Comparing Validation and Classification Methods for Motor Imagery Decoding through Common Spatial Pattern Filters

Rien Sonck

Supervisor: Prof. Dr. Daniele Marinazzo

Ghent University

Department of Experimental Psychology

Author Note

A project paper submitted to Ghent University in fulfilment of the requirements for the course Case Studies in the Analysis of Experimental Data.

Academic year: 2019 - 2020

Student Number: 01402685

Dataset: github.com/riensonck/CAED2020/tree/master/MNE-eegbci-data

Python Script: github.com/riensonck/CAED2020/tree/master/Analysis

Comparing Validation and Classification Methods for Motor Imagery Decoding through Common Spatial Pattern Filters

Brain-Computer Interfaces (BCIs) is an emerging neurotechnology that aims to translate brain activity to computer signals, as illustrated in figure 1, giving users the possibility to control external devices with their brain activity and without the need of their peripheral nerves and muscles. Users their brain activity is measured using either invasive intracranial electrodes such as electrocorticography (ECoG) (Hill et al., 2012) or by using non-invasive scalp electrodes such as electroencephalography (EEG) (Biasiucci et al., 2019). EEG is the most used technique of signal acquisition in the BCI field. After signal acquisition, machine learning techniques are employed to (1) preprocess the brain activity, (2) extract and select relevant features, (3) decode these features, and (4) translate it into a computer signal (see Iturrate et al., 2020, for a review). BCI systems are being developed for a wide range of fields and applications (see Abdulkader et al., 2015, for a review). One promising field is the usage of BCI for assisting motor-disabled users. Here, motor imaginary (MI), the imagination of different types of body movements (Decety, 1996), is used teach motor-disabled users to successfully control applications such as writing applications for communication or robotic tel-presence platforms for assistive mobility (Leeb et al., 2013). MI activates similar brain regions as motor planning and motor execution and retains the same temporal brain activity characteristics (Batula et al., 2017; Ehrsson et al., 2003; Jeannerod, 2001; Jeannerod & Frak, 1999; Lotze & Halsband, 2006; Munzert et al., 2009). Xygonakis et al. (2018) tested the performance of four machine learning techniques to classify brain signals belonging to either left-hand, right-hand, feet, or tongue MI tasks through common spatial pattern (CSP) filters (Ramoser et al., 2000). These classifiers were Linear Discriminant Analysis (LDA) (Tharwat et al., 2017), K-nearest Neighbors (KN) (Zhang, 2016), Decision Tree Learning (DTL) (Patel, 2012), and Naive Bayens (NB) (Raschka, 2017). They found that LDA, which is among the most common classification methods used in BCI systems (Bashashati et al., 2015; Iturrate et al., 2020), had superior performance among all subjects compared to the other three classifiers. Bashashati et al. (2015) found that the performance of a classifier depends on the feature extraction method being used and classifiers such as logistic regression (LR) (Peng et al., 2002) and support vector machines (SVM) (Evgeniou & Pontil, 2001) have the ability to perform better than LDA when using band-power (BP) features. In my current study, I will build on Xygonakis et al. (2018) their study and continue to use CSP as the feature extraction method. The main goal of this project is to investigate if I can replicate that LDA performs best when using CSP. To achieve this, I will use a data-set from Schalk et al. (2004) and compare LDA with other classifiers such as KN, DTL, NB, LR, and SVR. I will also use 10-fold cross-validation and Monte Carlo cross-validation, which are re-sampling procedures that divide the data into two sets: one set to train the model using the classifiers and the other set to test the performance of these classifiers (Refaeilzadeh et al., 2009). Each round, the data is re-sampled and new training and test sets are created which helps to minimize overfitting of the classifiers.

Materials and Methods

Data-set

The data-set from Schalk et al. (2004) is used, which is also available at PhysioNet (Goldberger et al., 2000). This data-set is provided by the Python MNE toolbox (Gramfort, 2013) and contains EEG recordings from 109 healthy participants that performed three runs of 4 motor imagery tasks, shown in table 1. For each subject, 45 MI trials per task were recorded from 64 EEG sensors placed according to the international 10-05 system. In my study, all 109 subject are analysed but only data from the right vs left hand imagery task is used, resulting in three runs per participant. During this task, participants had to imagine opening and closing their left or right hand in response to a target-stimulus appearing either left or right on the computer screen.

Software

All analyses are written in Python (v3.7.7, van Rossum, 1995) supplemented with the pandas library (v.1.0.3, pandas development team, 2020) and the numpy library (v1.16.4, Oliphant, 2006). The MNE library (v0.20.5, Gramfort, 2013) was used for loading the data, preprocessing the data, and extracting and selecting features. The classifier and cross-validation techniques were implemented using the scikit-learn library (v0.22.1, Pedregosa et al., 2011). All graphs were created using the matplotlib library (v3.1.0, Hunter, 2007). The statistical tests were implemented using the SciPy library (Virtanen et al., 2020, pp. v1.4.1,).

Procedure

Signal Preprocessing. For each participant, frequency filtering was done using a band-pass Butterworth infinite impulse response (IIR) filter (Selesnick & Burrus, 1997) to attenuate the power of undesired frequencies. A band-pass range of 7-30Hz was selected given that the mu (9–13Hz), the alpha (8-12Hz) and the beta (13-32Hz) frequency bands are the most important in motor imagery BCIs (Niedermeyer et al., 2005; Pfurtscheller et al., 2005). Next, epochs for each movement imagination were extracted from the filtered signal. To avoid that classifiers would classify signals based on their evoked responses, each epoch was cropped to only contain activity between 1 second and 2 seconds after cue onset.

Feature Extraction. The Common Spatial Patterns (CSP) method (Ramoser et al., 2000) is used for spatial filtering and feature extraction. It is widely used for binary classification problems, and one of the most popular methods to extract features in the motor imagery BCI field. This method improves the signal to nose ratio of the filtered EEG signal and applies a linear transformation to project the 64 EEG channels on a reduced sensor space. In this sensor space, CSP maximises the variance for one class while minimizing the variance for the other class resulting in better discrimination between both classes (e.g. right-hand vs left-hand MI movements) (Blankertz et al., 2008; Ramoser et al., 2000).

Decoding. Six classifiers were used to decode the CSP features: LDA, KN, DTL, NB, LR, and SVR using both 10-fold and Monte Carlo cross-validation. This resulted in 12 classifier and cross-validation pairs for each subject.

Results and Discussion

Across subjects

Friedman statistical tests (Garcıa & Herrera, 2008) were used to compare classifier performance within each cross-validation group. This test is non-parametric and assumes that the differences between classifiers their performance accuracy is random (H_0 hypothesis). In both the 10-fold cross-validation group ($\chi^2 = 32.05, p < 0.01$) and the Monte-Carlo cross-validation group ($\chi^2 = 23.52, p < 0.01$) the classifiers their accuracy performance are statistically different (H_0 hypothesis is rejected). To identify the source of this difference, the Holm-Bonferroni method (Holm, 1979) was used for multiple comparison. In the 10-fold cross-validation group, shown in table 2, the DTL classifier performed worse than the other classifiers. In contrast to the results found in Xygonakis et al. (2018), there was no superior performance of LDA, as shown in figure 2, 3, and 4. In the Monte-Carlo cross-validation group, shown in table 3, both DTL and NB performed worse than the other classifiers. Comparing both cross-validation methods, 10-fold performed superior than Monte-Carlo. According to this study, LDA, KB, LR, and SVC are recommended to use with 10-fold cross-validation when performing MI decoding through CSP filers.

Between Subjects

There is a lot of individual variation between subjects. Therefore, it is recommended to fit a range of classifiers on each subject and select the best one, as illustrated in figure 5 for subject 13. Here, the LR classifier provides the best area under the curve (AUC), which tells us how much the model is capable of distinguishing between both classes. It can also happen that none of the classifiers fit well on a subject, as shown in figure 6 for subject 73. This big difference in classifier performance between subjects is not new to the BCI field. In Leeb et al. (2013) their study only 50% of the participants achieved a good BCI performance and were able to successfully control an application. In this data-set 102 out of 109 participants (93.5%) were able to achieve a classifier accuracy above chance level, of those 30 participants were able to achieve an accuracy above 70%. There can be multiple sources to this difference such as inherent variability in participants their brain physiology and participants their motivation. Further research will have to investigate and describe sources of variability and how we can improve BCIs taking these sources of variability into account.

