Bagging Adversarial Neural Networks for Domain Adaptation in Non-Stationary EEG

Haider Raza and Spyridon Samothrakis
Institute for Analytics and Data Science,
School of Computer Science and Electronics Engineering, University of Essex,
Colchester, England, UK

***Abstract*—A major issue in bringing real-world applications of machine learning outside the laboratory is the difference in the data distributions between training and testing stages or domains. The diverging statistical properties in different domains can lead to decay the prediction performance. The technical term for a change in the distribution of features is covariate shift, which also happens to be a common challenge in electroencephalogram (EEG) based brain-computer interface (BCI); this is due to the presence of non-stationarities in the EEG signals. It is also the case that collecting and labelling samples is expensive, resulting in small datasets that are not in tune with the "big data" spirit that is the characteristic of the era. In this paper, we introduce a new method that handles domain adaptation in small datasets; the method combines elements of unsupervised domain adaptation with ensemble methods. We evaluate on real-world datasets corresponding to motor-imagery detection (BCI competition 2008 dataset 2A). The method produces state of the art results.**

I. Introduction

Machine learning methods have attained significant progress in many knowledge mining areas including classification, regression, and clustering. However, it is still challenging to bring these technologies outside the laboratory due to the divergence in the data distributions between training and testing stages (domains). A common assumption in machine learning is that the training and testing data are drawn from the same distribution or feature space. However, this assumption is often violated - we often operate in non-stationary environments. The shift in the joint distribution between training and testing domains is known as a dataset shift [1]. Under the condition of dataset shift, the statistical models need to be retrained on newly collected data. In many cases of real-world application, it is almost uneconomical or impossible to retrain the model or collect labels for them. In such cases, dataset shift adaptation or domain adaptation would be a possible solution.

Domain adaptation in predictive modelling is sometimes beneficial, an example is single-trial electroencephalography (EEG) classification in Brain-Computer Interface (BCI) system, where our goal is to classify a motor-imagery (MI) task in predefined classes. The BCI is an alternative communication's means, which allows a user to express his will without muscle exertion, provided that the brain signals are properly translated into computer commands [2], [3]. The non-stationarities in the EEG is a common challenge and it may be caused by various reasons such as changing user attention level, electrode placement, and user fatigue [4], [5], [6]. There are therefore notable variations or shifts in the EEG signals during trial-to-trial and session-to-session transfers. These variations often appear as covariate shifts, wherein the input data distributions differ significantly between training/calibration and testing/operating phases, while the conditional distribution remains the same [6], [7]. With EEG-based BCI system that operates online, it is required to consider input features that are invariant to shifts of the data, or learning approaches that can be able to track the changes that can repeat overtime, to update the classifier in a timely fashion. The low classification accuracy has been one of the main concerns of the developed BCI systems based on MI detection, which directly affects the reliability of the BCI decision-making process and the information transfer rate. To enhance BCI performance, several feature extraction, feature selection, and feature classification techniques have been proposed in the literature [8], [9], [10]. A large variety of features have been used in BCI based on MI detection such as band powers, power spectral density, time-frequency features, and common spatial patterns (CSP) based features [6], [8], [9], [11]. However, the EEG non-stationary characteristics result in shifts in feature distributions for all types of features in varying degrees. An example showing the change of distribution between the training and test datasets is depicted in Fig. 1. To solve this issue, adaptive learning and domain adaptation algorithms have been proposed for devising adaptive BCI systems with positive results [12], [13], [14]. Most of which have made efforts to reduce the non-stationarity aspect in the extracted features.

To overcome the issues discussed above, we employ a mixture of unsupervised domain adaptation methods (borrowed from the neural networks literature) and ensemble methods. The modified version of unsupervised domain adaptation that incorporates "Gan loss" and bagging ("Bootstrap aggregating"). Moreover, we add an additional pathway to the network that tries to predict if the data comes from the training or the test distribution, usually termed the discriminator. The approach has a stabilising effect on the smaller datasets used in EEG studies, helping achieve state of the art results.

The remainder of the paper is organized as follows: section-II gives a background of the problem statement, existing methods, and dataset; section III-B presents the system overview of the proposed domain adaptation method.

The datasets and the signal processing steps are described in Section III-D. The results are then presented in section-IV and discussed in Section-V.

