2	9.7 (0.6)	1.2	4.8 (1.7)	3.8
Cat. EDSS	78.5 (5.6)	12.8	98.7 (0.1)	2,2	14.4 (0.9)	2.1	17.4 (1.6)	3.7
T25W	1.7 (2.1)	4.7	0.5 (0.3)	0.7	15.2 (0.8)	1.8	16.2 (4.3)	9.6
9HP	3.7 (1.8)	4.1	0.2 (0.2)	0.5	14.5 (1.2)	2.7	24.4 (4.9)	11.0
PASAT	1.8 (1.7)	3.9	0.1 (0.2)	0.4	10.0 (0.7)	1.5	4.6 (2.5)	5.7
T2 BOD	1.7 (1.5)	3.3	0.1 (0.2)	0.4	11.9 (0.7)	1.6	14.8 (4.6)	10.5
BPF	2.5 (1.8)	4.0	0.1 (0.1)	0.2	11.8 (0.4)	0.9	8.8 (5.2)	11.7

[a] 95% margin of error, [b] Disease duration

Unlike LR and enSVM which are linear models, the distribution of feature contribution to the training and predictions of both non-parametric models (RF and AdBDT) when EDSS was treated as a categorical variable (Figure 3-3) was similar to when EDSS was treated as a continuous variable (Figure 3-2). The disproportionate dependence of LR and enSVM on EDSS for model training when it was treated as a categorical variable can also be seen in Figure 3-3.

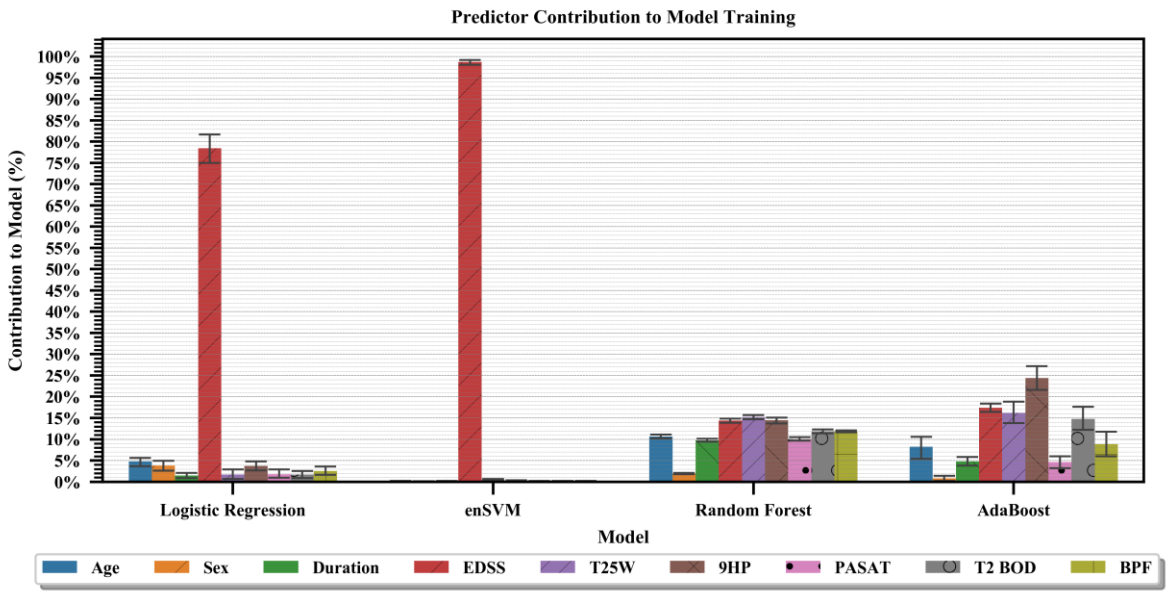

Figure 3-3 Feature importance to classifier training and predictions when EDSS was treated as a categorical variable, where EDSS = categorical EDSS

Chapter 4:

Automated Feature Extraction from Lesion Masks using Deep Learning for Predicting Short-term Confirmed Secondary Progressive Multiple Sclerosis Progression

4.1 Overview

The prognostic value of DLN-extracted brain lesion features was evaluated using a lesion mask DLN (lmDLN) classifier, which uses only lesion mask extracted features as independent variables, as well as a user-defined and deep-learned features combined DLN (coDLN) classifier. These DLNs were compared against L2-regularized LR using only user-defined clinical, demographic, and MRI features. Performance generalization was estimated using 10-fold cross validation.

All experiments, data processing, and statistical analyses were performed in Python 3.6 unless otherwise stated. Pandas 0.23.4 [36] and NumPy 1.15.4 [37] were used for data processing and statistical analysis. Logistic regression fitting using user-defined features was performed using Scikit-learn 0.21 [35]. The DLNs used for feature extraction and prediction were constructed and trained using Keras 2.1.6 [38] with Tensorflow 1.8.0 [39] on Nvidia Titan X graphics processing units.

4.2 Preprocessing of Brain Lesion Masks

4.2.1 Image Registration

Binary lesion masks with dimensions 256 x 256 x 50 and voxel dimensions 1 x 1 x 3 mm were generated by experts using a semi-automated method from T2w and PDw MRIs. The lesion masks were spatially aligned by applying transformations derived from the 12 degree-of-freedom affine registration used to align the T2w brain MR images to the MNI152 T1 1mm brain template and cropped to the same dimensions (182 x 218 x 182). Affine image registration was performed using FSL FLIRT [40, 41].

4.2.2 Signed Distance Transform

MS lesions are typically very dispersed, and the direct use of brain lesion masks can result in noise patterns being learned [31]. The signed Euclidean-distance transform [42] (EDT) was applied to the lesion masks to increase information density by assigning the Euclidean distance at each voxel to the closest lesion as the voxel intensity in the transformed image (Figure 1). EDT was applied using itk-SNAP's Convert3D tool [43]. A Gaussian filter (σ=2) was applied to the lesion masks before they were down-sampled by a factor of 2. To permit valid consecutive convolutions and pooling operations, the transformed lesion masks were padded from $91 \times 109 \times 91$ to $96 \times 112 \times 96$. Figure 4-1 shows a sample slice from a brain lesion mask and the same slice after the Euclidean-distance transform.

Figure 4-1 Euclidean distance transform of brain lesion mask. Left: Example slice of a brain lesion mask. Right: Slice from 3D Euclidean distance transform of lesion mask

4.3 Deep Learning Network Architectures

Identical CNNs were used to learn features from brain lesion masks in lmDLN and coDLN, while different DNNs were used for prediction of disability progression (depending on whether it used solely lesion distribution features, or combined them with user-defined clinical, demographic, and MRI features). An overview of data flow in both DLNs is shown in Figure 4-2.