References

Abdulkader, S. N., Atia, A., & Mostafa, M.-S. M. (2015). Brain computer interfacing: Applications and challenges. *Egyptian Informatics Journal*, *16*(2), 213–230. https://doi.org/10.1016/j.eij.2015.06.002

Bashashati, H., Ward, R. K., Birch, G. E., & Bashashati, A. (2015). Comparing different classifiers in sensory motor brain computer interfaces (D. Yao, Ed.). *PLOS ONE*, *10*(6), e0129435. https://doi.org/10.1371/journal.pone.0129435

Batula, A. M., Mark, J. A., Kim, Y. E., & Ayaz, H. (2017). Comparison of brain activation during motor imagery and motor movement using fNIRS. *Computational Intelligence and Neuroscience*, *2017*, 1–12. https://doi.org/10.1155/2017/5491296

Biasiucci, A., Franceschiello, B., & Murray, M. M. (2019). Electroencephalography. *Current Biology*, *29*(3), R80–R85. https://doi.org/10.1016/j.cub.2018.11.052

Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., & Muller, K.-r. (2008). Optimizing spatial filters for robust EEG single-trial analysis. *IEEE Signal Processing Magazine*, *25*(1), 41–56. https://doi.org/10.1109/MSP.2008.4408441

Decety, J. (1996). The neurophysiological basis of motor imagery. *Behavioural Brain Research*, *77*(1), 45–52. https://doi.org/10.1016/0166-4328(95)00225-1

Ehrsson, H. H., Geyer, S., & Naito, E. (2003). Imagery of voluntary movement of fingers, toes, and tongue activates corresponding body-part-specific motor representations. *Journal of Neurophysiology*, *90*(5), 3304–3316. https://doi.org/10.1152/jn.01113.2002

Evgeniou, T., & Pontil, M. (2001). Support vector machines: Theory and applications (G. Paliouras, V. Karkaletsis, & C. D. Spyropoulos, Eds.). In G. Paliouras, V. Karkaletsis, & C. D. Spyropoulos (Eds.). G. Goos, J. Hartmanis, & J. van Leeuwen (Eds.), *Machine learning and its applications*. Berlin, Heidelberg, Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-44673-7_12

Garcıa, S., & Herrera, F. (2008). An extension on “statistical comparisons of classifiers over multiple data sets” for all pairwise comparisons. *Journal of Machine Learning Research*, *9*, 2677–2694.

Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, *101*(23). https://doi.org/10.1161/01.CIR.101.23.e215

Gramfort, A. (2013). MEG and EEG data analysis with MNE-python. *Frontiers in Neuroscience*, *7*. https://doi.org/10.3389/fnins.2013.00267

Hill, N. J., Gupta, D., Brunner, P., Gunduz, A., Adamo, M. A., Ritaccio, A., & Schalk, G. (2012). Recording human electrocorticographic (ECoG) signals for neuroscientific research and real-time functional cortical mapping. *Journal of visualized experiments : JoVE*, (64), pubmed 22782131, 3993. https://doi.org/10.3791/3993

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, *6*(2), 65–70. http://www.jstor.org/stable/4615733

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Iturrate, I., Chavarriaga, R., & Millán, J. d. R. (2020). General principles of machine learning for brain-computer interfacing, In *Handbook of clinical neurology*. Elsevier. https://doi.org/10.1016/B978-0-444-63934-9.00023-8

Jeannerod, M. (2001). Neural simulation of action: A unifying mechanism for motor cognition. *NeuroImage*, *14*(1), S103–S109. https://doi.org/10.1006/nimg.2001.0832

Jeannerod, M., & Frak, V. (1999). Mental imaging of motor activity in humans. *Current Opinion in Neurobiology*, *9*(6), 735–739. https://doi.org/10.1016/S0959-4388(99)00038-0

Leeb, R., Perdikis, S., Tonin, L., Biasiucci, A., Tavella, M., Creatura, M., Molina, A., Al-Khodairy, A., Carlson, T., & Millán, J. d. (2013). Transferring brain–computer interfaces beyond the laboratory: Successful application control for motor-disabled users. *Artificial Intelligence in Medicine*, *59*(2), 121–132. https://doi.org/10.1016/j.artmed.2013.08.004

Lotze, M., & Halsband, U. (2006). Motor imagery. *Journal of Physiology-Paris*, *99*(4), 386–395. https://doi.org/10.1016/j.jphysparis.2006.03.012

Munzert, J., Lorey, B., & Zentgraf, K. (2009). Cognitive motor processes: The role of motor imagery in the study of motor representations. *Brain Research Reviews*, *60*(2), 306–326. https://doi.org/10.1016/j.brainresrev.2008.12.024

Niedermeyer, 1., Ernst, Lopes da Silva, 1., F. H., & Ovid Technologies, I. (2005). *Electroencephalography : Basic principles, clinical applications, and related fields* (5th ed) [Includes bibliographical references and index]. Philadelphia ; London : Lippincott Williams & Wilkins.

Oliphant, T. (2006). NumPy: A guide to NumPy [[Online; accessed <today>]]. http://www.numpy.org/

pandas development team, T. (2020). *Pandas-dev/pandas: Pandas* (Version latest). Zenodo. https://doi.org/10.5281/zenodo.3509134

Patel, B. N. (2012). Efficient classification of data using decision tree. *Bonfring International Journal of Data Mining*, *2*(1), 06–12. https://doi.org/10.9756/BIJDM.1098

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Peng, C.-Y. J., Lee, K. L., & Ingersoll, G. M. (2002). An introduction to logistic regression analysis and reporting. *The Journal of Educational Research*, *96*(1), 3–14. https://doi.org/10.1080/00220670209598786

Pfurtscheller, G., Neuper, C., Brunner, C., & da Silva, F. L. (2005). Beta rebound after different types of motor imagery in man. *Neuroscience Letters*, *378*(3), 156–159. https://doi.org/10.1016/j.neulet.2004.12.034

Ramoser, H., Muller-Gerking, J., & Pfurtscheller, G. (2000). Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Transactions on Rehabilitation Engineering*, *8*(4), 441–446. https://doi.org/10.1109/86.895946

Raschka, S. (2017). Naive bayes and text classification i - introduction and theory. *arXiv:1410.5329 [cs]*, arxiv 1410.5329. Retrieved May 29, 2020, from http://arxiv.org/abs/1410.5329

Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-validation (L. LIU & M. T. ÖZSU, Eds.). In L. LIU & M. T. ÖZSU (Eds.), *Encyclopedia of database systems*. Boston, MA, Springer US. https://doi.org/10.1007/978-0-387-39940-9_565

Schalk, G., McFarland, D., Hinterberger, T., Birbaumer, N., & Wolpaw, J. (2004). BCI2000: A general-purpose brain-computer interface (BCI) system. *IEEE Transactions on Biomedical Engineering*, *51*(6), 1034–1043. https://doi.org/10.1109/TBME.2004.827072

Selesnick, I. W., & Burrus, C. S. (1997). Generalized digital butterworth filter design. *IEEE Transactions on Signal Processing*, 18.

Tharwat, A., Gaber, T., Ibrahim, A., & Hassanien, A. E. (2017). Linear discriminant analysis: A detailed tutorial. *AI Communications*, *30*(2), 169–190. https://doi.org/10.3233/AIC-170729

van Rossum, G. (1995). *Python tutorial* (tech. rep. CS-R9526). Centrum voor Wiskunde en Informatica (CWI). Amsterdam.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Jarrod Millman, K., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … Contributors, S. 1. 0. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/https://doi.org/10.1038/s41592-019-0686-2

Xygonakis, I., Athanasiou, A., Pandria, N., Kugiumtzis, D., & Bamidis, P. D. (2018). Decoding motor imagery through common spatial pattern filters at the EEG source space. *Computational Intelligence and Neuroscience*, *2018*, 1–10. https://doi.org/10.1155/2018/7957408

Zhang, Z. (2016). Introduction to machine learning: K-nearest neighbors. *Annals of Translational Medicine*, *4*(11), 218–218. https://doi.org/10.21037/atm.2016.03.37

Table 1

Table showing the task structure implemented by Schalk et al. (2004).