II. Background

A. Problem Statement

We will work within the EEG paradigm. Let us consider a learning framework in which a training dataset is denoted by $X_{Tr} = \{(x_i, y_i)\}_{i=1}^{N}$, where N is the total number of trials, and a label y_i is associated with each trial input features x_i. Depending upon the number of inputs and outputs, x_i and y_i may be a scalar or vector variables. Let us consider a two-class single-trial EEG classification problem i.e., $y \in \{\omega_1, \omega_2\}$. The probability distribution of the inputs at time i can thus be defined as, $P(x_i) = P(\omega_1)P(x_i|\omega_1) + P(\omega_2)P(x_i|\omega_2)$ where $P(\omega_1)$ and $P(\omega_2)$ are the prior probabilities of getting a sample of the classes ω_1 and ω_2, respectively, while $P(x_i|\omega_1)$ and $P(x_i|\omega_2)$ are the conditional probability distribution for the time period i. The goal is to predict the labels $\hat{y}_i$ of the upcoming trials resulting into $X_{Ts} = (\hat{y}_i|x_i)_{i=1}^{M}$, where M is the total number of trials in the testing phase.

B. Covariate shift in EEGs

The following key methods are focused to tackle the issue of domain-adaptation in EEG-based BCI systems:

1) FBCSP: In EEG-based BCI systems, the covariate shift in the training and testing distributions is a common issue [15]. In order to discriminate two classes, an effective approach named common spatial filtering (CSP) was proposed for the spatial filtering, where it maximizes the covariance of one class and minimizes the covariance of the second class. This approach helped BCI systems to work under covariate shift up-to some extent. Later, traditional CSP algorithm has been extended to filter-bank-CSP (FBCSP), where the data were filtered into different frequency range and on each range, a CSP was applied. FBCSP is mainly used nowadays to extract the best discriminative features in the single-trial EEG-classification. However, the covariate shift still exists in the extracted features due to various reasons as discussed in section-I.

2) Covariate Shift-Detection based Supervised Adaptive Learning (CSD-ADL): In CSD-ADL [4], the covariate shift points in the streaming FBCSP features were detected based on an exponential weighted moving average (EWMA) model and in the next step, supervised adaptation was performed on MI-based brain responses. In other words, following the CSD test, the learning initiates adaptation in supervised mode by updating the classifier's feature space at each detected covariate shift. This method is only applicable to the applications, where the labels are present at testing stage such as in the BCI-based neuro-rehabilitation system. [16].

3) Combination of Transductive and Inductive Learning (CTIL): In CTIL [17], a novel hybrid learning method was proposed, which is based on two classifiers, wherein the first classifier allows adding novel information in the training feature space, and the second classifier performs an overall classification. CTIL was developed based on the smoothness assumption in semi-supervised learning, i.e. the points that are closest to each other are more likely to share the same label, and may be added online to enrich the training feature space.

4) Covariate Shift Estimation-based Adaptive Ensemble Learning (CSE-UAEL): In CSE-UAEL [18], an integrative approach was proposed for covariate shift estimation and unsupervised adaptive ensemble learning (CSE-UAEL) to tackle non-stationarity in MI-related EEG classification. Where an exponentially weighted moving average (EWMA) model was used to detect the covariate shifts in the FBCSP-based features extracted from MI related brain responses. Then, an ensemble of classifiers was created and updated over time to handle the shift in streaming input data, where at each covariate shift estimation, a new member was added to the ensemble. The limitations of this method are given as follows: 1) If the input feature vector is high-dimensional then CSE may not be so robust and the possibility of getting false-positives are high, 2) if the CSE gives high false alarms, then an unwanted classifier will be added to the ensemble, which may not bring additional information to the existing feature distribution.