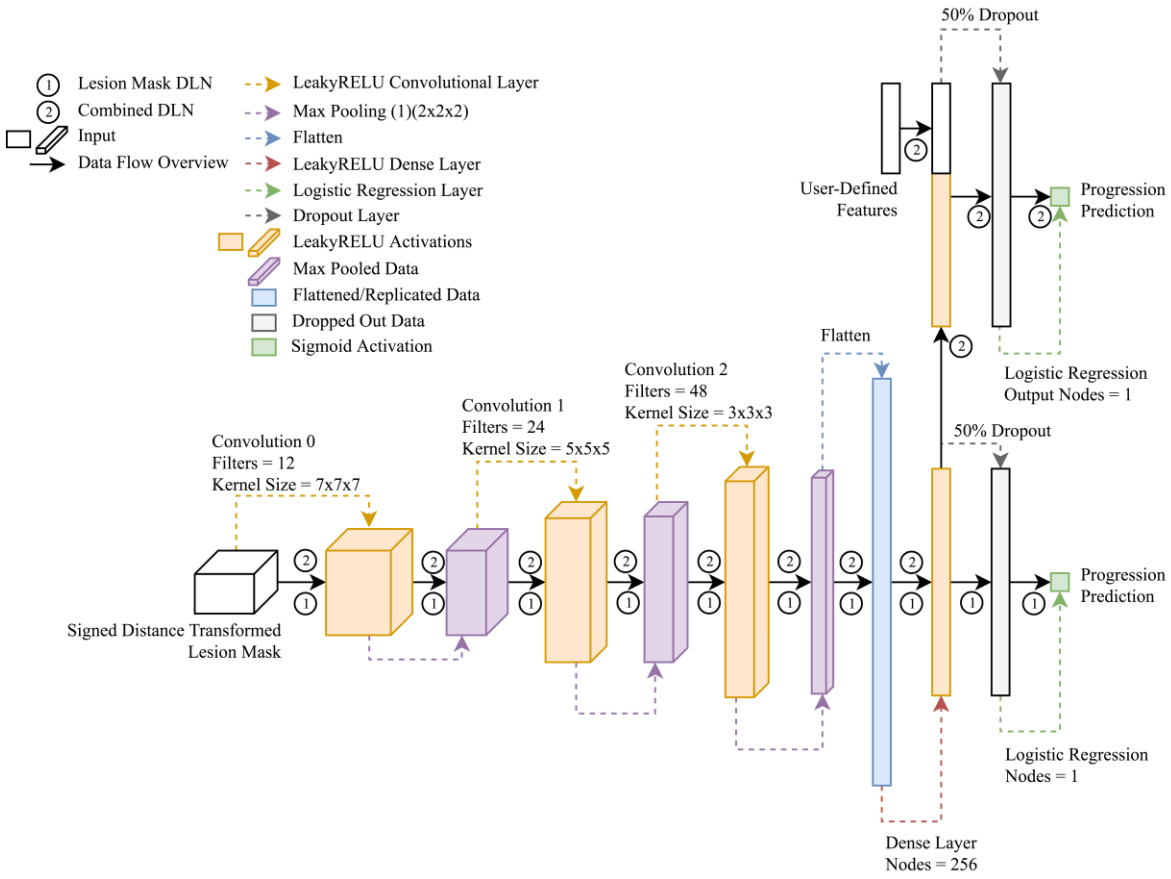

Figure 4-2 Overview of lesion mask deep learning network (lmDLN) and combined deep learning network (coDLN) data flow with identical CNN, differing DNN pathways, and dropout layers illustrated.

The CNN used for extracting features from the signed distance transformed lesion masks is comprised of three convolutional layers, each using leaky rectified linear unit (LeakyReLU) activation for introducing nonlinearity [44]. The convolutional layers consisted of 12, 24, and 48 filters of sizes 7x7x7, 5x5x5, and 3x3x3 respectively with max-pooling layers of size 2x2x2 used after each convolutional layer for dimensionality reduction. The output of the final max-pooling layer was then flattened into a one-dimensional feature vector and used as input to the DNNs. Figure 4-3 illustrates the CNN architecture used for learning lesion mask features.

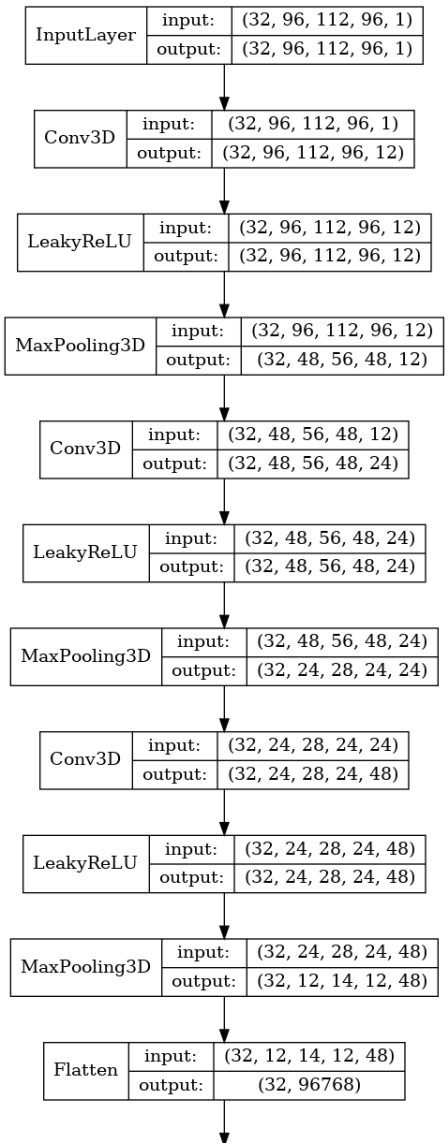

Figure 4-3 Detailed CNN architecture for both lmDLN and coDLN. Input and output refer to the input shape and output shape of each layer, arranged as (batch size, width, length, depth, channels) for 3D layers, and (batch_size, length) for 1D vectors. The CNN takes in signed distance transformed 3D lesion masks (InputLayer). The final flattened activations are fed into the DNNs.

The flattened features from the CNN were then passed into one dense layer with LeakyReLU activation to learn relationships between the 96,768 lesion mask features and reduce feature dimensionality to 256. In lmDLN, these 256 features were then passed as

independent variables into a logistic regression layer which performed classification. In coDLN, the 256 lesion mask features were concatenated with the user-defined clinical and demographic features before being passed into the logistic regression layer for classification. Logistic regression layers were constructed from a dense layer with sigmoid activation.