Run	Task
1	Baseline, eyes open
2	Baseline, eyes closed
3, 7, 11	Motor execution: left vs right hand
4, 8, 12	Motor imagery: left vs right hand
5, 9, 13	Motor execution: hands vs feet
6, 10, 14	Motor imagery: hands vs feet

Table 2

Multiple Comparison for 10fold Cross-Validation: Holm–Bonferroni method.

classifier 1	classifier 2	stat	p-value	p-value corrected	reject H0
DTL	NB	-4.3728	0.0	0.0003	True
DTL	KN	-4.6165	0.0	0.0001	True
DTL	LDA	-5.6094	0.0	0.0	True
DTL	LR	-5.1668	0.0	0.0	True
DTL	SVC	-4.8381	0.0	0.0001	True
NB	KN	-1.4799	0.1418	0.9927	False
NB	LDA	-1.5587	0.122	0.976	False
NB	LR	-2.5364	0.0126	0.1263	False
NB	SVC	-1.8583	0.0658	0.5926	False
KN	LDA	-0.1829	0.8552	1.0	False
KN	LR	-1.1324	0.26	1.0	False
KN	SVC	-0.3551	0.7232	1.0	False
LDA	LR	-0.5421	0.5888	1.0	False
LDA	SVC	-0.0503	0.96	1.0	False
LR	SVC	0.9092	0.3653	1.0	False

Note. Corrected alpha = 0.003.

Table 3

Multiple Comparison for Monte Carlo Cross-Validation: Holm–Bonferroni method.

classifier 1	classifier 2	stat	p-value	p-value corrected	reject H0
DTL	NB	-0.1668	0.8679	1.0	False
DTL	KN	-2.4806	0.0147	0.1319	False
DTL	LDA	-3.0027	0.0033	0.0399	True
DTL	LR	-2.0206	0.0458	0.3663	False
DTL	SVC	-2.8937	0.0046	0.0507	False
NB	KN	-3.0639	0.0028	0.0359	True
NB	LDA	-3.5651	0.0005	0.0076	True
NB	LR	-2.8845	0.0047	0.0507	False
NB	SVC	-3.9467	0.0001	0.0021	True
kn	LDA	-0.2341	0.8153	1.0	False
KN	LR	0.8753	0.3833	1.0	False
KN	SVC	-0.6661	0.5068	1.0	False
LDA	LR	1.1387	0.2573	1.0	False
LDA	SVC	-0.3271	0.7442	1.0	False
LR	SVC	-1.6787	0.0961	0.6727	False

Note. Corrected alpha = 0.003.

Figure 1

An illustrated BCI-system process showing the methods used in this paper.

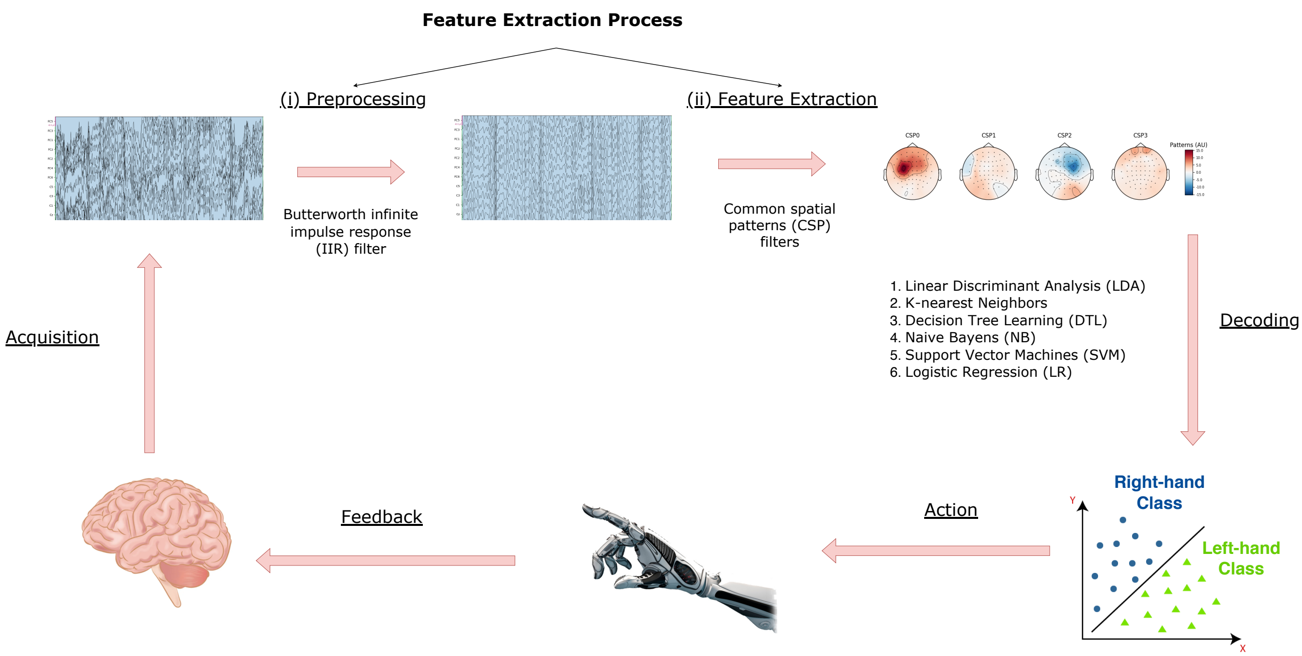


Figure 2

Across subject classifier performance accuracy.

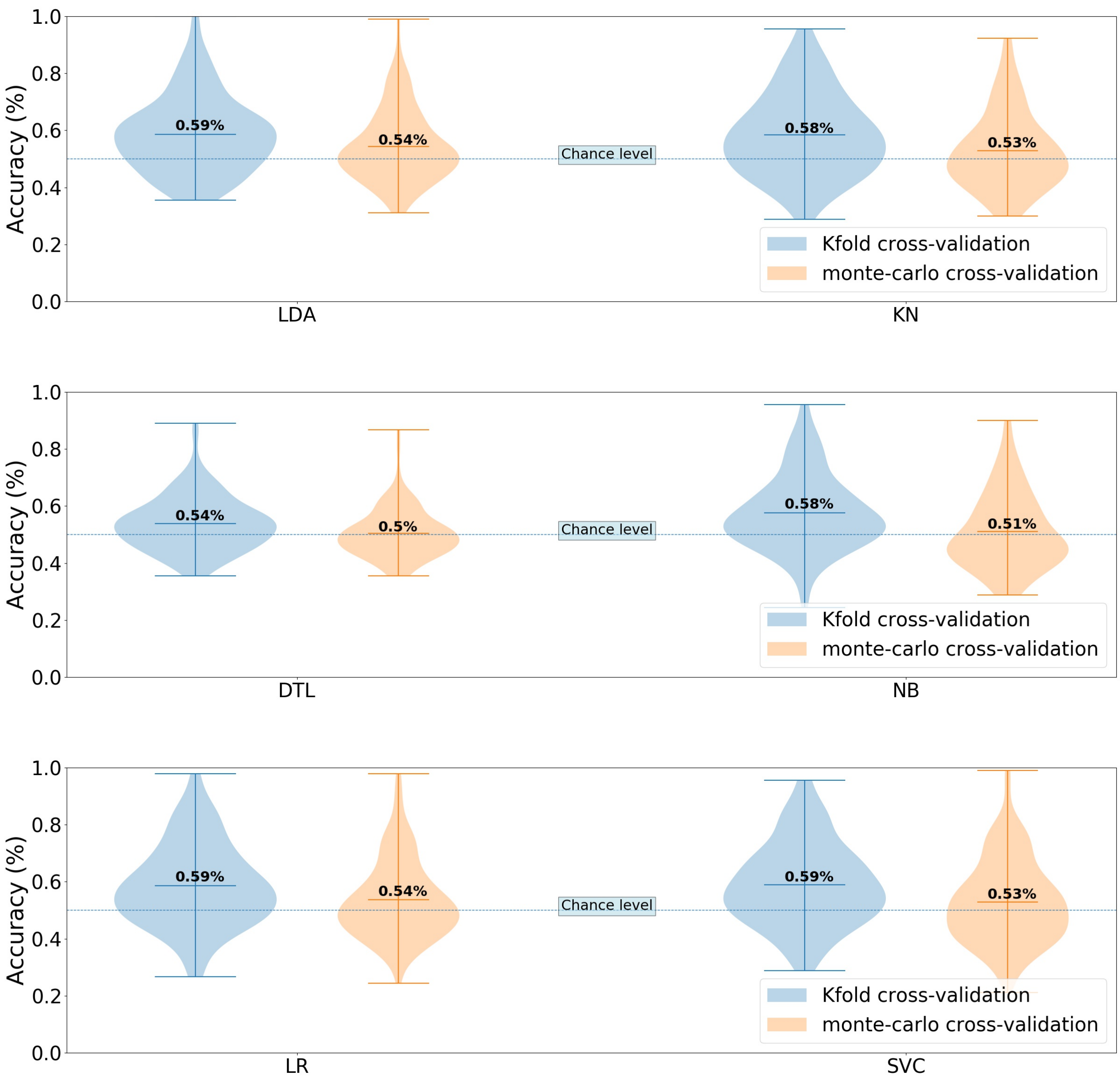


Figure 3

Confusion Matrices: 10-fold Cross-Validation. The Matthews Correlation Coefficient (MCC), a measure of quality of binary (two-class) classification (worst value: –1; best value: +1) , is shown at the top of each matrix.

