PREDICTING TRAVEL MODE CHOICE WITH 86 MACHINE LEARNING CLASSIFIERS: AN EMPIRICAL BENCHMARK STUDY

Shenhao Wang (Corresponding Author)
Department of Urban Studies and Planning
Massachusetts Institute of Technology
77 Massachusetts Avenue, Cambridge, MA 02139, USA.
Tel: 617-335-7764, email: shenhao@mit.edu

Baichuan Mo
Department of Civil and Environmental Engineering
Massachusetts Institute of Technology
77 Massachusetts Avenue, Cambridge, MA 02139, USA.
Tel: 857-999-5906, email: baichuan@mit.edu

Jinhua Zhao
Department of Urban Studies and Planning
Massachusetts Institute of Technology
77 Massachusetts Avenue, Cambridge, MA 02139, USA.
Tel: 617-324-7594, email: jinhua@mit.edu

Word Count: 4175 words + 3 figures$\times$ 0 + 2 tables$\times$ 250 = 4675 words

Submission Date: Tuesday 30th July, 2019

ABSTRACT

Researchers are applying a large number of machine learning (ML) classifiers to predict travel behavior, but the results are data-specific and the selection of ML classifiers is author-specific. To obtain generalizable results, this paper provides an empirical benchmark by using 86 classifiers from 14 model families to predict the travel mode choice based on the National Household Travel Survey (NHTS) 2017 dataset. The 86 ML classifiers from 14 model families incorporate all the important ML classifiers discussed in previous studies. The large number of observations (about $800,000$) in the NHTS2017 dataset enables us to analyze the effect of different sample sizes as a meta-dimension on prediction accuracy. We found that **ensemble models**, including boosting, bagging, and random forests, perform the best among all the classifiers, and that **deep neural networks** (DNNs) perform the best among all the non-ensemble models. Classical **discrete choice models** (DCMs) only predict at the medium or relatively low range of prediction accuracy among all the models. Particularly, mixed logit model cannot be trained in a reasonable amount of time owing to its computational difficulty in sampling. Larger sample size generally leads to higher prediction accuracy, particularly for the models with high model complexity. Overall, this study provides an empirical benchmark result for the future, and future studies can build upon our results by testing more ML classifiers on the same NHTS2017 dataset, thus yielding more comparable, replicable, and generalizable knowledge shared by the whole research community.

Keywords: Machine Learning, Travel Behavior

1. INTRODUCTION

In the transportation field, travel demand prediction functions as the foundation of transportation system optimization, economic analysis, and discussions about congestion mitigation policies. Travel demand includes the choice of trip purposes, trip modes, travel frequency, travel scheduling, destination and origin, travel route, long-term and short-term activity, locations, car ownership, and many other travel-related behaviors (*1–6*). Whereas demand forecasting is traditionally addressed by using discrete choice models (DCMs), including multinomial logit (MNL) model, nested logit (NL) model, and mixed logit (MXL) model (*7*), researchers can actually choose from a long list of machine learning (ML) classifiers for prediction because many travel behavioral decisions can be represented by discrete variables (*8–10*). In the previous studies that focus on the performance of ML classifiers on predicting travel demand, a typical procedure is to compare DCMs to one or several ML classifiers and to select the best one based on the comparison of prediction accuracy. However, the selection of the alternative ML classifiers is limited because it is often based on researchers' expertise. Also the results are data-specific depending on the geographical locations where the datasets were collected and limited by the sample size the researcher has. These author-specific and data-specific limitations need to be overcome so that the research community can know the generally best classifier for travel demand prediction.

This study seeks to find the best classifier with the highest possible prediction accuracy for travel mode choice prediction, by comparing 86 *classifiers from* 14 *model families based on the National Household Travel Survey 2017 (NHTS 2017) dataset.* The travel mode choice is the focus because it is the classical question in choice modeling. The 86 classifiers are chosen from 14 of the most important classifier families, summarized from the review of the previous studies, including (1) discrete choice models (DCMs; 3 models), (2) deep neural networks (DNNs; 16 models), (3) discriminant analysis (DA; 12 models), (4) Bayesian methods (BM; 6 models), (5) support vector machines (SVM; 7 models), (6) K nearest neighbors (KNN; 4 models), (7) decision trees (DT; 12 models), (8) generalized linear models (GLM; 10 models), (9) Gaussian process (GP; 3 models), (10) rule-based models (RBM; 3 models); (11) bagging (BAGGING; 3 models), (12) random forests (RF; 2 models), (13) boosting (BOOSTING; 3 models), and (14) others (OTHERS; 2 models). While it is impossible to exhaust all the available ML classifiers, our list of classifiers is designed to represent the most important ones and cover all the methods used in the past studies concerning travel behavior prediction. The NHTS dataset is used because the dataset covers the whole United States and the sample size is large enough to test the effects of different sample sizes by resampling from the full dataset. Readers can treat each single model as one "data point" in our study, and our analysis largely expands along two meta-dimensions: different classifiers and sample sizes, as opposed to previous studies that only examine one or several "data points". Overall, our study seeks to (1) find the globally best classifier for the prediction of travel mode choice; (2) rank the importance of each model family and classifier in a robust way; and (3) provide insights into particularly important model families, such as DNNs and DCMs.

This paper serves as an empirical benchmark for any future study that seeks to predict travel behavior, particularly when the geographical location of the dataset is within the U.S. For example, future researchers could use the recommended classifier in a specific context for travel demand prediction without involving another large-scale comparison of ML classifiers. This study also provides intellectual insights into the characteristics of travel behaviors based on the performance of the classifiers. The ensemble classifiers, as shown to be the globally best, can capture the heterogeneity of travel behavior better than each individual classifier, revealing that the behavioral heterogeneity exists not only at the individual level but also at the model level. Last but not least, we suggest future studies, particularly those focus on modeling methods, to use standard datasets (e.g. NHTS 2017 dataset) to test the performance of models, so that the knowledge gleaned from individual researchers can become replicable, generalizable, and comparable across the whole research community.

The next section reviews the papers that compared classifiers for travel behavior prediction. Section 3 discusses our choice of classifiers and the NHTS dataset. Section 4 shows the performance of the ML classifiers and discusses specific model families such as DCMs. Section 5 concludes our findings.