To regularize the DLNs, dropout layers with 50% dropout were placed before the logistic regression layer. During training, the dropout layers randomly set 50% of the activations of the preceding layer to zero so that in each training epoch, random units and connections are dropped; this has been shown to greatly reduce overfitting by preventing learnable units from co-adapting to the data [45]. When validating, the dropout layers were disabled, and layer weights were adjusted to reflect the dropout frequency impact on weight-learning. Both lmDLN and coDLN DNNs are illustrated in Figure 4-4.

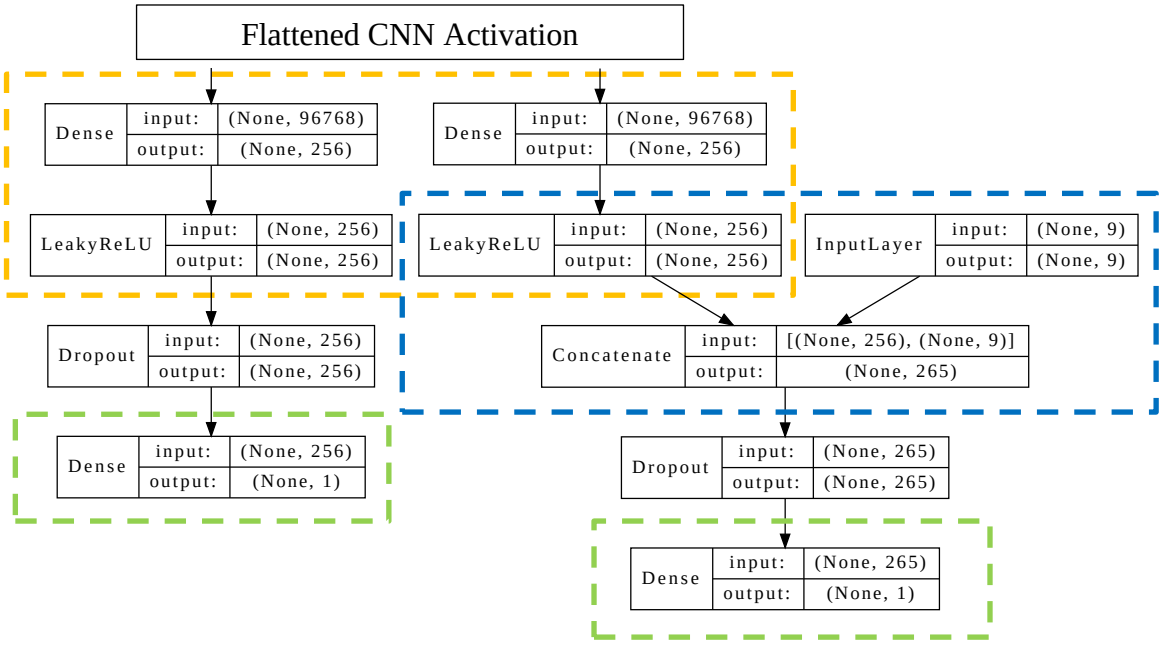

Figure 4-4 DNN for lmDLN (left) and coDLN (right). Input and output refer to the data shape (batch size, vector length). Dropout was applied during training only. Orange: Flattened CNN activations are passed through a dense layer and a LeakyReLU activation layer. Blue: Activations are concatenated with user-defined features. Green: Logistic regression layer is 1 output sigmoid-activated

All layer weights were initialized with the He normal initializer used in [46], which draws samples from a zero-centered, truncated normal distribution with a standard deviation of $\sqrt{\frac{2}{fan_{in}}}$, where fan_{in} is the number of input units to the weight tensor.

4.4 Training and Evaluation with 10CV

Ten-fold cross validation (as described in Section 3.2) was used to train and evaluate classifiers on their estimated generalization performance.

4.4.1 Data Processing

After class imbalance was corrected, the training data of each fold was scaled using the same outlier-robust approach discussed in Section 3.2.1. To scale the signed distance transform lesion masks, each pixel location was treated as an individual feature, and outlier-robust scaling was performed by calculating median and IQR statistics across the training dataset for each pixel location.

4.4.2 Class Imbalance

To prevent class imbalance from biasing DLN learning and LR fitting towards predicting non-progression (as non-progression has a class frequency of 76.3% versus progression with 23.7%), random under-sampling was performed on the training data of each fold in the 10CV. Details regarding random under-sampling can be found in Section 3.2.2.

4.4.3 Deep Learning Network Training Parameters

DLNs were trained for each fold of 10CV using the Adam optimizer as discussed in [15], with an initial learning rate of 1e-6, in mini-batches of 32, for 350 epochs.

4.4.4 Performance Evaluations

Classification performance of lmDLN, coDLN, and LR was evaluated on their ability to separate progressors from non-progressors, their ability to predict progression, as well as their ability to predict non-progression. The same metrics used in Chapter 3: were used here; additional details on the metrics can be found in Section 3.2.4.

4.4.5 EDSS analysis as categorical

As discussed in Section 1.1.2, EDSS is commonly used as a continuous variable despite it being an ordinal variable. The performance of logistic regression and coDLN was evaluated with EDSS analyzed as a continuous variable and as a categorical variable.

4.4.6 Statistical Analysis

Paired t-tests were performed on all metrics used to evaluate classifier generalizability. A significance threshold of $P < .05$ was used.

4.5 Experimental Results

4.5.1 EDSS as a Continuous Variable

While the conventional logistic regression was only able to achieve an AUC of 45.0%, both deep learning approaches performed significantly better. The lesion mask deep learning network performed 10.1% better (AUC=55.0%) while the addition of clinical, demographic, and user-defined MRI data in the coDLN (AUC=55.2%) did not improve performance. A summary of AUC performance can be seen in Table 4-1.

Table 4-1 Summary of area under the curve validation performance for logistic regression using only user-defined features (LR), lesion mask only deep learning network (lmDLN), and the combined user-defined and lesion mask features deep learning network (coDLN) when EDSS was treated as a continuous variable

Ref. Model	% AUC n = 10			Mean % AUC Difference [a] n = 10, *df* = 9					
				lmDLN – Ref.			coDLN – Ref.		
	Mean	**SD**	**Error[b]**	Mean	95% CI	*P*	Mean	95% CI	*P*
LR	45.0	8.3	6.2	**10.0**	**(0.2, 19.8)**	**0.04**	**10.2**	**(0.6, 19.8)**	**0.04**
lmDLN	55.0	8.2	6.2				0.3	(-1.3, 1.8)	0.72
coDLN	55.2	8.7	6.5						

[a] paired t-test, [b] 95% margin of error

Both DLNs achieved significantly higher precision (27.0% with lmDLN and 26.8% with coDLN) than logistic regression (22.2%). There was no difference in precision between

lmDLN and coDLN. Logistic regression performed worse than random class assignment based on progression prevalence (23.7%) whereas the lesion mask and combined deep learning networks provided an improvement in positive pre- to post-test probability of 3.3% and 3.1%, respectively. Table 4-2 summarizes these findings.