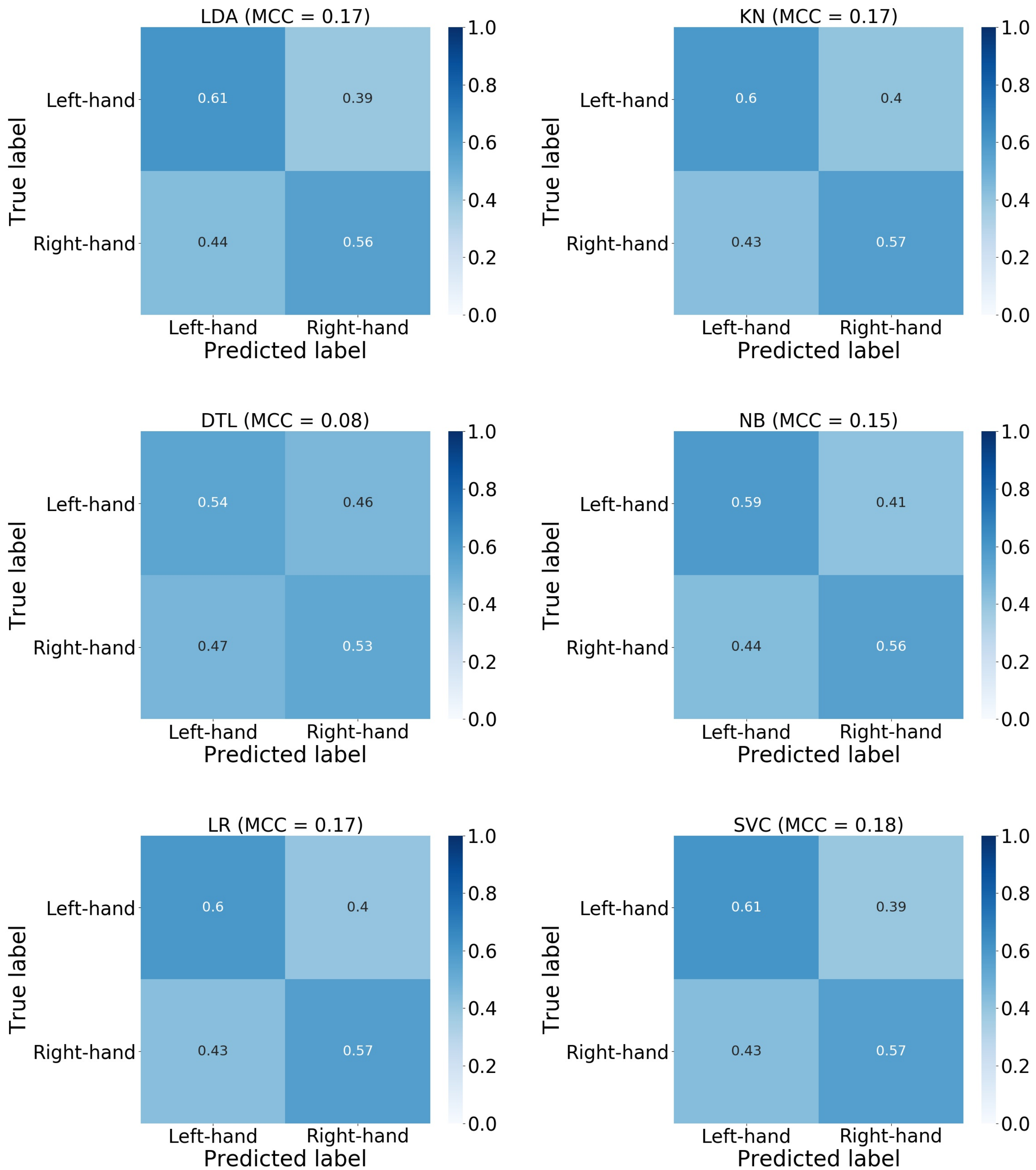


Figure 4

Confusion Matrices: Monte-Carlo Cross-Validation. The Matthews Correlation Coefficient (MCC), a measure of quality of binary (two-class) classification (worst value: –1; best value: +1) , is shown at the top of each matrix.