2. LITERATURE REVIEW

Table 1 summarizes 15 past studies that focused on predicting travel mode choice. The 15 studies are by no means exhaustive of all the relevant studies, but suffice to provide valuable information for the setup of our experiments. For example, DCMs (including MNL and NL) are the dominant classifiers used in these studies for comparison: any study that involves comparison of several ML classifiers uses DCMs as the benchmark classifier. This is not a surprise given the historically important role DCMs play in the field of demand analysis (*7, 11*). Besides DCMs, DNNs are the second most frequently used: 9 out of the 15 studies used DNNs. Other than DCMs and DNNs, researchers also used SVM, DT, BOOSTING, BAGGING, RF, and other classifiers to model travel mode choice. In terms of results, DCMs perform worse than the alternative classifiers in all of these previous studies, except for one study that does not provide a conclusive result between DNN and NL (*12*). The models with higher performance are typically DNN (8 out of 15) and ensemble models (4 out of 15). When neither DNN nor ensemble models are found to be dominant, the studies (3 out of 15) did not use them in the comparison at all. The sample size of these studies range from the magnitude of 10^3 to 10^5, which are the most common sample sizes from questionnaire surveys or observational datasets. These insights about model choice, performance comparison, and sample sizes aid in structuring our experiments.

TABLE 1: ML classifiers in past studies; (abbreviations are the same as introduced in Section 1)

Author (Year)	Task	Sample Size	Models	Best Model
Nijkamp et al. (1996) (*13*)	Travel Mode	1,396	DNN, MNL	DNN
Rao et al. (1998) (*14*)	Travel Mode	4,335	DNN, MNL	DNN
Hensher and Ton (2000) (*12*)	Travel Mode	801	DNN, NL	DNN/NL
Xie et al. (2003) (*15*)	Travel Mode	34,680	DT, DNN, MNL	DNN
Cantarella et al. (2005) (*2*)	Travel Mode	1,067	DNN, MNL	DNN
Celikoglu (2006) (*16*)	Travel Mode	N.A.	DNN, RBFNN, GRNN, MNL	RBFNN
Pulugurta et al. (2013) (*17*)	Travel Mode	5,822	RBM, MNL	RBM
Tang et al. (2015) (*18*)	Travel Mode	14,000	DT, MNL	DT
Omrani (2015) (*19*)	Travel Mode	9,500	DNN, RBFNN, MNL, SVM	DNN
Sekhar and Madhu (2016) (*20*)	Travel Mode	5,000	RF, DT, MNL	RF
Hagenauer and Helbich (2017) (*8*)	Travel Mode	230,608	MNL, DNN, NB, SVM, CTs, BOOSTING, BAGGING, RF	RF
Tang et al. (2018)	Travel Mode	14,000	DNN	DNN
Wang and Ross (2018) (*9*)	Travel Mode	51,910	BOOSTING, MNL	BOOSTING
Cheng et al. (2019) (*10*)	Travel Mode	7,276	RF, SVM, BOOSTING, MNL	RF
Pirra and Dianna (2019) (*21*)	Travel Mode	39,167	SVM	SVM

However, Table 1 also demonstrates the weaknesses of the past studies. First, the choice of alternative ML classifiers seems quite author-specific. In all of these studies, there is no clear reasoning why certain ML classifiers are included but not the others. Second, the comparisons are typically limited in its scope: Hagenauer and Helbich (2017) (*8*) is the study that has the largest number of ML classifiers, and it incorporates only 8 major ML classifiers. Third, somewhat surprisingly, whereas DCMs are the typical benchmark model in these comparison studies, the authors only focus on MNL model, except for one study that uses NL model as a comparison to DNN (*12*). The limited scope of DCMs is problematic because the state-of-practice DCMs are the NL and the MXL models, not MNL (*7*). Lastly, the conclusions from these previous studies are highly data-specific, particularly depending on the sample size. Different sample sizes could influence the model performance because complex models typically need a large sample size to achieve high prediction accuracy (*22, 23*).

Many studies used ML models to predict other travel-related behaviors, such as traffic flow, accidents (*24–26*), car ownership (*27, 28*), and activity patterns (*29*). Studies germane to our approach that similarly used a large number of ML classifiers for comparison are Fernandez-Delgado et al. (2014) and Kotsiantis et al. (2007) (*30, 31*). Besides prediction accuracy, many other important topics are also related to the application of ML classifiers to travel demand analysis. For example, interpretability and robustness are both critical for the full application of ML methods in practice (*32–36*). These topics are beyond the scope of our study.

3. METHODS AND DATA

3.1. Selection of Classifiers

In light of the selection of classifiers in the past studies (Table 1), we select our ML classifiers based on a balanced concern about *completeness*, *relevance*, and *representativeness*. The full list of our classifiers is summarized in Table 2. We seek to provide a complete list of ML classifiers so that our conclusion about the best classifier does not omit any important alternative model. As a result, the list of ML classifiers has incorporated all the classifiers used in the past studies as summarized in Table 1. However, it is literally impossible to exhaust all ML classifiers in one paper, so we make the list of classifiers representative of all ML classifiers by choosing the most important ones within each one of the 14 model families. For example, in the model family of DCMs, we incorporate three major categories: MNL, NL, and MXL with specific assumptions on the structure of nests and randomness in coefficients, because it is impossible to exhaust all the nest structures and all the combinations of coefficient randomness. Hence the three DCMs, including MNL, NL, and MXL, are representative of DCMs, although not a complete list. Also the selection of ML classifiers is inevitably limited by the practical feasibility of using each software package. The list of ML classifiers might appear slightly redundant in certain model families. For instance, nine models (from DNN_1_30_P to DNN_5_200_P) in the DNN family with varying depth and width are included and counted as nine different models; five naive Bayesian models (from naive_bayes_R to NaiveBayes_W) from the BM family with slightly different hyperparameters are also counted as five different models. The reason for the former is the dramatic impacts of architectural hyperparameters on DNN performance, and the reason for the latter is that different software packages have significantly different underlying algorithms, potentially leading to different model performance. In addition, the list of classifiers are highly relevant to travel behavioral analysis, because the order of this list is roughly sorted according to the importance of the ML classifiers based on the number of papers that used each classifier in the past. It is intuitive that

DCMs are the most relevant ones because the transportation field have a long tradition of using DCMs for behavioral analysis, and DNNs are the second most important ones due to its rising popularity in many subdomains in transportation (*37*).