Table 4-2 Summary of validation precision and change from pre- to post-positive predictive value of logistic regression using only user-defined features (LR), lesion mask only deep learning network (lmDLN), and the combined user-defined and lesion mask features deep learning network (coDLN) when EDSS was treated as a continuous variable

Ref. Model	**% Precision** n = 10			**Mean % Precision Difference [a]** n = 10, *df* = 9						**Mean % ΔPPV[c]** n = 10
				lmDLN – Ref.			**coDLN – Ref.**			
	Mean	**SD**	**Error[b]**	Mean	95% CI	*P*	Mean	95% CI	*P*	
LR	22.2	4.9	3.7	**4.8**	**(0.8, 8.9)**	**0.02**	**4.6**	**(0.4, 8.8)**	**0.03**	-1.5
lmDLN	27.0	3.8	2.9				-0.2	(-2.0, 1.7)	0.82	3.3*
coDLN	26.8	3.1	2.3							3.1*

[a] paired t-test, [b] 95% margin of error, [c] compared to progression prevalence of 23.7%
*statistically significant ΔPPV ($P < 0.05$)

Although both lmDLN and coDLN had higher sensitivity than LR, on average identifying 54.8% and 53.0% of progressors in test sets compared to the 45.1% of progressors identified by LR, the differences were not significant. A summary of sensitivities is shown in Table 4-3.

Table 4-3 Summary of sensitivity of logistic regression using only user-defined features (LR), lesion mask only deep learning network (lmDLN), and the combined user-defined and lesion mask features deep learning network (coDLN) when EDSS was treated as a continuous variable

Ref. Model	% Sensitivity n = 10			Mean % Sensitivity Difference [a] n = 10, *df* = 9					
				lmDLN – Ref.			coDLN – Ref.		
	Mean	**SD**	**Error[b]**	Mean	95% CI	*P*	Mean	95% CI	*P*
LR	45.1	16.1	12.1	9.8	(-1.2, 20.8)	0.08	8.0	(-3.4, 19.4)	0.15
lmDLN	54.8	10.3	7.7				-1.8	(-6.6, 3.0)	0.42
coDLN	53.0	8.1	6.1						

[a] paired t-test, [b] 95% margin of error

Both networks had greater mean NPV over LR, but only the lesion mask deep learning network significantly outperformed logistic regression in classifying non-progressors as measured by the negative predictive value, achieving an NPV of 79.1% (4.2% better than LR). coDLN achieved an improvement of NPV over non-progression prevalence (negative pre- to post-test probability) of ΔNPV=2.4%. The addition of user-defined predictors in coDLN did not result in any NPV changes. Findings of NPV are found in Table 4-4.

Table 4-4 Summary of negative predictive value and change from pre- to post-negative predictive value of logistic regression using only user-defined features (LR), lesion mask only deep learning network (lmDLN), and the combined user-defined and lesion mask features deep learning network (coDLN) when EDSS was treated as a continuous variable

Ref. Model	% NPV n = 10			Mean % NPV Difference [a] n = 10, *df* = 9						Mean % ΔNPV[c] n = 10
				lmDLN – Ref.			coDLN – Ref.			
	Mean	**SD**	**Error[b]**	Mean	95% CI	*P*	Mean	95% CI	*P*	
LR	74.9	5.2	3.9	**4.2**	**(0.5, 7.8)**	**0.03**	3.7	(-0.1, 7.5)	0.05	-1.4
lmDLN	79.1	4.6	3.5				-0.4	(-2.7, 1.8)	0.66	2.8
coDLN	78.7	3.3	2.5							2.4*

[a] paired t-test, [b] 95% margin of error, [c] compared to non-progression prevalence of 76.3%
*statistically significant ΔNPV ($P < 0.05$)

There were no significant differences in model specificity, with logistic regression detecting 51.3% of non-progressors, while lmDLN and coDLN identified 53.5% and 54.3%, respectively. DLN-learned lesion mask features, with or without user-defined features, did not improve the identification rate of non-progressors. These findings can be found in Table 4-5.

Table 4-5 Summary of specificity of logistic regression using only user-defined features (LR), lesion mask only deep learning network (lmDLN), and the combined user-defined and lesion mask features deep learning network (coDLN) when EDSS was treated as a continuous variable

Ref. Model	% Specificity n = 10			Mean % Specificity Difference [a] n = 10, *df* = 9					
				lmDLN – Ref.			coDLN – Ref.		
	Mean	SD	Error[b]	Mean	95% CI	*P*	Mean	95% CI	*P*
LR	51.3	13.4	10.1	2.2	(-6.0, 10.4)	0.57	3.0	(-6.0, 12.0)	0.47
lmDLN	53.5	8.9	6.7				0.8	(-1.6, 3.2)	0.47
coDLN	54.3	9.7	7.3						

[a] paired t-test, [b] 95% margin of error

4.5.2 EDSS as a Categorical Variable

No differences were observed in model AUC when EDSS was treated as a categorical variable as opposed to a continuous variable. These findings are summarized in Table 4-6. Both DLNs were more stable with respect to AUC performance, with 95% margins of error of 6.1% and 6.2% respectively compared to LR with a 9.8% margin of 95% error.

Table 4-6 Summary of area under the curve validation performance for logistic regression using only user-defined features (LR), lesion mask only deep learning network (lmDLN), and the combined user-defined and lesion mask features deep learning network (coDLN) when EDSS was treated as a categorical variable

Ref. Model	% AUC n = 10			Mean % AUC Difference [a] n = 10, *df* = 9					
				lmDLN – Ref.			coDLN – Ref.		
	Mean	SD	Error[b]	Mean	95% CI	*P*	Mean	95% CI	*P*
LR	59.9	13.0	9.8	-5.0	(-16.6, 6.6)	0.35	-4.7	(-16.1, 6.8)	0.38
lmDLN	54.9	8.2	6.1				0.4	(-0.7, 1.4)	0.45
coDLN	55.3	8.2	6.2						

[a] paired t-test, [b] 95% margin of error

Similar model stability was observed in DLN precision performance (Table 4-7) where lmDLN and coDLN had tighter 95% margins of error of 2.9% and 2.7% compared to LR with a margin of 8.1%. Both DLNs outperformed prevalence-based random progression prediction by 3.3% and 3.0% respectively despite no significant differences observed between LR and DLN precision.