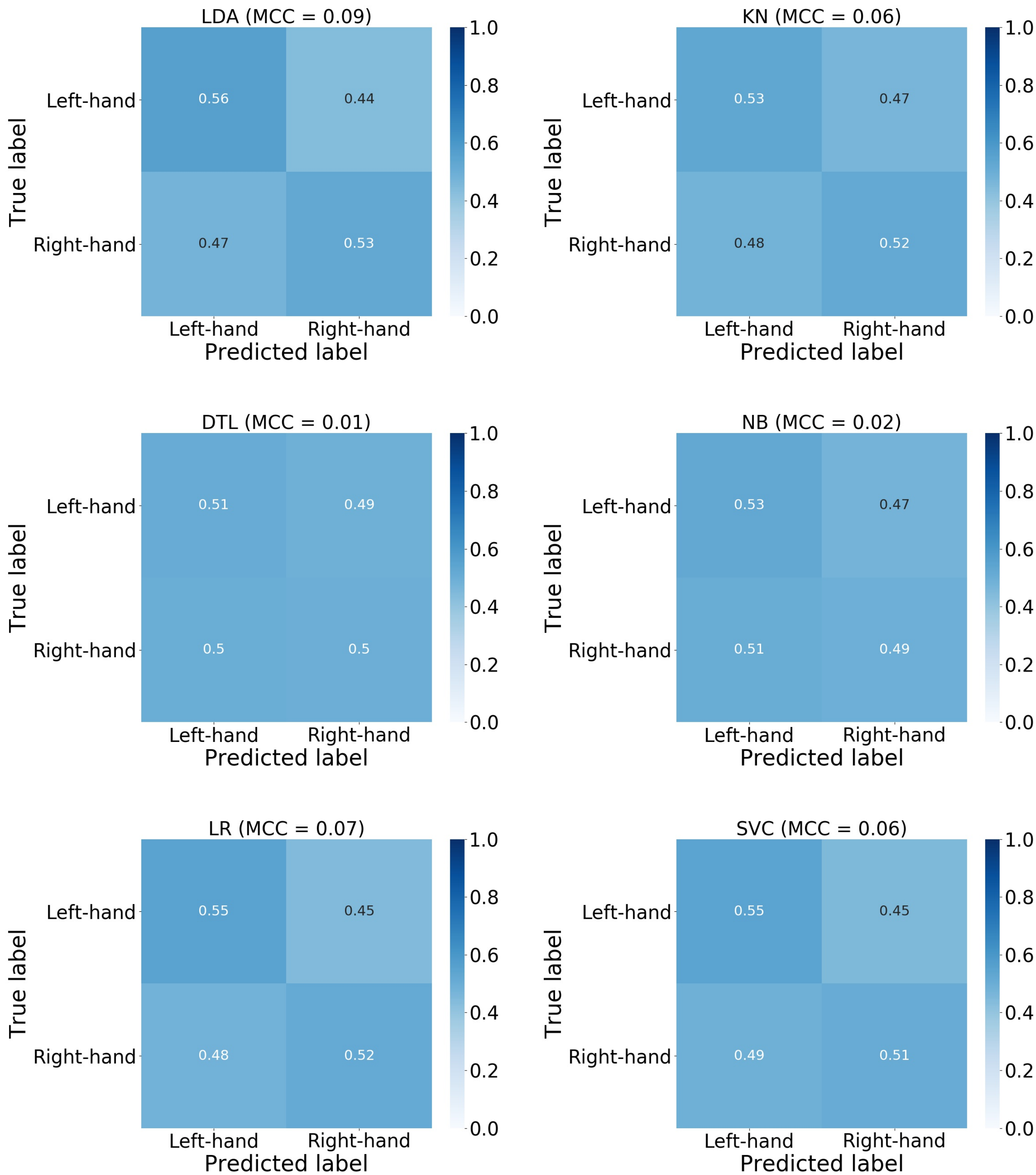
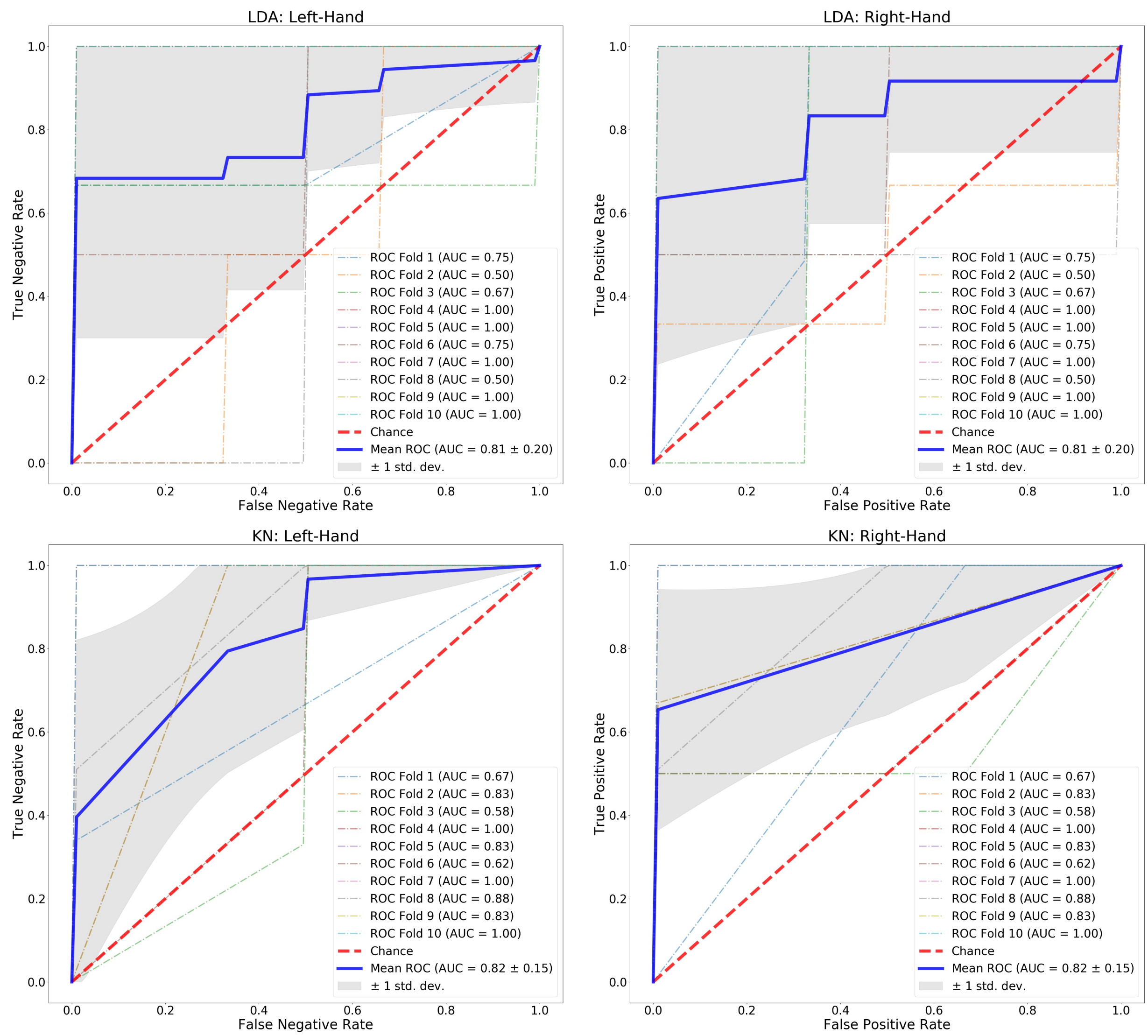


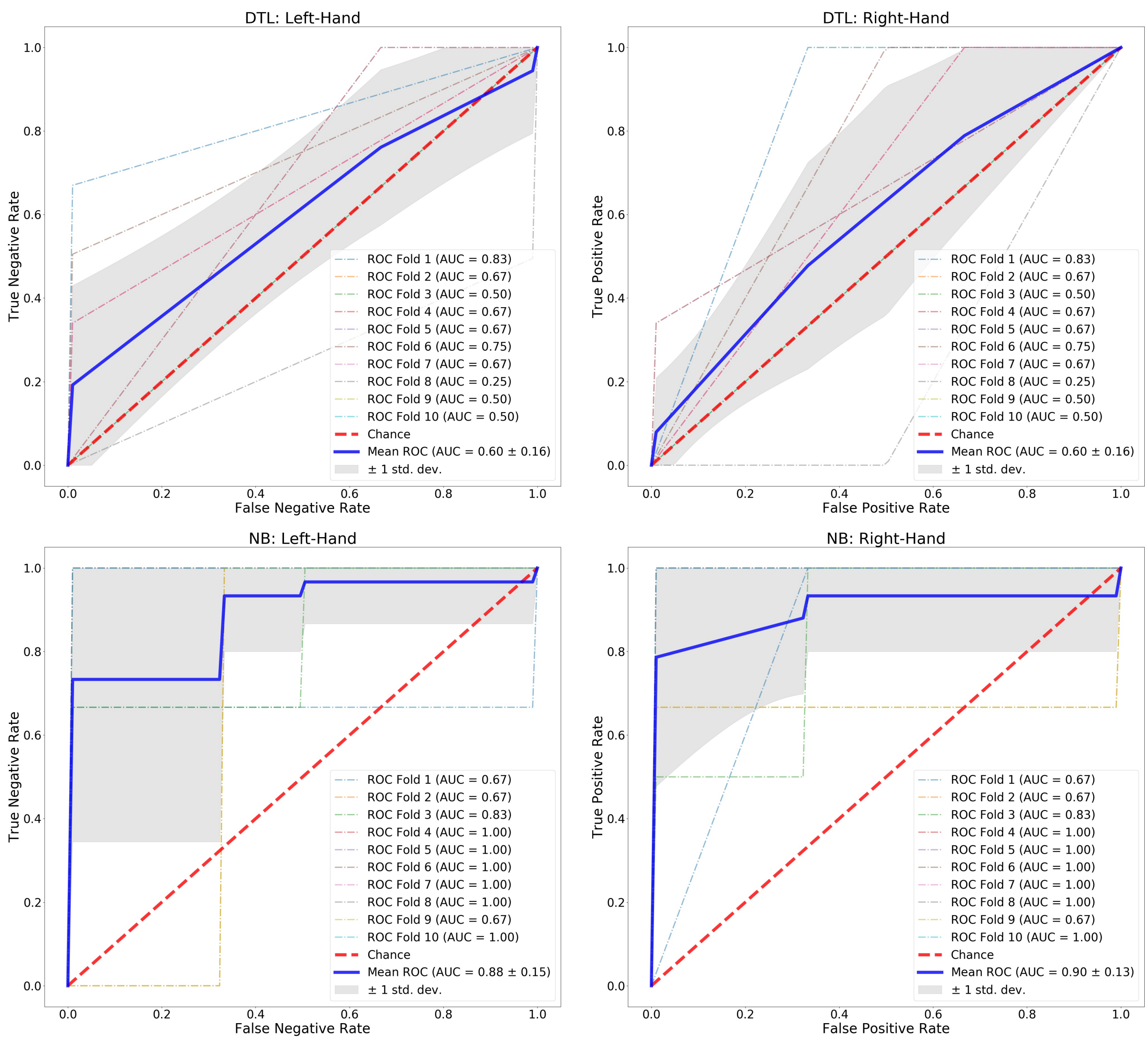
Figure 5

Receiver Operating Characteristics (ROC) curve comparing classifier performance for subject 13 using 10-fold cross-validation.