TABLE 2: List of 86 ML classifiers from 14 model families

Classifiers	Model Families	Description	Language & Function
1. Discrete Choice Models (3 Models)			
mnl_B	DCM	Multinomial logit model	Python Biogeme
nl_B	DCM	Nested logit model (motor vs. nonmotor nests)	Python Biogeme
mxl_B	DCM	Mixed logit model (ASC's as random variables)	Python Biogeme
2. Deep Neural Networks (16 Models)			
mlp_R	DNN	Multi-layer perceptrons (MLP)	R RSNNS mlp
mlpWeightDecay_R	DNN	MLP with weight decay	R Caret mlpWeightDecay
avNNet_R	DNN	Neural network with random seeds with averaged scores; (*38*)	R Caret avNNet
nnet_R	DNN	Single layer neural network with BFGS algorithm	R Caret nnet
pcaNNet_R	DNN	PCA pretraining before applying neural networks	R Caret pcaNNet
monmlp_R	DNN	MLP with monotone constraints (*39*)	R Caret monmlp
mlp_W	DNN	MLP with sigmoid hidden neurons and unthresholded linear output neurons	Weka MultilayerPerceptron
DNN_1_30_P	DNN	MLP with one hidden layer and 30 neurons in each layer	Python Tensorflow
DNN_3_30_P	DNN	MLP with three hidden layers and 30 neurons in each layer	Python Tensorflow
DNN_5_30_P	DNN	MLP with five hidden layer and 30 neurons in each layer	Python Tensorflow
DNN_1_100_P	DNN	MLP with one hidden layer and 100 neurons in each layer	Python Tensorflow
DNN_3_100_P	DNN	MLP with three hidden layers and 100 neurons in each layer	Python Tensorflow
DNN_5_100_P	DNN	MLP with five hidden layers and 100 neurons in each layer	Python Tensorflow
DNN_1_200_P	DNN	MLP with one hidden layer and 200 neurons in each layer	Python Tensorflow
DNN_3_200_P	DNN	MLP with three hidden layers and 200 neurons in each layer	Python Tensorflow
DNN_5_200_P	DNN	MLP with five hidden layers and 200 neurons in each layer	Python Tensorflow
3. Discriminant Analysis (12 Models)			
lda_R	DA	Linear discriminant analysis (LDA) model	R MASS lda
lda2_R	DA	LDA tuning the number of components to retain up to #classes - 1	R MASS Caret
lda_P	DA	LDA solved by singular value decomposition without shrinkage	Python sklearn LinearDiscriminantAnalysis
sda_R	DA	LDA with Correlation-Adjusted T (CAT) scores for variable selection	R Caret
lda_shrink_P	DA	LDA solved by least squares with automatic shrinkage based on Ledoit-Wolf lemma used.	Python sklearn LinearDiscriminantAnalysis
slda_R	DA	LDA developed based on left-spherically distributed linear scores	R Caret ipred

stepLDA_R	DA	LDA model with forward/backward stepwise feature selection	R Caret klaR
pda_R	DA	Penalized discriminant analysis (PDA) with shrinkage penalty coefficients (*40*)	R mda gen.ridge
mda_R	DA	Mixture discriminant analysis (MDA) where the number subclass is tuned to 3 (*41*)	R mda
rda_R	DA	Regularized discriminant analysis (RDA) with regularized group covariance matrices (*42*)	R klaR
hdda_R	DA	High dimensional discriminant analysis (hdda) assuming each class in a Gaussian subspace (*43*)	R HD
qda_R	DA	Quadratic discriminant analysis (qda)	Python sklearn QuadraticDiscriminantAnalysis
4. Bayesian Models (6 Models)			
naïve_bayes_R	BM	Naive Bayes (NB) classifier with the normal kernel density (Laplace correction factor = 2 and Bandwidth Adjustment = 1)	R naïvebayes
NaiveBayes_R	BM	NB classifier with the normal kernel density (Laplace correction factor = 2 and Bandwidth Adjustment = 1)	R klaR NaiveBayes
BernoulliNB_P	BM	NB model with Bernoulli kernel density function	Python sklearn BermoulliNB
GaussianNB_P	BM	NB model with Gaussian kernel density function (smoothing = 5, according to the variance portions)	Python sklearn GaussianNB
BayesNet_W	BM	Bayes network models by hill climbing algorithm (*44*)	Weka BayesNet
NaiveBayes_W	BM	NB model with Gaussian kernel density function	Weka NaiveBayes
5. Support Vector Machines (7 Models)			
svmRadial_R	SVM	Support Vector Machine (SVM) model with Gaussian kernel (inverse kernel width = 1)	R Caret kernlab
svmRadialCost_R	SVM	SVM with Gaussian kernel (automatic spread of the Gaussian kernel)	R Caret kernlab
svmPoly_R	SVM	SVM with polynomial kernel	R Caret kernlab
lssvmRadial_R	SVM	Least Squares SVM model with Gaussian kernel	R Caret kernlab
LinearSVC_l1_P	SVM	SVM with linear kernel and l1 penalty	Python sklearn LinearSVC
LinearSVC_l2_P	SVM	SVM with linear kernel and l2 penalty	Python sklearn LinearSVC
SVM_tf_P	SVM	SVM with l2 penalty using Adam optimizer	Python Tensorflow
6. K Nearest Neighbors (4 Models)			
KNN_1_P	KNN	k-nearest neighbors (KNN) classifier with number of neighbors equal to 1	Python sklearn KNeighborsClassifier
KNN_5_P	KNN	KNN classifier with number of neighbors equal to 5	Python sklearn KNeighborsClassifier
lBk_1_W	KNN	KNN classifier with number of neighbors equal to 1 (brute force searching and Euclidean distance) (*45*)	Weka lBk
lBk_5_W	KNN	KNN classifier with number of neighbors equal to 5 (brute force searching and Euclidean distance) (*45*)	Weka lBk
7. Decision Tree (12 Models)			
rpart_R	DT	Recursive partitioning and regression trees (RPART) model (max depth = 30)	R rpart
rpart2_R	DT	RPART (max depth = 10)	R Caret klaR
C5.0Tree_R	DT	C5.0 decision tree (confidence factor = 0.25)	R Caret C50
ctree_R	DT	Conditional inference trees (*46*)	R Caret C50
ctree2_R	DT	ctree (max depth = 10)	R Caret C50
DecisionTree_P	DT	Decision tree classification model with Gini impurity split measure	Python sklearn DecisionTreeClassifier