Table 4-7 Summary of validation precision and change from pre- to post-positive predictive value of logistic regression using only user-defined features (LR), lesion mask only deep learning network (lmDLN), and the combined user-defined and lesion mask features deep learning network (coDLN) when EDSS was treated as a categorical variable

Ref. Model	% Precision n = 10			Mean % Precision Difference [a] n = 10, *df* = 9						Mean % ΔPPV[c] n = 10
				lmDLN – Ref.			coDLN – Ref.			
	Mean	**SD**	**Error[b]**	Mean	95% CI	*P*	Mean	95% CI	*P*	
LR	29.1	10.8	8.1	-2.1	(-10.1, 6.0)	0.49	-2.4	(-10.3, 5.5)	0.51	5.3
lmDLN	27.0	3.8	2.9				-0.3	(-2.3, 1.6)	0.72	3.3*
coDLN	26.7	3.6	2.7							3.0*

[a] paired t-test, [b] 95% margin of error, [c] compared to progression prevalence of 23.7%
*statistically significant ΔPPV ($P < 0.05$)

No significant differences were observed in classifier sensitivity between LR, lmDLN and coDLN. These findings are summarized in Table 4-8. Compared to LR with a 18.5% margin of error, both lmDLN and coDLN had smaller margins of 7.7% and 6.5% respectively.

Table 4-8 Summary of sensitivity of logistic regression using only user-defined features (LR), lesion mask only deep learning network (lmDLN), and the combined user-defined and lesion mask features deep learning network (coDLN) when EDSS was treated as a categorical variable

Ref. Model	% Sensitivity n = 10			Mean % Sensitivity Difference [a] n = 10, *df* = 9					
				lmDLN – Ref.			coDLN – Ref.		
	Mean	**SD**	**Error[b]**	Mean	95% CI	*P*	Mean	95% CI	*P*
LR	58.0	24.6	18.5	-3.1	(-20.7, 14.5)	0.70	-5.7	(-22.6, 11.3)	0.47
lmDLN	54.8	10.3	7.7				-2.6	(-6.7, 1.5)	0.19
coDLN	52.3	8.6	6.5						

[a] paired t-test, [b] 95% margin of error

No significant differences were observed in negative predictive values of the three classifiers. Only LR achieved a significant ΔNPV of 6.1%. NPV and ΔNPV results are summarized in Table 4-9.

Table 4-9 Summary of negative predictive value and change from pre- to post-negative predictive value of logistic regression using only user-defined features (LR), lesion mask only deep learning network (lmDLN), and the combined user-defined and lesion mask features deep learning network (coDLN) when EDSS was treated as a categorical variable

Ref. Model	% NPV n = 10			Mean % NPV Difference [a] n = 10, *df* = 9						Mean % ΔNPV[c] n = 10
				lmDLN – Ref.			coDLN – Ref.			
	Mean	**SD**	**Error[b]**	Mean	95% CI	*P*	Mean	95% CI	*P*	
LR	82.4	7.5	5.6	-3.3	(-9.5, 2.9)	0.26	-3.9	(-9.2, 1.4)	0.13	6.1*
lmDLN	79.1	4.6	3.5				-0.6	(-2.5, 1.3)	0.49	2.8
coDLN	78.5	3.3	2.5							2.2

[a] paired t-test, [b] 95% margin of error, [c] compared to non-progression prevalence of 76.3%
*statistically significant ΔNPV ($P < 0.05$)

No significant differences were observed in classifier specificities between LR, lmDLN, and coDLN. Findings are summarized in Table 4-10.

Table 4-10 Summary of specificity of logistic regression using only user-defined features (LR), lesion mask only deep learning network (lmDLN), and the combined user-defined and lesion mask features deep learning network (coDLN) when EDSS was treated as a categorical variable

Ref. Model	% Specificity n = 10			Mean % Specificity Difference [a] n = 10, *df* = 9					
				lmDLN – Ref.			coDLN – Ref.		
	Mean	**SD**	**Error[b]**	Mean	95% CI	*P*	Mean	95% CI	*P*
LR	57.8	9.8	7.4	-4.3	(-13.0, 4.3)	0.29	-3.2	(-13.1, 6.6)	0.48
lmDLN	53.5	8.9	6.7				1.1	(-2.4, 4.5)	0.49
coDLN	54.6	9.8	7.4						

[a] paired t-test, [b] 95% margin of error

Chapter 5: Discussion & Conclusion

In most studies of prognostic factors for disability progression, predictive models use statistical approaches such as linear regression for continuous response prediction or logistic regression for binary response prediction [47] and Cox regression or Kaplan-Meier analyses for survival analysis [48]. These analyses do not provide any estimation of their generalizability on samples not used for model fitting. For example, logistic regression was used to evaluate brain atrophy and lesion load as prognostic factors for predicting EDSS score at 10 years [49]. R^2 values were reported for model goodness of fit to the data, but no estimation of how the model would perform on data not used for model fitting was provided. Our study evaluated model performance based on their estimated generalizability by validating models on data withheld from training in each cycle of 10CV.

5.1 Predicting SPMS Disability Progression with Machine Learning and User-defined Features

5.1.1 Treating EDSS as a Continuous Variable

In our study population of 485 SPMS participants, we found that RF and AdBDT outperformed the naïve, black-box implementation of logistic regression typically seen in data science in separating CDP+ from CDP- (AUC), CDP+ predictive accuracy (PPV), and CDP- predictive accuracy (NPV) only when EDSS was analyzed as a continuous variable. In fact, when continuous EDSS was used, on average, the black-box implementation of logistic regression identified less than half of progressors and non-progressors in our study population.

We observed that using an ensemble of linear SVMs, there was no significant difference in performance compared to logistic regression. These findings were in line with those by Zhao et al. when using only baseline features [25]. This may be due to the limitations of its linearity as there was no evidence of improvement over prevalence-based random CDP+ or CDP- prediction. On the other hand, random forest and the AdaBoost ensemble of simple decision trees were not restricted to linear relationships and outperformed logistic regression and linear support vector machines in predictive accuracies PPV and NPV. Performance between random forest and AdBDT was comparable, with no statistically significant difference between AdBDT and RF performance. Both non-linear machine learning methods increased the accuracy of predicting progression over prevalence-based random prediction while only AdaBoost resulted in a significant ΔNPV.

Despite improvements in PPV and NPV demonstrated by RF and AdaBoost, no statistically significant improvements were observed in their sensitivity and specificity measures over enSVM and LR. This may be due to the relatively small validation sets (approximately 48 samples per validation dataset) generated by 10-CV.

Logistic regression continues to be the standard approach in modeling binary disability progression in multiple sclerosis, evaluated based on goodness of fit and not on generalizability. However, our findings suggest that the linear assumption for modeling disability progression in SPMS and black-box implementations of LR in data science should be questioned. As we have shown, non-linear classification models outperformed the black-box implementations of linear models.