(a)



(b)


DTL: Left-Hand
True Negative Rate
False Negative Rate
ROC Fold 1 (AUC = 0.83)
ROC Fold 2 (AUC = 0.67)
ROC Fold 3 (AUC = 0.50)
ROC Fold 4 (AUC = 0.67)
ROC Fold 5 (AUC = 0.67)
ROC Fold 6 (AUC = 0.75)
ROC Fold 7 (AUC = 0.67)
ROC Fold 8 (AUC = 0.25)
ROC Fold 9 (AUC = 0.50)
ROC Fold 10 (AUC = 0.50)
Chance
Mean ROC (AUC = 0.60 ± 0.16)
± 1 std. dev.
DTL: Right-Hand
True Positive Rate
False Positive Rate
ROC Fold 1 (AUC = 0.83)
ROC Fold 2 (AUC = 0.67)
ROC Fold 3 (AUC = 0.50)
ROC Fold 4 (AUC = 0.67)
ROC Fold 5 (AUC = 0.67)
ROC Fold 6 (AUC = 0.75)
ROC Fold 7 (AUC = 0.67)
ROC Fold 8 (AUC = 0.25)
ROC Fold 9 (AUC = 0.50)
ROC Fold 10 (AUC = 0.50)
Chance
Mean ROC (AUC = 0.60 ± 0.16)
± 1 std. dev.
NB: Left-Hand
True Negative Rate
False Negative Rate
ROC Fold 1 (AUC = 0.67)
ROC Fold 2 (AUC = 0.67)
ROC Fold 3 (AUC = 0.83)
ROC Fold 4 (AUC = 1.00)
ROC Fold 5 (AUC = 1.00)
ROC Fold 6 (AUC = 1.00)
ROC Fold 7 (AUC = 1.00)
ROC Fold 8 (AUC = 1.00)
ROC Fold 9 (AUC = 0.67)
ROC Fold 10 (AUC = 1.00)
Chance
Mean ROC (AUC = 0.88 ± 0.15)
± 1 std. dev.
NB: Right-Hand
True Positive Rate
False Positive Rate
ROC Fold 1 (AUC = 0.67)
ROC Fold 2 (AUC = 0.67)
ROC Fold 3 (AUC = 0.83)
ROC Fold 4 (AUC = 1.00)
ROC Fold 5 (AUC = 1.00)
ROC Fold 6 (AUC = 1.00)
ROC Fold 7 (AUC = 1.00)
ROC Fold 8 (AUC = 1.00)
ROC Fold 9 (AUC = 0.67)
ROC Fold 10 (AUC = 1.00)
Chance
Mean ROC (AUC = 0.90 ± 0.13)
± 1 std. dev.

(c)

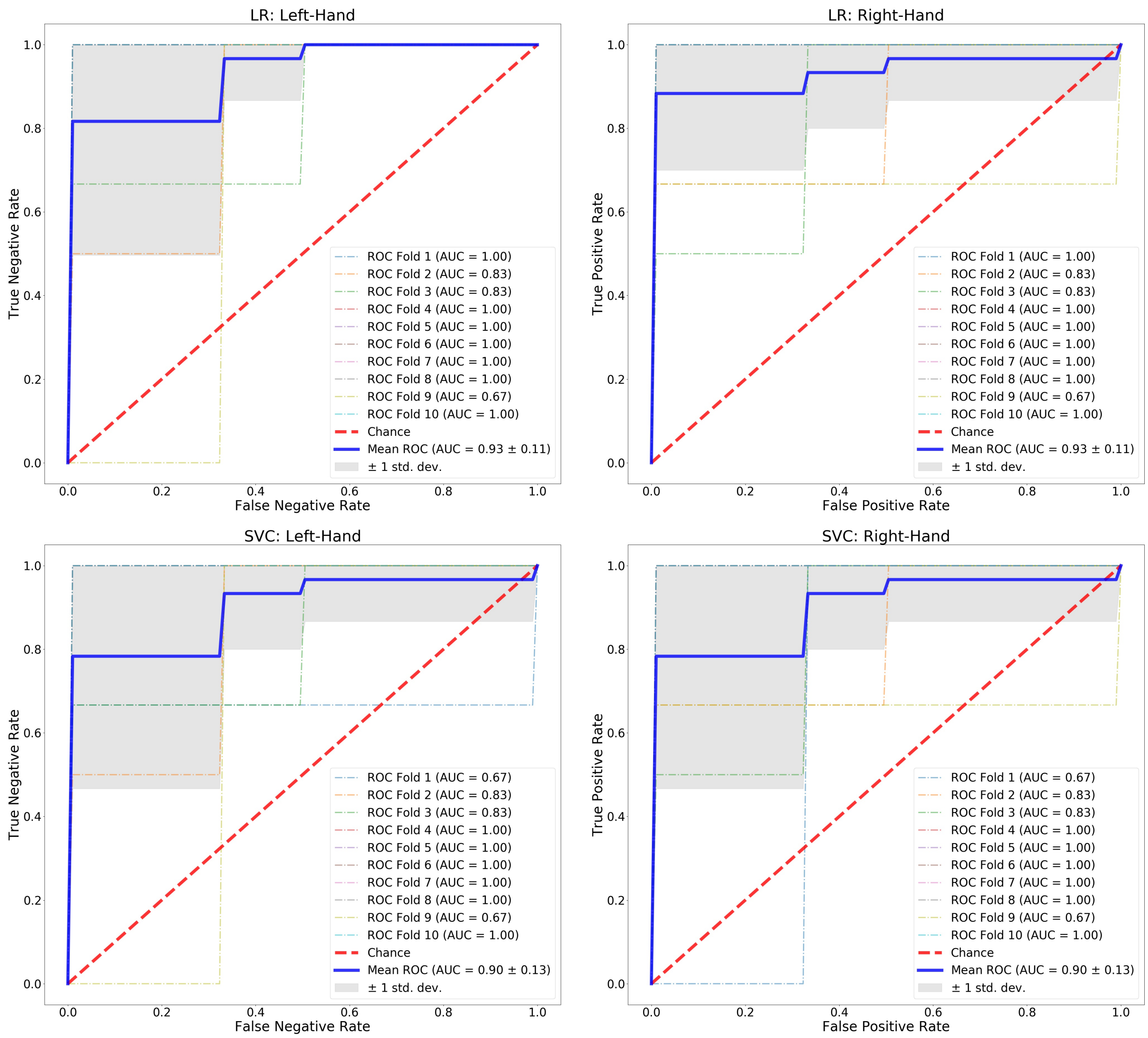
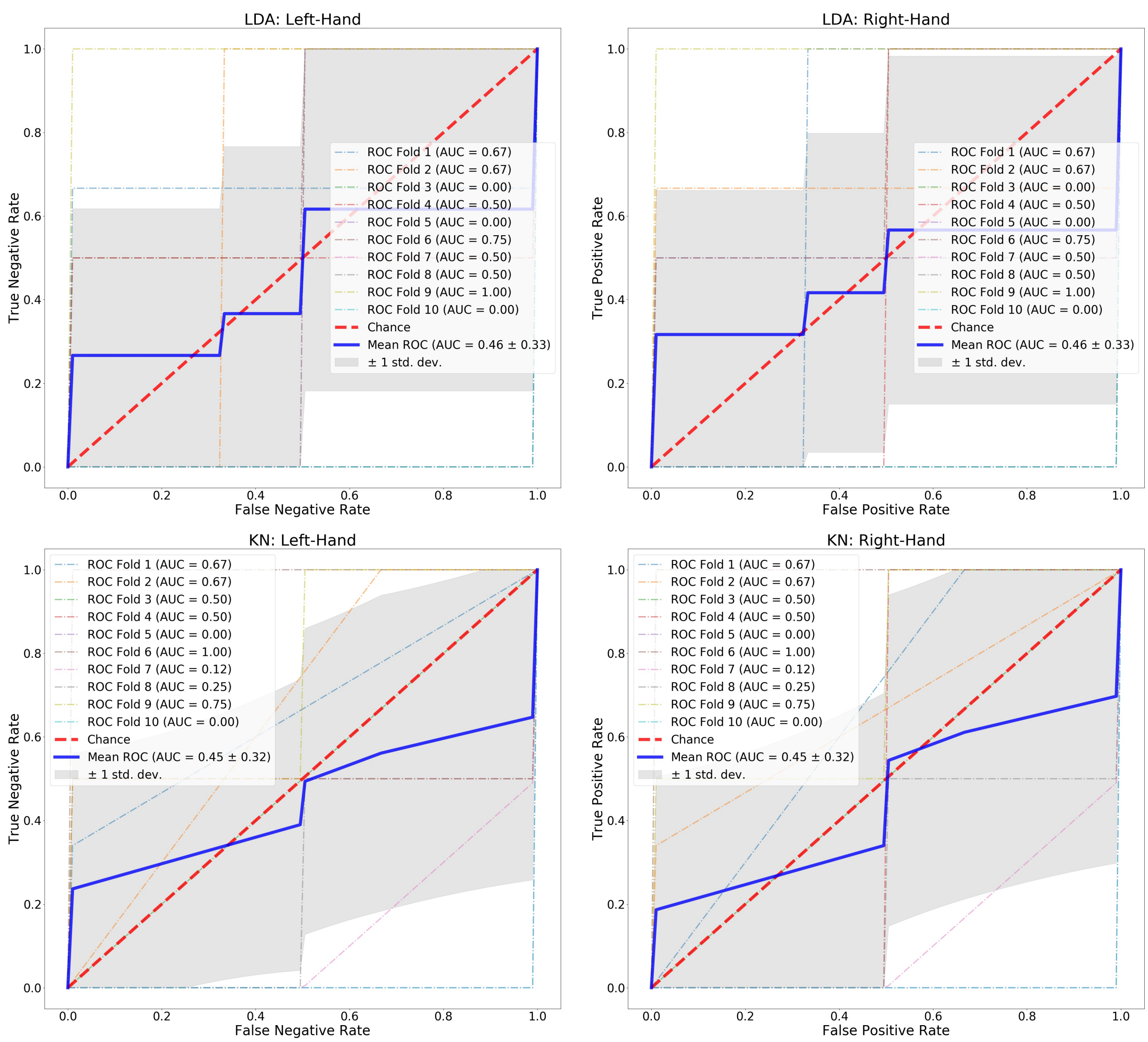