ExtraTree_P	DT	Tree classifier with best splits and features chosen from random splits and randomly selected features (*47*)	Python sklearn ExtraTreeClassifier
DecisionStump_W	DT	Tree model with decision stump	Weka DecisionStump
RandomTree_W	DT	Tree model that considers K randomly chosen features at each node	Weka_RandomTree
HoeffdingTree_W	DT	An incremental tree with inductive algorithm. (*48*)	Weka HoeffdingTree
REPTree_W	DT	Tree model using information gain/variance	Weka REPTree
J48_W	DT	Pruned C4.5 decision tree model	Weka J48
8. Generalized Linear Models (10 Models)			
Logistic Regression_l1_P	GLM	Logistic regression model with l1 penalty	Python sklearn LogisticRegression
Logistic Regression_l2_P	GLM	Logistic regression model with l2 penalty	Python sklearn LogisticRegression
Logistic_W	GLM	Logistic regression model with a ridge estimator (*49*)	Weka Logistic
SimpleLogistic_W	GLM	Linear logistic regression models fitted by using LogitBoost (*50*)	Weka SimpleLogistic
Ridge_P	GLM	Classifier using Ridge regression	Python sklearn RidgeClassifier
Passive Aggressive_P	GLM	Passive-aggressive algorithms for classification with hinge loss (*51*)	Python sklearn PassiveAggressiveClassifier
SGD_Hinge_P	GLM	Linear classifier with hinge loss and SGD training	Python sklearn SGDClassifier
SGD_Squared Hinge_P	GLM	Linear classifiers of SGD training with squared hinge loss function	Python sklearn SGDClassifier
SGD_Log_P	GLM	Linear classifiers of SGD training with log loss function	Python sklearn SGDClassifier
SGD_Modified Huber_P	GLM	Linear classifiers of SGD training with modified huber loss function	Python sklearn SGDClassifier
9. Gaussian Process (3 Models)			
GP_Constant_P	GP	Gaussian Processes classification model with constant kernel	Python sklearn GaussianProcessClassifier
GP_DotProduct_P	GP	Gaussian Processes classification model with DotProduct kernel	Python sklearn GaussianProcessClassifier
GP_Matern_P	GP	Gaussian Processes classification model with Matern kernel	Python sklearn GaussianProcessClassifier
10. Rule-Based Methods (3 Models)			
DecisionTable_W	RBM	Simple decision table majority classier that uses BestFirst as search method (*52*)	Weka DecisionTable
OneR_W	RBM	A classifier using one-rule on the input with the lowest error (*53*)	Weka OneR
ZeroR_W	RBM	A classifier that predicts the mean class for all the test patterns	Weka ZeroR
11. Bagging (3 Models)			
Bagging_SVM_P	BAGGING	A bagging classifier that fits base classifiers based on random subsets of the original dataset; SVM is the base classifier	Python sklearn BaggingClassifier
Bagging_Tree_P	BAGGING	A bagging classifier with DecisionTree as the base classifier	Python sklearn BaggingClassifier
Bagging_REP_W	BAGGING	A bagging classifier with REPTree as the base classifier (*54*)	Weka Bagging
12. Random Forests (2 Models)			
RandomForest_P	RF	A random forest model with 10 trees in the forest	Python sklearn RandomForestClassifier

ExtraTrees_P	RF	A meta estimator that fits 10 ExtraTree classifiers	Python sklearn ExtraTreeClassifier
13. Boosting (3 Models)			
AdaBoost_P	BOOSTING	AdaBoost classifier. The DecisionTree with maximum depth =10 is set as the base estimator. (*55*)	Python sklearn AdaBoostClassifier
AdaBoostM1_W	BOOSTING	Boosting method with DecisionStump as the base classifier	Weka AdaboostM1
Gradient Boosting_P	BOOSTING	An additive model trained in a forward stage-wise fashion (*56*)	Python sklearn GradientBoostingClassifier
14. Others (2 Models)			
Voting_P	OTHERS	A classifier which combine machine learning classifiers and use a majority vote. We use lda_P, LinearSVM and Logistic classifiers here.	Python sklearn VotingClassifier
Attribute Selected_W	OTHERS	Use J48 trees to classify patterns reduced by attribute selection (Hall, 1998)	Weka AttributeSelected

The classifiers in Table 2 come from four predominant coding languages: Python, R, Biogeme, and Weka. Each one of the coding languages is abbreviated as _P, _R, _B, and _W, attached after the name of each classifier in the first column of Table 2. The third and fourth Columns of Table 2 describe each classifier and the specific functions in each coding language. Overall, our list of classifiers are relatively complete, highly representative of all ML classifiers, and highly relevant to the travel behavioral analysis.

3.2. NHTS 2017 Dataset

The NHTS2017 dataset is used for this empirical study because it has a wide geographical coverage and a large sample size. NHTS2017 broadly covers all the states and the major metropolitan areas in the United States. The full sample size is $781,831$, larger than all the sample sizes used in previous studies. The NHTS2017 dataset is also publically available, so future studies can continue to work on this dataset to improve our results [1]. One caveat with NHTS dataset is the lack of alternative-specific variables. This is because the origin-destination (OD) information is not granular enough to compute meaningful travel cost and time for each travel alternative. But the missing information should not have a large impact on the relative relationship between models, although it does have an impact on the maximum possible prediction accuracy achieved by our classifiers. Nonetheless, the wide geographical coverage, the large sample size, and the publicity of NHTS prompt us to use it for this empirical benchmark paper.

3.3. Training

To test the effects of sample size, we resample our training and testing sets with a ratio of $4:1$ and the total number of observations equal to $1,000$, $10,000$, and $100,000$. Five-fold cross-validation is used to compute the average prediction accuracy in the testing sets for each classifier. The dependent variable of this study is only the travel mode choice, although the dataset does incorporate other important decision variables such as car ownership and activity patterns that are not used in our current study. The travel mode choice incorporates 6 travel modes, including walking+bicycles, car, SUV, van+truck, public transit, and others. In total, 115 independent variables are used for prediction, including income, age, gender, and many other important socio-economic and travel-related variables. These input variables are selected from the full NHTS2017 dataset as

[1]The data is available in https://nhts.ornl.gov/

those most relevant to the output prediction. The inputs are further normalized before the model training. In total, we trained and examined 1,290 models.

4. RESULTS

4.1. Comparing Prediction Accuracy of Model Families

Table 1 summarizes the 14 model families sorted from the highest to the lowest according to their median prediction accuracy. In Table 1, the green bars connect the minimum and maximum prediction accuracy of the models in each model family; each green dot represents the median prediction accuracy and each red represents the mean. The sorting is based on the median prediction accuracy because median values are more robust than mean to extremely large or small outliers. Overall, the prediction accuracy of all the model families range between 35% and 55%. Whereas these prediction accuracy values seem low compared to previous studies, the difficulty could be caused by the large number and the high imbalance of the travel mode alternatives.