Analyzing predictor contributions to each of the models, we can see that both linear models heavily depended on baseline EDSS on predicting progression. In contrast, T25W

contributed the least. This led us to hypothesize that there may be a linear relationship present between continuous EDSS and progression which is lacking with T25W. However, both linear models performed worse than the non-linear methods which were able to make use of the information provided by T25W. Additionally, we found that sex as a predictor had a near-zero contribution on non-linear models, which suggests that it may potentially have no value for predicting progression in SPMS. We observed sex to be used more generously in logistic regression and enSVM which once again may solely be due to the existence of a linear relationship. Ultimately, these linear relationships were inadequate in optimizing the linear models for prediction of CDP.

5.1.2 Treating EDSS as a Categorical Variable

When EDSS was treated as a categorical variable, performance increases were more notable in LR and enSVM. enSVM achieved a significantly higher AUC than LR, RF, and AdBDT. The increased AUC of RF was not as much as the linear classifiers, while AdBDT had a slightly lower AUC compared to using continuous EDSS. Although there were no significant differences in classifier precision when using categorical EDSS and all classifiers performed better than a prevalence-based random prediction, the pre- to post-positive predictive values of LR and enSVM were greater than RF and AdBDT. Additionally, while LR and enSVM NPV were outperformed by RF and AdBDT when using continuous EDSS, these differences were eliminated when EDSS was analyzed as a categorical variable. No significant improvement was observed in LR and enSVM ΔNPV with continuous EDSS, but categorical EDSS resulted in improvements in both of these classifiers.

Although continuous EDSS saw EDSS contributing the greatest to model training for LR and enSVM and it was hypothesized that it had the strongest linear relationship of all

user-defined features, the treatment of EDSS as a categorical variable resulted in a much greater gap between EDSS contribution and that of the other user-defined features. The dependency on EDSS by both linear classifiers was much greater with categorical EDSS (Figure 3-3) than it was with continuous EDSS, demonstrating the sensitivity of linear classifiers on pre-processing of input data.

Unlike the linear parametric classifiers, the non-parametric classifiers were less affected by how EDSS was treated, with comparable performance metrics between analyzing EDSS as a categorical or continuous variable. Qualitative analysis of predictor contributions between using continuous EDSS and categorical EDSS showed similar patterns. RF and AdBDT both relied on a set of decision rules for constructing decision boundaries and were less affected by how variables are treated. This allows them to be more robust than LR and enSVM where domain knowledge is important in correctly analyzing input data.

5.2 Deep learning brain lesion masks for predicting SPMS disability progression

5.2.1 Treating EDSS as a Continuous Variable

A basic deep learning network for automated extraction of lesion distribution features from binary lesion masks was able to improve distinguishability of progressors and non-progressors by approximately 10% based on area under the receiver-operator characteristic curve, and detection of progressors (PPV) and non-progressors (NPV) by 4.8 and 4.2% respectively compared to logistic regression. While there were no additional improvements by adding user-defined demographic with continuous EDSS, clinical and MRI features with

the deep-learned lesion mask features with respect to AUC, PPV, NPV, sensitivity, and specificity, these features improved the positive and negative post-test probabilities (ΔPPV and ΔNPV) by reducing variance in predictions.

The improvements in PPV and NPV over the naïve multivariate logistic regression of user-defined features when using deep-learned features from binary lesion masks may be due to its ability to consider spatial information in addition to volumetric information from the masks. In conventional MRI metrics such as BPF and T2LV, spatial information is lost. Additionally, as disability monitored by EDSS is weighted towards physical disabilities, it is likely that the DLN placed heavier weighting on lesions located in regions of the brain that affect mobility – a hypothesis which would require further testing. Deep learning has previously been used by Yoo et al. for predicting conversion from CIS to MS using deep-learned features from brain lesion masks and was also shown to outperform multivariate logistic regression [31].

5.2.2 Treating EDSS as a Categorical Variable

Benefits of both DLNs on this dataset over naïve logistic regression was lost when categorical EDSS was used in the user-defined features mainly due to the improved performance in LR, similar to the changes discussed in Section 5.1.2 when EDSS was analyzed as a categorical variable with ML. Although no significant difference in performance was observed between LR, lmDLN, and coDLN, the lesion mask DLN was able to use solely features from transformed binary lesion masks to match LR predictive performance using user-defined features. Both lmDLN and coDLN were also more stable in performance as they had tighter 95% error margins in AUC, PPV, and sensitivity. The

stability of the DLNs also enabled them to have statistically significant improvements in ΔPPV.

Although LR performance increased when using categorical EDSS, the improvements did not translate to the use of categorical EDSS in coDLN. We hypothesize that this may be due to the ratio between lesion mask features and EDSS (256:1) entering the logistic regression layer of coDLN. It is likely that the improvements due to categorical EDSS are trumped by the number of lesion mask features entering the logistic regression layer. Additionally, as lmDLN performed as well as LR, in conjunction with a small sample size (discussed later), it is possible that there was not enough variance in the data for additional relationships between lesion mask features and user-defined features to be learned.

5.3 Challenges and Limitations

While the models developed from this study provide an improvement in performance over the conventional black-box implementation of the logistic regression model and prevalence-based baseline performance when continuous EDSS was used, additional work has to be done. A definition of progression defined by an increase in EDSS is weighted towards physical disabilities and mobility issues. Using a broader or more comprehensive definition of progression that includes changes in cognition as well as mobility may provide improved prediction results.

Our sample of 485 is considered small for machine learning and deep learning purposes and demonstrates a difficulty in training machine learning models – the need for large amounts of data. Only 23.7% of the study population (115 participants) were progressors. This sample size is unlikely to fully capture the variation of lesion distributions

or user-defined features for modelling with either logistic regression or deep learning and is likely the main contributor to the observed trend in higher NPVs than PPVs. We hypothesize that in a larger dataset, the improvements in PPV and NPV would be better reflected in model sensitivity and specificity. This may also contribute to the increased precision and negative predictive value of deep learning not being reflected in sensitivity and specificity, as test sets in each 10CV fold had approximately only 49 participants. As discussed in Section 5.1.2, while there were minimal differences between LR, enSVM, RF, and AdBDT when using categorical EDSS, both RF and AdBDT made use of more user-defined features than LR and enSVM. With a larger sample size, RF and AdBDT may be able to outperform LR and enSVM by better learning relationships within non-EDSS predictors whose variance, necessary to represent the population, was unfortunately not captured in the limited data set used in our experiments.

In addition to the limited sample size, we also only used baseline data for prediction and a basic method for integrating user-defined features. The inclusion of longitudinal data, both user-defined features as well as lesion masks, may provide important information on the rate of change that could add predictive value.