LR: Left-Hand
True Negative Rate
False Negative Rate
ROC Fold 1 (AUC = 1.00)
ROC Fold 2 (AUC = 0.83)
ROC Fold 3 (AUC = 0.83)
ROC Fold 4 (AUC = 1.00)
ROC Fold 5 (AUC = 1.00)
ROC Fold 6 (AUC = 1.00)
ROC Fold 7 (AUC = 1.00)
ROC Fold 8 (AUC = 1.00)
ROC Fold 9 (AUC = 0.67)
ROC Fold 10 (AUC = 1.00)
Chance
Mean ROC (AUC = 0.93 ± 0.11)
± 1 std. dev.
LR: Right-Hand
True Positive Rate
False Positive Rate
ROC Fold 1 (AUC = 1.00)
ROC Fold 2 (AUC = 0.83)
ROC Fold 3 (AUC = 0.83)
ROC Fold 4 (AUC = 1.00)
ROC Fold 5 (AUC = 1.00)
ROC Fold 6 (AUC = 1.00)
ROC Fold 7 (AUC = 1.00)
ROC Fold 8 (AUC = 1.00)
ROC Fold 9 (AUC = 0.67)
ROC Fold 10 (AUC = 1.00)
Chance
Mean ROC (AUC = 0.93 ± 0.11)
± 1 std. dev.
SVC: Left-Hand
True Negative Rate
False Negative Rate
ROC Fold 1 (AUC = 0.67)
ROC Fold 2 (AUC = 0.83)
ROC Fold 3 (AUC = 0.83)
ROC Fold 4 (AUC = 1.00)
ROC Fold 5 (AUC = 1.00)
ROC Fold 6 (AUC = 1.00)
ROC Fold 7 (AUC = 1.00)
ROC Fold 8 (AUC = 1.00)
ROC Fold 9 (AUC = 0.67)
ROC Fold 10 (AUC = 1.00)
Chance
Mean ROC (AUC = 0.90 ± 0.13)
± 1 std. dev.
SVC: Right-Hand
True Positive Rate
False Positive Rate
ROC Fold 1 (AUC = 0.67)
ROC Fold 2 (AUC = 0.83)
ROC Fold 3 (AUC = 0.83)
ROC Fold 4 (AUC = 1.00)
ROC Fold 5 (AUC = 1.00)
ROC Fold 6 (AUC = 1.00)
ROC Fold 7 (AUC = 1.00)
ROC Fold 8 (AUC = 1.00)
ROC Fold 9 (AUC = 0.67)
ROC Fold 10 (AUC = 1.00)
Chance
Mean ROC (AUC = 0.90 ± 0.13)
± 1 std. dev.

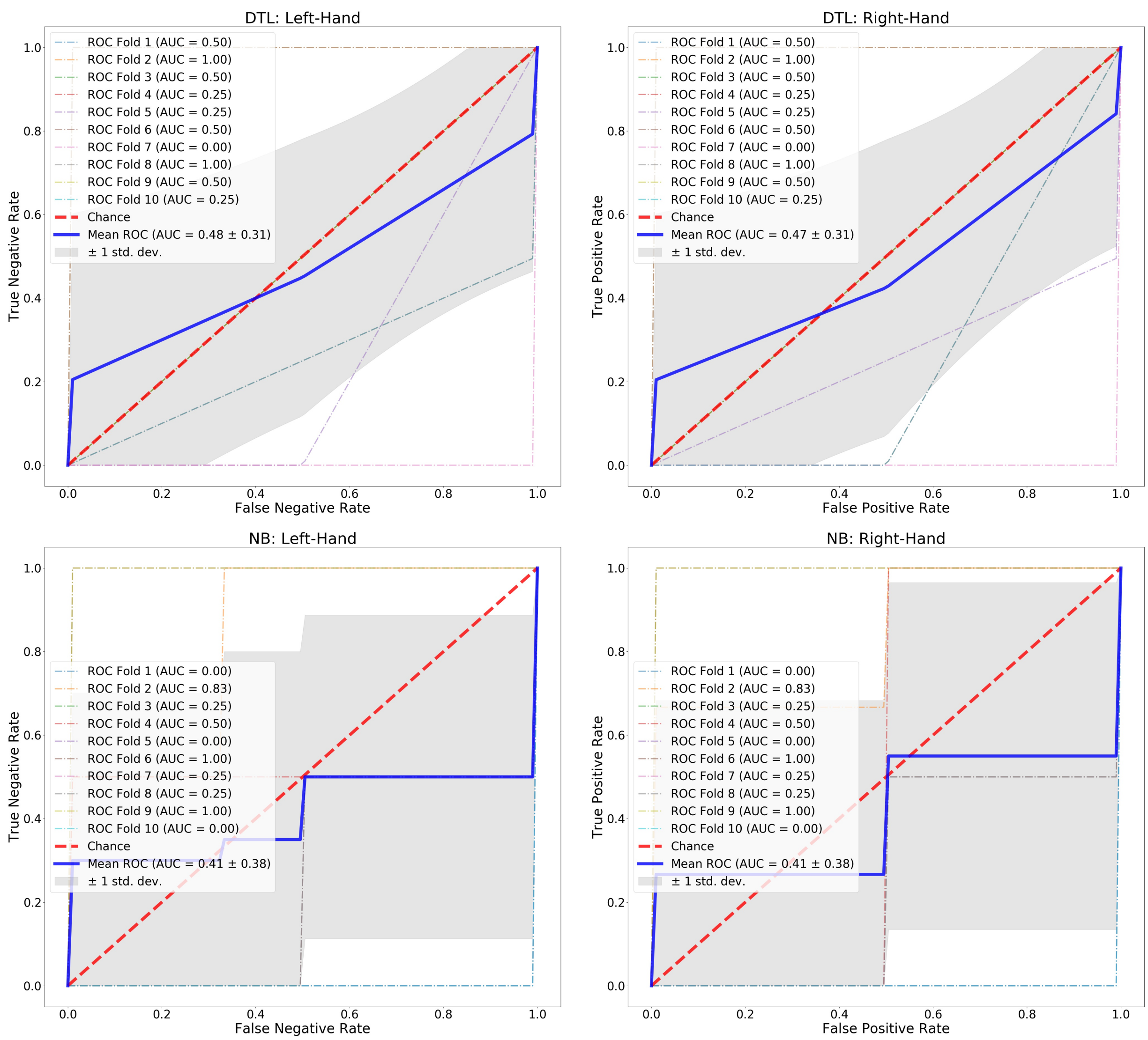
Figure 6

Receiver Operating Characteristics (ROC) curve comparing classifier performance for subject 73 using 10-fold cross-validation.