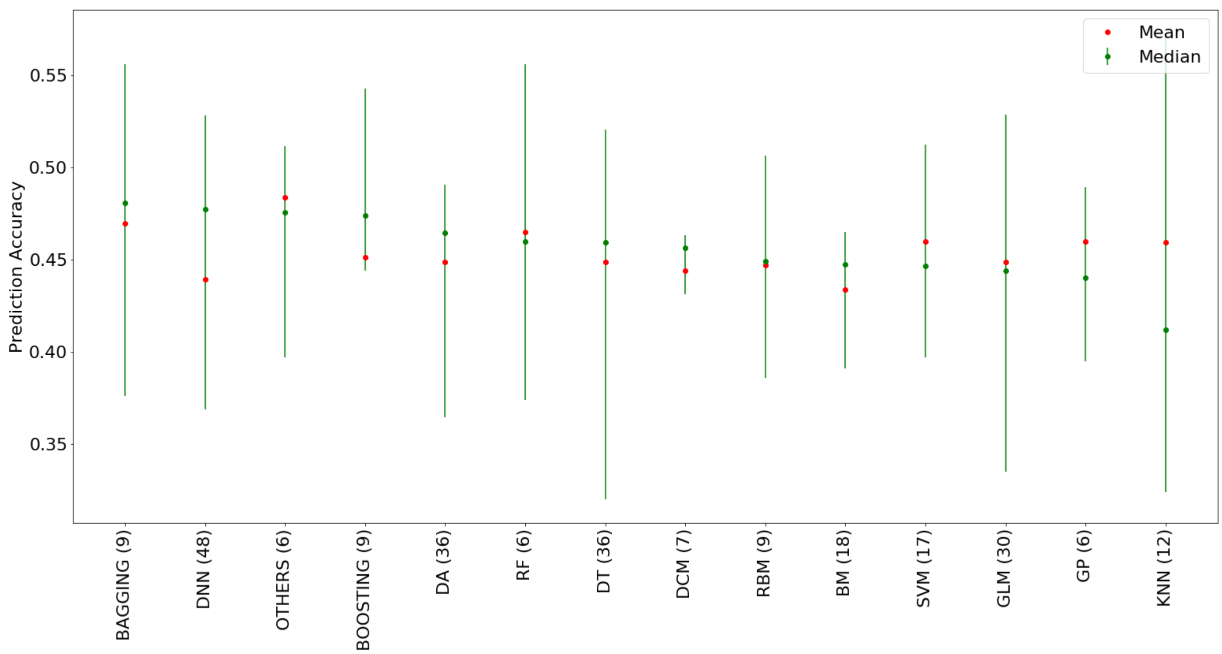

FIGURE 1: Prediction accuracy of 14 model families

Figure 1 shows that the best models are the ensemble models that integrate several models into one for prediction. The ensemble models include bagging, boosting, random forests, and the other ensemble methods, which are ranked as the first, the fourth, the sixth, and the third among all the 14 classifier families. Among these ensemble methods, bagging achieves the highest prediction accuracy among the 14 model families. Besides the median, the mean prediction accuracy of bagging methods is also relatively high, ranked as the second among the 14 model families. The highest prediction accuracy of the bagging methods reaches about 55.6%, much higher than the prediction accuracy of all the classifiers in the top 5 model families. This finding is quite reasonable. Researchers argued that ensemble models can be treated as one regularization method (*57*), and regularization is very important for the models with high complexity because it reduces the large variance in estimation (*58*). Researchers even demonstrated the relationship between model ensemble and other regularization methods, such as dropout in DNN (*59*) and Bayesian prior as the mixing weights of ensemble models (*60*). Moreover, the dominant performance of ensemble models is intuitive because the ensemble models literally use more models than individual ones. For example, the Voting method in the category of OTHERS, uses the majority vote of many individual classifiers, leading to more robust results than each separate one.

DNN, DCM, and KNN are the other three noteworthy model families. First, DNN performs the best among all the non-ensemble models (ranked as the second). Note that we have not incorporated a DNN ensemble model into our list of classifiers, so it is possible that DNN ensemble can perform better than all the individual DNN models and the ensemble models currently incorporated in the list. Second, DCM models perform in the medium range of the 14 model families. Given that DCM models have dominated the field of choice modeling for decades, our results show that DCM models are far from the best choice even for the classical travel mode choice analysis, at least for the sake of prediction. This finding is consistent with a large number of previous studies that found the worse performance of DCMs relative to other classifiers, as summarized in Table 1. Whereas most of the previous studies limit their scope of analysis to only MNL model, our results about DCMs have incorporated both nested logit (NL) and mixed logit (MXL) models. Therefore, our results provide stronger evidence that DCMs, even incorporating the nest structures or randomness in ASCs and coefficients, cannot perform better than ensemble models and DNNs. Lastly, one of the KNN models perform the best among all the models, but the KNN model family has the worst performance according to our ranking, and the variance of the KNN classifiers is also the largest. The large variance implies that the highest performance of one KNN classifier is data and model specific, thus not generalizable.

4.2. Comparing Prediction Accuracy of Individual Models

With a format similar to Figure 1, Figure 2 summarizes the prediction accuracy of each single ML classifier, ranked based on the median prediction accuracy from the highest to the lowest. Figure 2 highlights the DCM models by red.

Again, we observe the high performance of the ensemble models in Figure 2. For example, GradientBoosting_P perform the best out of the 86 ML classifiers, and Bagging_REP_W is ranked as the fourth. These two classifiers have not only relatively high median prediction accuracy, but also low variance (short green bars), in comparison to other three top 5 models. Interestingly, the ML classifiers ranked as the second, the third, and the fifth, belong to the same model family GLM, although GLM is not ranked as high in Figure 1. Therefore, we treat the high performance of these GLM models as single instances rather than a consistent pattern. As to the DCM models, again all three models (MNL, NL, and MXL) perform in the medium range of all the 86 models. The MNL and NL models perform even slightly better than the MXL model. The problem with the MXL model is not exactly the prediction accuracy, but more related to the computational issue. Note that the MXL model has *only* one point concerning its prediction accuracy, evaluated as sample size equals to 1000. The MXL model cannot be trained in a reasonable amount of time (< 24 hours) in Python Biogeme when the sample size reaches even $10,000$. This is because the training of MXL relies on sampling, which takes much more time than the gradient-based methods used in many other classifiers, even including DNNs, which are notoriously known for the difficulty in training and convergence.