With respect to user-defined features, only a small set of predictors were used in our experiments. The improvement in performance using non-parametric models may be amplified by the inclusion of additional predictors whose relationships with progression may be better captured using non-linear or non-parametric methods. Other methods of joint modelling may improve the results of combining automatically learned features with user-defined features.

Finally, the generalizability of these results is limited to identifying short-term progression. The non-progressors may show evidence of disability progression after the 2-year study window.

5.4 Concluding Statements & Future Work

Existing research on AI applications in MS have mostly been focused on classification and disease state transitions. Our work is one of many steps required to develop a clinically-usable prognostic tool. Even in its current form, its improvement over a prevalence-based classification scheme and logistic regression may aid in streamlining clinical trial recruitment and suggests that non-linear modeling may be better suited for evaluating the prognostic value of factors of progression.

In the design of clinical trials and statistical testing, balanced designs are preferred over unbalanced design when possible. Balanced designs results in tests with greater statistical power as it gives the maximal information regarding treatment differences [50]. In [51], it was shown that unbalanced randomized control trials (RCT) results often favor new treatments when compared to balanced trials. While control/treatment groups can be balanced, unforeseen group imbalances may arise over the duration of the trial. The ideal RCT should consider time-dependent changes (i.e. progression) in the cohort and reduce potential group imbalances. The identification of those most at risk of disability progression during a trial and most likely to benefit from treatment would improve the efficiency of the trial and the power associated with treatment effect findings.

Machine learning applications in Alzheimer's disease for clinical trial enrichment and design have been shown to enable smaller trials with high statistical power by selecting

participants at higher risk of cognitive decline [52, 53]. Based on our results, the use of the AdaBoost model would hypothetically reduce the imbalance between progressors and non-progressors by identifying five more progressors and five fewer non-progressors in every 100 individuals screened for study eligibility, regardless of whether EDSS was analyzed as a continuous or categorical variable. The incorporation of predictive machine learning models into SPMS clinical trial design may allow those at highest risk of disease worsening to access experimental therapies and yield treatment findings with acceptable statistical power using a smaller study cohort.

Deep learning was able to extract self-taught lesion distribution features from binary lesion masks. A deep learning network using only the binary lesion masks was superior to logistic regression of user-defined features when continuous EDSS was used for predicting short-term confirmed disability progression in our cohort of SPMS. When categorical EDSS was used, the same DLN using only brain lesion masks performed as well as naïve logistic regression. Regardless of how EDSS was analyzed, the use of lesion mask features led to more stable performance.

From our experiments, we showed that machine learning is more robust to data processing methods. Unlike the simple ML models such as LR and enSVM, non-parametric RF and AdaBoost-DT performance was robust to changes in how EDSS was processed. Non-parametric models appear to be less sensitive to data processing methods, making them more suitable for applications where the proper treatment of input features is unclear. Feature importance in RF and AdBDT were also more resilient to changes in how EDSS was processed.

Future work would look at increasing sample size (particularly that of progressors), including longitudinal lesion mask data and user-defined features, experimenting with different definitions of progression, using different DL network architectures, and validating the models on an independent dataset. The visualization of automatically learned features may also provide additional insight into MS pathology and pathogenesis.

Bibliography

[1] C. Barillot, G. Edan, and O. Commowick, “Imaging biomarkers in multiple Sclerosis: From image analysis to population imaging,” *Med. Image Anal.*, vol. 33, pp. 134–139, Oct. 2016.

[2] F. D. Lublin *et al.*, “Defining the clinical course of multiple sclerosis: The 2013 revisions,” *Neurology*, vol. 83, no. 3, pp. 278–286, Jul. 2014.

[3] F. D. Lublin and S. C. Reingold, “Defining the clinical course of multiple sclerosis: Results of an international survey,” *Neurology*, vol. 46, no. 4, pp. 907–911, Apr. 1996.

[4] J. F. Kurtzke, “Rating neurologic impairment in multiple sclerosis: an expanded disability status scale (EDSS).,” *Neurology*, vol. 33, no. 11, pp. 1444–52, Nov. 1983.

[5] J. S. Fischer, R. A. Rudick, G. R. Cutter, and S. C. Reingold, “The Multiple Sclerosis Functional Composite Measure (MSFC): an integrated approach to MS clinical outcome assessment. National MS Society Clinical Outcomes Assessment Task Force.,” *Mult. Scler.*, vol. 5, no. 4, pp. 244–50, Aug. 1999.

[6] B. Hurwitz, “The diagnosis of multiple sclerosis and the clinical subtypes,” *Ann. Indian Acad. Neurol.*, vol. 12, no. 4, p. 226, 2009.

[7] M. Filippi and F. Agosta, “Imaging biomarkers in multiple sclerosis,” *J. Magn. Reson. Imaging*, vol. 31, no. 4, pp. 770–788, Apr. 2010.

[8] J. Bell, *Machine Learning*. Indianapolis, IN, USA: John Wiley & Sons, Inc, 2014.

[9] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[10] D. W. Hosmer Jr., S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2013.

[11] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[12] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*, 1st ed. Boca Raton: Chapman & Hall/CRC, 1984.

[13] L. Breiman, “Random Forests,” *Mach. Learn.*, vol. 45, no. 5, 2001.

[14] Y. Freund and R. E. Schapire, “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,” *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[15] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” Dec. 2014.

[16] D. J. Felleman and D. C. Van Essen, “Distributed Hierarchical Processing in the Primate Cerebral Cortex,” *Cereb. Cortex*, vol. 1, no. 1, pp. 1–47, Jan. 1991.

[17] K. Sakai and K. Yamada, “Machine learning studies on major brain diseases: 5-year trends of 2014–2018,” *Jpn. J. Radiol.*, Nov. 2018.

[18] A. Ion-Mărgineanu *et al.*, “Machine Learning Approach for Classifying Multiple Sclerosis Courses by Combining Clinical Data with Lesion Loads and Magnetic Resonance Metabolic Features,” *Front. Neurosci.*, vol. 11, Jul. 2017.

[19] M. Zurita *et al.*, “Characterization of relapsing-remitting multiple sclerosis patients using support vector machine classifications of functional and diffusion MRI data,” *NeuroImage Clin.*, vol. 20, pp. 724–730, 2018.

[20] J. Zhong, D. Q. Chen, J. C. Nantes, S. A. Holmes, M. Hodaie, and L. Koski, “Combined structural and functional patterns discriminating upper limb motor disability in multiple sclerosis using multivariate approaches,” *Brain Imaging Behav.*, vol. 11, no. 3, pp. 754–768, Jun. 2017.