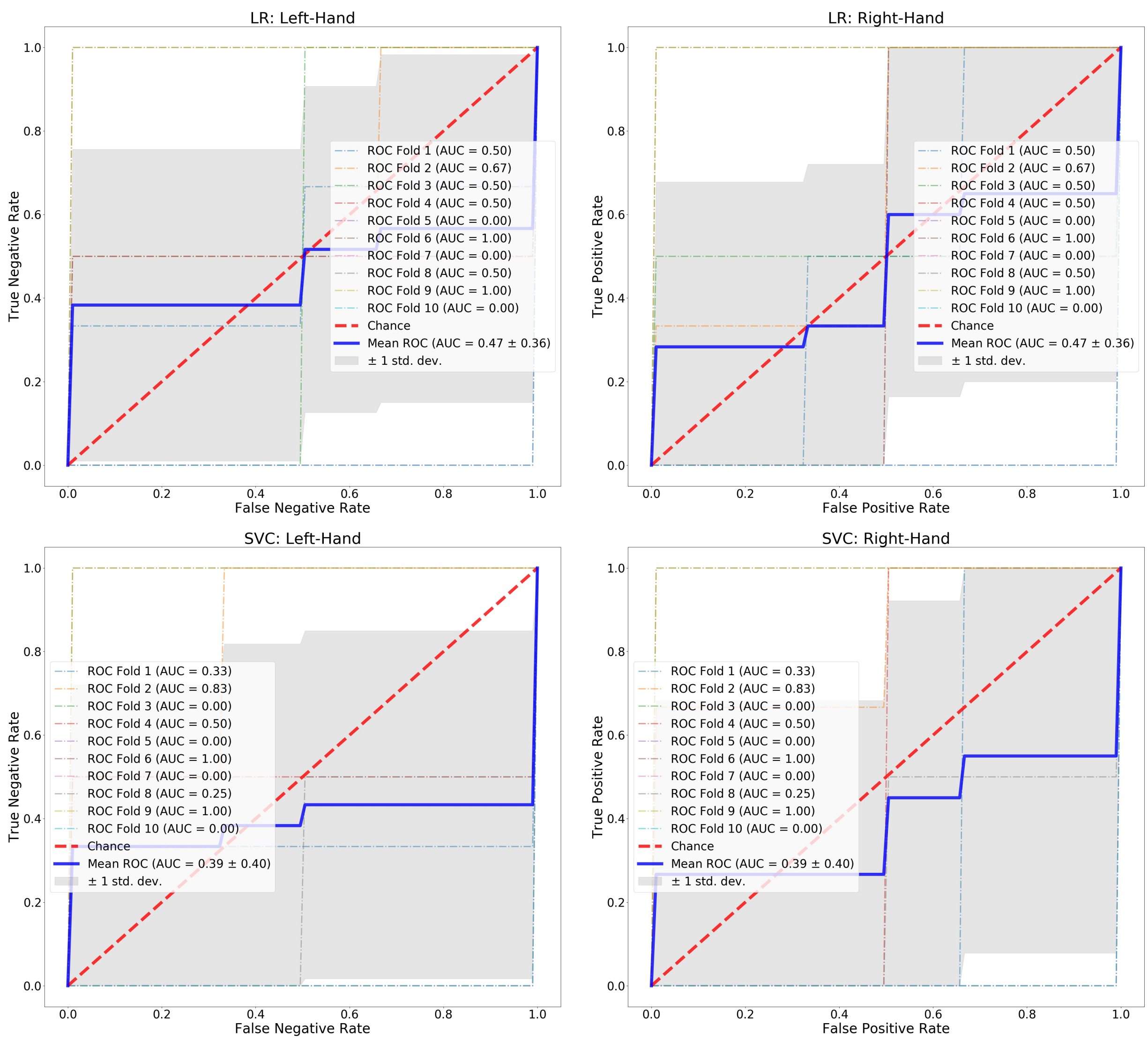
(a)



(b)


DTL: Left-Hand
ROC Fold 1 (AUC = 0.50)
ROC Fold 2 (AUC = 1.00)
ROC Fold 3 (AUC = 0.50)
ROC Fold 4 (AUC = 0.25)
ROC Fold 5 (AUC = 0.25)
ROC Fold 6 (AUC = 0.50)
ROC Fold 7 (AUC = 0.00)
ROC Fold 8 (AUC = 1.00)
ROC Fold 9 (AUC = 0.50)
ROC Fold 10 (AUC = 0.25)
Chance
Mean ROC (AUC = 0.48 ± 0.31)
± 1 std. dev.
True Negative Rate
False Negative Rate
DTL: Right-Hand
ROC Fold 1 (AUC = 0.50)
ROC Fold 2 (AUC = 1.00)
ROC Fold 3 (AUC = 0.50)
ROC Fold 4 (AUC = 0.25)
ROC Fold 5 (AUC = 0.25)
ROC Fold 6 (AUC = 0.50)
ROC Fold 7 (AUC = 0.00)
ROC Fold 8 (AUC = 1.00)
ROC Fold 9 (AUC = 0.50)
ROC Fold 10 (AUC = 0.25)
Chance
Mean ROC (AUC = 0.47 ± 0.31)
± 1 std. dev.
True Positive Rate
False Positive Rate
NB: Left-Hand
ROC Fold 1 (AUC = 0.00)
ROC Fold 2 (AUC = 0.83)
ROC Fold 3 (AUC = 0.25)
ROC Fold 4 (AUC = 0.50)
ROC Fold 5 (AUC = 0.00)
ROC Fold 6 (AUC = 1.00)
ROC Fold 7 (AUC = 0.25)
ROC Fold 8 (AUC = 0.25)
ROC Fold 9 (AUC = 1.00)
ROC Fold 10 (AUC = 0.00)
Chance
Mean ROC (AUC = 0.41 ± 0.38)
± 1 std. dev.
True Negative Rate
False Negative Rate
NB: Right-Hand
ROC Fold 1 (AUC = 0.00)
ROC Fold 2 (AUC = 0.83)
ROC Fold 3 (AUC = 0.25)
ROC Fold 4 (AUC = 0.50)
ROC Fold 5 (AUC = 0.00)
ROC Fold 6 (AUC = 1.00)
ROC Fold 7 (AUC = 0.25)
ROC Fold 8 (AUC = 0.25)
ROC Fold 9 (AUC = 1.00)
ROC Fold 10 (AUC = 0.00)
Chance
Mean ROC (AUC = 0.41 ± 0.38)
± 1 std. dev.
True Positive Rate
False Positive Rate

(c)


LR: Left-Hand
True Negative Rate
False Negative Rate
ROC Fold 1 (AUC = 0.50)
ROC Fold 2 (AUC = 0.67)
ROC Fold 3 (AUC = 0.50)
ROC Fold 4 (AUC = 0.50)
ROC Fold 5 (AUC = 0.00)
ROC Fold 6 (AUC = 1.00)
ROC Fold 7 (AUC = 0.00)
ROC Fold 8 (AUC = 0.50)
ROC Fold 9 (AUC = 1.00)
ROC Fold 10 (AUC = 0.00)
Chance
Mean ROC (AUC = 0.47 ± 0.36)
± 1 std. dev.
LR: Right-Hand
True Positive Rate
False Positive Rate
ROC Fold 1 (AUC = 0.50)
ROC Fold 2 (AUC = 0.67)
ROC Fold 3 (AUC = 0.50)
ROC Fold 4 (AUC = 0.50)
ROC Fold 5 (AUC = 0.00)
ROC Fold 6 (AUC = 1.00)
ROC Fold 7 (AUC = 0.00)
ROC Fold 8 (AUC = 0.50)
ROC Fold 9 (AUC = 1.00)
ROC Fold 10 (AUC = 0.00)
Chance
Mean ROC (AUC = 0.47 ± 0.36)
± 1 std. dev.
SVC: Left-Hand
True Negative Rate
False Negative Rate
ROC Fold 1 (AUC = 0.33)
ROC Fold 2 (AUC = 0.83)
ROC Fold 3 (AUC = 0.00)
ROC Fold 4 (AUC = 0.50)
ROC Fold 5 (AUC = 0.00)
ROC Fold 6 (AUC = 1.00)
ROC Fold 7 (AUC = 0.00)
ROC Fold 8 (AUC = 0.25)
ROC Fold 9 (AUC = 1.00)
ROC Fold 10 (AUC = 0.00)
Chance
Mean ROC (AUC = 0.39 ± 0.40)
± 1 std. dev.
SVC: Right-Hand
True Positive Rate
False Positive Rate
ROC Fold 1 (AUC = 0.33)
ROC Fold 2 (AUC = 0.83)
ROC Fold 3 (AUC = 0.00)
ROC Fold 4 (AUC = 0.50)
ROC Fold 5 (AUC = 0.00)
ROC Fold 6 (AUC = 1.00)
ROC Fold 7 (AUC = 0.00)
ROC Fold 8 (AUC = 0.25)
ROC Fold 9 (AUC = 1.00)
ROC Fold 10 (AUC = 0.00)
Chance
Mean ROC (AUC = 0.39 ± 0.40)
± 1 std. dev.