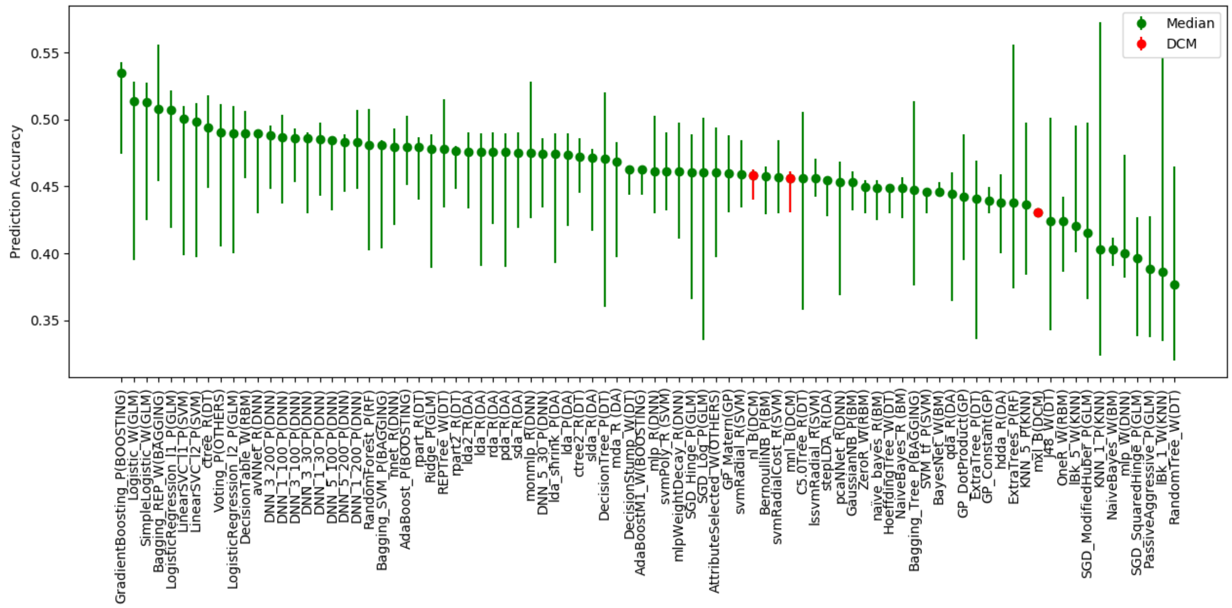

FIGURE 2: Prediction accuracy of 86 ML classifiers

4.3. Sample Size Effects

Sample size is one important meta-dimension in our study. Figure 3 summarizes how prediction accuracy of all the models vary with different sample sizes. As shown in this figure, sample size has a clear impact on prediction accuracy: as sample size increases, the prediction accuracy of all the models increase dramatically. When the sample size is about 1000, the average prediction accuracy is only about 42.7%, and it increases to about 47.5% and 48.9% as the sample size becomes 10000 and 100000.

The sample size effect exists particularly for the models with high complexity, such as DTs and DNNs, because they need large sample sizes to control their estimation errors, thus achieving high prediction accuracy. For example, the prediction accuracy of MNL models is about 43.1% when sample size equals to $1,000$, and it becomes 46.1% when sample size equals to $100,000$, showing 3% increase. As a comparison, the prediction accuracy of DNN_1_100_P model is about 43.7% when sample size equals to 1000, and it becomes 50.3% when sample size equals to $100,000$, showing about 7% increase, which is much larger than MNL models. Theoretically, this difference is caused by the different model complexity of MNL and DNN (*22, 23, 61, 62*). The estimation error of simple models such as MNL is always well bounded even when sample size is relatively small, whereas the estimation error of complex models such as DNN is not well bounded, leading to the result that larger sample sizes enable DNNs to achieve higher prediction accuracy than small sample. This large sample size effect also exists in other models with high model complexity, such as BAGGING, BOOSTING, and RF.

5. CONCLUSION AND DISCUSSIONS

This study is motivated by the importance of using ML classifiers to predict travel demand and the limitation of data-specific and author-specific conclusions in the recent studies that compare choice models to a small number of ML classifiers. To achieve a generalizable result and provide

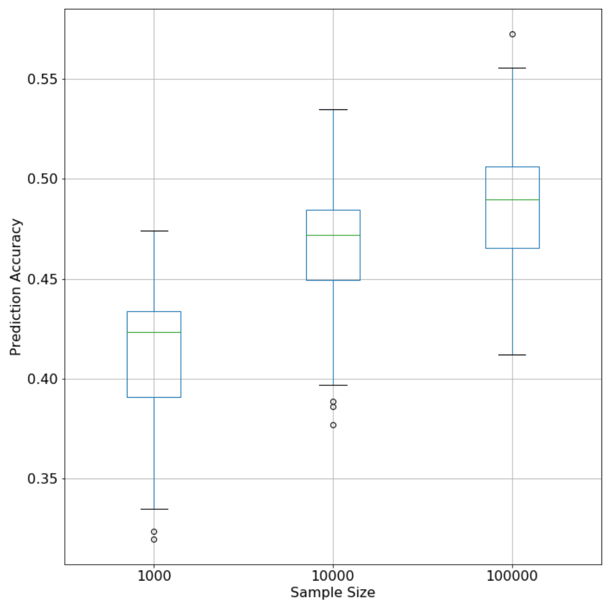

FIGURE 3: Prediction Accuracy with Sample Sizes

an empirical benchmark for future studies, we analyze the prediction accuracy of 86 models from 14 model families on the travel mode choice based on the NHTS2017 dataset. The 86 models and the 14 model families include all the important ML classifiers used in the previous studies that focus on any type of travel behavioral prediction with ML classifiers. The 86 models are chosen based on the principles of completeness, relevance, and representativeness, and they also include the important models (e.g. MXL model) that are never examined in the previous studies that compared ML classifiers. Besides the number of ML classifiers, our experiment also expands to the meta-dimension of sample size, covering the range from a typical size in a questionnaire survey (10^3) to the maximum found in previous studies (10^5). With this setup, this study yields the following major findings.

First, ensemble models including BAGGING, BOOSTING, and Random Forests perform the best in the 14 model families. This result is intuitive because ensemble models combines many individual classifiers, thus being more powerful than each individual one. Among non-ensemble methods, DNN has the highest prediction accuracy. Second, DCMs have only medium and relatively low level of prediction accuracy. This result holds for all three major DCMs, including MNL, NL, and MXL. To make things even worse, it is computationally impossible to train MXL model when sample size reaches 10^4 or 10^5 in a reasonable amount of time, at least given the current algorithm coded in Python Biogeme. These results about ensemble models, DNNs, and DCMs are actually consistent with the past studies, the majority of which found the superior prediction accuracy of ensemble models such as RF and DNNs over traditional DCMs. Lastly, we observe a clear effect of sample size on prediction accuracy. With larger sample sizes (from 10^3 to 10^5), ML classifiers can achieve significantly higher prediction accuracy. But this effect mainly holds for the models with high model complexity, such as DNN, BAGGING, and BOOSTING; this effect is much more limited for simple models, such as DCMs.