[21] J. J. Cerqueira *et al.*, “Time matters in multiple sclerosis: can early treatment and long-term follow-up ensure everyone benefits from the latest advances in multiple sclerosis?,” *J. Neurol. Neurosurg. Psychiatry*, vol. 89, no. 8, pp. 844–850, Aug. 2018.

[22] V. Wottschel *et al.*, “Predicting outcome in clinically isolated syndrome using machine learning,” *NeuroImage Clin.*, vol. 7, pp. 281–287, 2015.

[23] H. Zhang *et al.*, “Predicting conversion from clinically isolated syndrome to multiple sclerosis–An imaging-based machine learning approach,” *NeuroImage Clin.*, Nov. 2018.

[24] K. Bendfeldt *et al.*, “MRI-based prediction of conversion from clinically isolated syndrome to clinically definite multiple sclerosis using SVM and lesion geometry,” *Brain Imaging Behav.*, Aug. 2018.

[25] Y. Zhao *et al.*, “Exploration of machine learning techniques in predicting multiple sclerosis disease course,” *PLoS One*, vol. 12, no. 4, p. e0174866, Apr. 2017.

[26] T. Brosch, Y. Yoo, D. K. B. Li, A. Traboulsee, and R. Tam, “Modeling the variability in brain morphology and lesion distribution in multiple sclerosis by deep learning,” *Med Image Comput Comput Assist Interv*, vol. 17, no. 2, p. 462, 2014.

[27] E. M. Sweeney *et al.*, “A Comparison of Supervised Machine Learning Algorithms and Feature Vectors for MS Lesion Segmentation Using Multimodal Structural MRI,” *PLoS One*, vol. 9, no. 4, p. e95753, Apr. 2014.

[28] T. Brosch, L. Y. W. Tang, Y. Yoo, D. K. B. Li, A. Traboulsee, and R. Tam, “Deep 3D Convolutional Encoder Networks With Shortcuts for Multiscale Feature Integration Applied to Multiple Sclerosis Lesion Segmentation,” *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1229–1239, May 2016.

[29] Y. Yoo *et al.*, “Deep learning of joint myelin and T1w MRI features in normal-appearing

brain tissue to distinguish between multiple sclerosis patients and healthy controls," *NeuroImage Clin.*, vol. 17, pp. 169–178, 2018.

[30] Y. Yoo *et al.*, "Hierarchical Multimodal Fusion of Deep-Learned Lesion and Tissue Integrity Features in Brain MRIs for Distinguishing Neuromyelitis Optica from Multiple Sclerosis," 2017, pp. 480–488.

[31] Y. Yoo *et al.*, "Deep learning of brain lesion patterns and user-defined clinical and MRI features for predicting conversion to multiple sclerosis from clinically isolated syndrome," *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.*, pp. 1–10, Aug. 2017.

[32] M. S. Freedman *et al.*, "A phase III study evaluating the efficacy and safety of MBP8298 in secondary progressive MS.," *Neurology*, vol. 77, no. 16, pp. 1551–60, Oct. 2011.

[33] J. McAusland, R. C. Tam, E. Wong, A. Riddehough, and D. K. B. Li, "Optimizing the Use of Radiologist Seed Points for Improved Multiple Sclerosis Lesion Segmentation," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 11, pp. 2689–2698, Nov. 2010.

[34] C. Jones, D. K. Li, G. Zhao, D. W. Paty, and P. S. Group, "Atrophy Measurements in Multiple Sclerosis," in *Proc. Intl. Soc. Mag. Reson. Med 9*, 2001.

[35] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," Jan. 2012.

[36] W. McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference*, 2010, pp. 51–56.

[37] E. Jones, T. Oliphant, and P. Peterson, "SciPy: Open Source Scientific Tools for Python," 2001. [Online]. Available: http://www.scipy.org/. [Accessed: 01-Jan-2019].

[38] F. Chollet, "Keras," 2015. [Online]. Available: https://keras.io.

[39] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems." 2015.

[40] M. Jenkinson and S. Smith, “A global optimisation method for robust affine registration of brain images.,” *Med. Image Anal.*, vol. 5, no. 2, pp. 143–56, Jun. 2001.

[41] M. Jenkinson, P. Bannister, M. Brady, and S. Smith, “Improved optimization for the robust and accurate linear registration and motion correction of brain images.,” *Neuroimage*, vol. 17, no. 2, pp. 825–41, Oct. 2002.

[42] C. R. Maurer, Rensheng Qi, and V. Raghavan, “A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 2, pp. 265–270, Feb. 2003.

[43] P. A. Yushkevich *et al.*, “User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability,” *Neuroimage*, vol. 31, no. 3, pp. 1116–1128, Jul. 2006.

[44] B. Xu, N. Wang, T. Chen, and M. Li, “Empirical Evaluation of Rectified Activations in Convolutional Network,” May 2015.

[45] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[46] K. He, X. Zhang, S. Ren, and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,” Feb. 2015.

[47] G. Tripepi, K. J. Jager, F. W. Dekker, and C. Zoccali, “Linear and logistic regression analysis,” *Kidney Int.*, vol. 73, no. 7, pp. 806–810, Apr. 2008.

[48] V. Bewick, L. Cheek, and J. Ball, “Statistics review 12: survival analysis.,” *Crit. Care*, vol. 8, no. 5, pp. 389–94, Oct. 2004.

[49] V. Popescu *et al.*, “Brain atrophy and lesion load predict long term disability in multiple

sclerosis," *J. Neurol. Neurosurg. Psychiatry*, vol. 84, no. 10, pp. 1082–1091, Oct. 2013.

[50] D. A. Berry, "Sequential Statistical Methods," in *International Encyclopedia of the Social & Behavioral Sciences*, 2nd ed., Elsevier, 2015, pp. 634–638.

[51] C. Dibao-Dina, A. Caille, and B. Giraudeau, "Unbalanced rather than balanced randomized controlled trials are more often positive in favor of the new treatment: an exposed and nonexposed study," *J. Clin. Epidemiol.*, vol. 68, no. 8, pp. 944–949, Aug. 2015.

[52] V. K. Ithapu, V. Singh, O. C. Okonkwo, R. J. Chappell, N. M. Dowling, and S. C. Johnson, "Imaging-based enrichment criteria using deep learning algorithms for efficient clinical trials in mild cognitive impairment," *Alzheimer's Dement.*, vol. 11, no. 12, pp. 1489–1499, Dec. 2015.

[53] V. K. Ithapu, V. Singh, and S. C. Johnson, "Randomized Deep Learning Methods for Clinical Trial Enrichment and Design in Alzheimer's Disease," in *Deep Learning for Medical Image Analysis*, Elsevier, 2017, pp. 341–378.