Many limitations exist for this study. For instance, whereas this study has incorporated a massive number of ML classifiers, the list of ML classifiers can never be truly complete. On the one hand, researchers constantly create more ML classifiers for various domain-specific questions, and these classifiers can always be augmented to our list. On the other hand, even conditioning on our current list of classifiers, the infinite possibilities of hyperparameters in even one model preclude a truly complete list of ML classifiers. Moreover, the current study considers sample size as the only meta-dimension, whereas several other meta-dimensions could render our results more generalizable. For example, prediction accuracy of ML classifiers heavily depends on the specific travel behaviors in prediction. It is intriguing to compare the results of predicting travel mode choice to others such as car ownership choices. Nonetheless, with the large scale of ML classifiers tested and the meta-dimensions incorporated, our study provides valuable benchmarks for future empirical studies.

More importantly, we see our study as one first step for the field of travel demand analysis to start working on some publicly accepted benchmark dataset, rather than the datasets collected by each individual researcher. The shared public benchmark dataset enables researchers to consistently build their own work upon others and to avoid confusing results that could be caused by data-specific and author-specific issues. We think the NHTS2017 dataset as a public dataset with wide geographical coverage and a large sample size suffices to be the empirical benchmark dataset in this field. We encourage future studies to explore more interesting and novel ML classifiers and test them on the same dataset to beat our results, making the knowledge of each individual researcher more comparable, replicable, and generalizable.

ACKNOWLEDGEMENTS

The research is supported by the National Research Foundation (NRF), Prime Minister's Office, Singapore, under her CREATE programme, Singapore-MIT Alliance for Research and Technology (SMART) Centre, Future Urban Mobility (FM) IRG.

AUTHOR CONTRIBUTION

The authors confirm contribution to the paper as follows: study conception and design: S. Wang and J. Zhao; data collection and computation: B. Mo; analysis and interpretation of results: S. Wang; draft manuscript preparation: S. Wang; supervising: J. Zhao; All authors reviewed the results and approved the final version of the manuscript.

REFERENCES

[1] Ben-Akiva, M., J. L. Bowman, and D. Gopinath, Travel demand model system for the information era. *Transportation*, Vol. 23, No. 3, 1996, pp. 241–266.

[2] Cantarella, G. E. and S. de Luca, Multilayer feedforward networks for transportation mode choice analysis: An analysis and a comparison with random utility models. *Transportation Research Part C: Emerging Technologies*, Vol. 13, No. 2, 2005, pp. 121–155.

[3] Ben-Akiva, M., M. Bierlaire, D. McFadden, and J. Walker, *Discrete Choice Analysis*, 2014.

[4] Small, K. A., E. T. Verhoef, and R. Lindsey, Travel Demand. In *The economics of urban transportation*, Routledge, Vol. 2, 2007.

[5] De Dios Ortuzar, J. and L. G. Willumsen, *Modelling transport*. John Wiley and Sons, 2011.

[6] Annaswamy, A. M., Y. Guan, H. E. Tseng, H. Zhou, T. Phan, and D. Yanakiev, Transactive Control in Smart Cities. *Proceedings of the IEEE*, Vol. 106, No. 4, 2018, pp. 518–537.

[7] Train, K. E., *Discrete choice methods with simulation*. Cambridge university press, 2009.

[8] Hagenauer, J. and M. Helbich, A comparative study of machine learning classifiers for modeling travel mode choice. *Expert Systems with Applications*, Vol. 78, 2017, pp. 273–282.

[9] Wang, F. and C. L. Ross, Machine learning travel mode choices: Comparing the performance of an extreme gradient boosting model with a multinomial logit model. *Transportation Research Record*, Vol. 2672, No. 47, 2018, pp. 35–45.

[10] Cheng, L., X. Chen, J. De Vos, X. Lai, and F. Witlox, Applying a random forest method approach to model travel mode choice behavior. *Travel behaviour and society*, Vol. 14, 2019, pp. 1–10.

[11] Ben-Akiva, M. E. and S. R. Lerman, *Discrete choice analysis: theory and application to travel demand*, Vol. 9. MIT press, 1985.

[12] Hensher, D. A. and T. T. Ton, A comparison of the predictive potential of artificial neural networks and nested logit models for commuter mode choice. *Transportation Research Part E: Logistics and Transportation Review*, Vol. 36, No. 3, 2000, pp. 155–172.

[13] Nijkamp, P., A. Reggiani, and T. Tritapepe, Modelling inter-urban transport flows in Italy: A comparison between neural network analysis and logit analysis. *Transportation Research Part C: Emerging Technologies*, Vol. 4, No. 6, 1996, pp. 323–338.

[14] Rao, P. S., P. Sikdar, K. K. Rao, and S. Dhingra, Another insight into artificial neural networks through behavioural analysis of access mode choice. *Computers, environment and urban systems*, Vol. 22, No. 5, 1998, pp. 485–496.

[15] Xie, C., J. Lu, and E. Parkany, Work travel mode choice modeling with data mining: decision trees and neural networks. *Transportation Research Record: Journal of the Transportation Research Board*, , No. 1854, 2003, pp. 50–61.

[16] Celikoglu, H. B., Application of radial basis function and generalized regression neural networks in non-linear utility function specification for travel mode choice modelling. *Mathematical and Computer Modelling*, Vol. 44, No. 7, 2006, pp. 640–658.

[17] Pulugurta, S., A. Arun, and M. Errampalli, Use of artificial intelligence for mode choice analysis and comparison with traditional multinomial logit model. *Procedia-Social and Behavioral Sciences*, Vol. 104, 2013, pp. 583–592.

[18] Tang, L., C. Xiong, and L. Zhang, Decision tree method for modeling travel mode switching in a dynamic behavioral process. *Transportation Planning and Technology*, Vol. 38, No. 8, 2015, pp. 833–850.

[19] Omrani, H., Predicting travel mode of individuals by machine learning. *Transportation Research Procedia*, Vol. 10, 2015, pp. 840–849.

[20] Sekhar, C. R. and E. Madhu, Mode Choice Analysis Using Random Forrest Decision Trees. *Transportation Research Procedia*, Vol. 17, 2016, pp. 644–652.

[21] Pirra, M. and M. Diana, A study of tour-based mode choice based on a Support Vector Machine classifier. *Transportation Planning and Technology*, Vol. 42, No. 1, 2019, pp. 23–36.

[22] Vapnik, V., *The nature of statistical learning theory*. Springer science and business media, 2013.

[23] Vapnik, V. N., An overview of statistical learning theory. *IEEE transactions on neural networks*, Vol. 10, No. 5, 1999, pp. 988–999.

[24] Mozolin, M., J.-C. Thill, and E. L. Usery, Trip distribution forecasting with multilayer perceptron neural networks: A critical evaluation. *Transportation Research Part B: Methodological*, Vol. 34, No. 1, 2000, pp. 53–73.

[25] Polson, N. G. and V. O. Sokolov, Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies*, Vol. 79, 2017, pp. 1–17.

[26] Wu, Y., H. Tan, L. Qin, B. Ran, and Z. Jiang, A hybrid deep learning based traffic flow prediction method and its understanding. *Transportation Research Part C: Emerging Technologies*, Vol. 90, 2018, pp. 166–180.

[27] Paredes, M., E. Hemberg, U.-M. O'Reilly, and C. Zegras, Machine learning or discrete choice models for car ownership demand estimation and prediction? In *Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2017 5th IEEE International Conference on*, IEEE, 2017, pp. 780–785.

[28] Kaewwichian, P., L. Tanwanichkul, and J. Pitaksringkarn, Car Ownership Demand Modeling Using Machine Learning: Decision Trees and Neural Networks. *International Journal of Geomate*, Vol. 17, No. 62, 2019, pp. 219–230.

[29] Allahviranloo, M. and W. Recker, Daily activity pattern recognition by using support vector machines with multiple classes. *Transportation Research Part B: Methodological*, Vol. 58, 2013, pp. 16–43.

[30] Fernández-Delgado, M., E. Cernadas, S. Barro, and D. Amorim, Do we need hundreds of classifiers to solve real world classification problems. *Journal of Machine Learning Research*, Vol. 15, No. 1, 2014, pp. 3133–3181.

[31] Kotsiantis, S. B., I. Zaharakis, and P. Pintelas, Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, Vol. 160, 2007, pp. 3–24.

[32] Lipton, Z. C., The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

[33] Doshi-Velez, F. and B. Kim, Towards a rigorous science of interpretable machine learning, 2017.

[34] Montavon, G., W. Samek, and K.-R. Muller, Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, Vol. 73, 2018, pp. 1–15.

[35] Nguyen, A., J. Yosinski, and J. Clune, Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 427–436.

[36] Goodfellow, I. J., J. Shlens, and C. Szegedy, Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2015.

[37] Karlaftis, M. G. and E. I. Vlahogianni, Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. *Transportation Research Part C: Emerging Technologies*, Vol. 19, No. 3, 2011, pp. 387–399.

[38] Ripley, B. D. and N. Hjort, *Pattern recognition and neural networks*. Cambridge university press, 1996.

[39] Zhang, H. and Z. Zhang, Feedforward networks with monotone constraints. In *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339)*, IEEE, 1999, Vol. 3, pp. 1820–1823.

[40] Hastie, T., A. Buja, and R. Tibshirani, Penalized discriminant analysis. *The Annals of Statistics*, 1995, pp. 73–102.

[41] Hastie, T. and R. Tibshirani, Discriminant analysis by Gaussian mixtures. *Journal of the Royal Statistical Society: Series B (Methodological)*, Vol. 58, No. 1, 1996, pp. 155–176.

[42] Friedman, J. H., Regularized discriminant analysis. *Journal of the American statistical association*, Vol. 84, No. 405, 1989, pp. 165–175.

[43] Bouveyron, C., S. Girard, and C. Schmid, High-dimensional discriminant analysis. *Communications in Statistics—Theory and Methods*, Vol. 36, No. 14, 2007, pp. 2607–2623.

[44] Cooper, G. F. and E. Herskovits, A Bayesian method for the induction of probabilistic networks from data. *Machine learning*, Vol. 9, No. 4, 1992, pp. 309–347.

[45] Aha, D. W., D. Kibler, and M. K. Albert, Instance-based learning algorithms. *Machine learning*, Vol. 6, No. 1, 1991, pp. 37–66.

[46] Hothorn, T., K. Hornik, and A. Zeileis, Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical statistics*, Vol. 15, No. 3, 2006, pp. 651–674.

[47] Geurts, P., D. Ernst, and L. Wehenkel, Extremely randomized trees. *Machine learning*, Vol. 63, No. 1, 2006, pp. 3–42.

[48] Hulten, G., L. Spencer, and P. Domingos, Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2001, pp. 97–106.

[49] Le Cessie, S. and J. C. Van Houwelingen, Ridge estimators in logistic regression. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, Vol. 41, No. 1, 1992, pp. 191–201.

[50] Landwehr, N., M. Hall, and E. Frank, Logistic model trees. *Machine learning*, Vol. 59, No. 1-2, 2005, pp. 161–205.

[51] Crammer, K., O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, Online passive-aggressive algorithms. *Journal of Machine Learning Research*, Vol. 7, No. Mar, 2006, pp. 551–585.

[52] Kohavi, R., The power of decision tables. In *European conference on machine learning*, Springer, 1995, pp. 174–189.

[53] Holte, R. C., Very simple classification rules perform well on most commonly used datasets. *Machine learning*, Vol. 11, No. 1, 1993, pp. 63–90.

[54] Breiman, L., Bagging predictors. *Machine learning*, Vol. 24, No. 2, 1996, pp. 123–140.

[55] Freund, Y. and R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, Vol. 55, No. 1, 1997, pp. 119–139.

[56] Friedman, J. H., Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 2001, pp. 1189–1232.

[57] Bishop, C. M., *Pattern recognition and machine learning*. springer, 2006.
[58] Goodfellow, I., Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, Vol. 1. MIT press Cambridge, 2016.
[59] Srivastava, N., G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, Vol. 15, No. 1, 2014, pp. 1929–1958.
[60] Boyd, S. and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
[61] Von Luxburg, U. and B. Schölkopf, Statistical learning theory: Models, concepts, and results. In *Handbook of the History of Logic*, Elsevier, Vol. 10, 2011, pp. 651–706.
[62] Wainwright, M. J., *High-dimensional statistics: A non-asymptotic viewpoint*, Vol. 48. Cambridge University Press, 2019.