C. Datasets

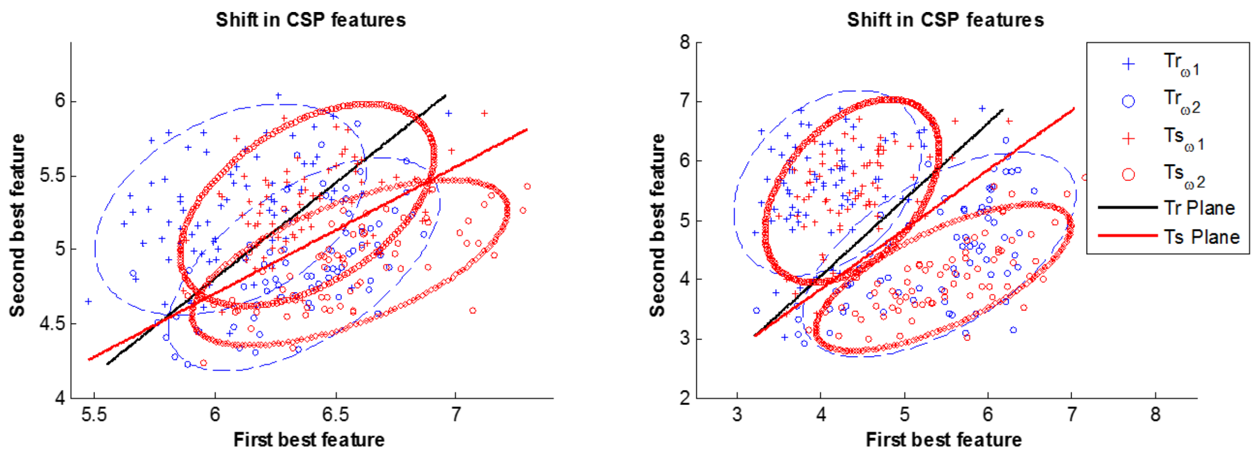
The BCI Competition IV dataset 2A is comprised of EEG data collected from nine subjects that were recorded during two sessions on separate days for each subject [19]. The data consists of 25 channels and includes 22 mono-polar EEG channels, and 3 mono-polar EOG channels with a sampling frequency of 250 Hz. Among the 22 EEG channels, 10 channels are selected for this study, which is responsible for capturing most of the MI related activations: C3, FC3, CP3, C5, C1, C4, FC4, CP4, C2, and C6. Each session consists of six runs separated by short breaks, each run comprised of 48 trials of 7.5 seconds (12 for each class), resulting in a total of 288 trials. Only the classes corresponding to the left hand and right hand were considered in the present study. The MI data from the session I was used to training the classifiers and the MI data from the session-II was used for testing purposes.

III. Proposed Method

In this section, we discuss the method used in this paper; the method comprises of a small neural network, a modified version of unsupervised domain adaptation that incorporates "Gan loss" and bagging ("Bootstrap aggregating"). We will discuss in the details of the implementation below.

A. Neural Networks

Neural Networks [20] are a form of function approximation that has gained tremendous popularity in the form of deep learning [21]. The recent successes of neural networks can be attributed to a set of new methodological insights and hardware improvements that helped tame the vast complexity of the topic. In our case, due to the size of the data, we use



(a) [8-12] Hz

(b) [14-30] Hz

Fig. 1: Covariate shift in the EEG dataset 2A subject A03, between training and testing input distribution for different frequency bands. (a) Mu band [8-12] Hz, and (b) Beta band [14-30]Hz. The red circle denote the features of the left hand motor imagery and blue crosses denote the features of the right hand MI. The black and red lines represent the decision boundaries obtained by the training data and test data respectively.

a simple neural network (see Fig.3); in its simplest form, the network comprises of two layers of hyperbolic tangent (”tanh”) neurons, followed by a softmax output layer. All layers use Batch Normalization [22], while the first layer also has a dropout of 0.5 [23].

B. Unsupervised Domain adaptation

As explained in the previous sections, the train and the test set used in this piece of work come from different distributions. In order to compensate for this, a method termed “unsupervised domain adaptation” [24] is adopted in a modified format. The original method uses a simple - yet efficient - scheme that forces the features learned by the neural network to be statistically indistinguishable between the train and the test datasets. This is achieved by adding an additional pathway to the network that tries to predict if the data comes from the training or the test distribution, usually termed the discriminator. The discriminator tries to differentiate if the network features are part of the training vs the test distribution; the gradient accumulated is reversed and passed to the rest of the network reversed; that is the upper part of the network (the common parts) try are trained as to make it as hard for the lower part of the network (the discriminator) to detect any change. This loss is $\mathcal{L}_{\text{adv}_M} = -\mathcal{L}_{\text{adv}_D}$ is added to the normal loss function. It is quite often however easier to minimise what is termed “GAN loss” [25] $\mathcal{L}_{\text{adv}_M}(X_{Tr}, X_{Ts}, D) = -\mathbb{E}_{X_{Ts}\sim X_{Ts}}[logD(M_{Ts}(X_{Ts}))]$.; this loss (which in practical terms simply means reversing the labels M_{Ts} - with X_{Tr} and X_{Ts} being the training and the test set features sets respectively) is what we use here. We furthermore use Batch normalisation at each layer, which has shown to be beneficial in domain adaptation problems [26]. The GAN loss is modification of the loss used in adversarial networks, thought this time there is the generator part is replaced by the source/test distribution data, while the discriminator part tries to differentiate between them. The unsupervised part in the algorithm comes from the fact that the labels of the testing distribution are not known at any stage.

C. Bagging

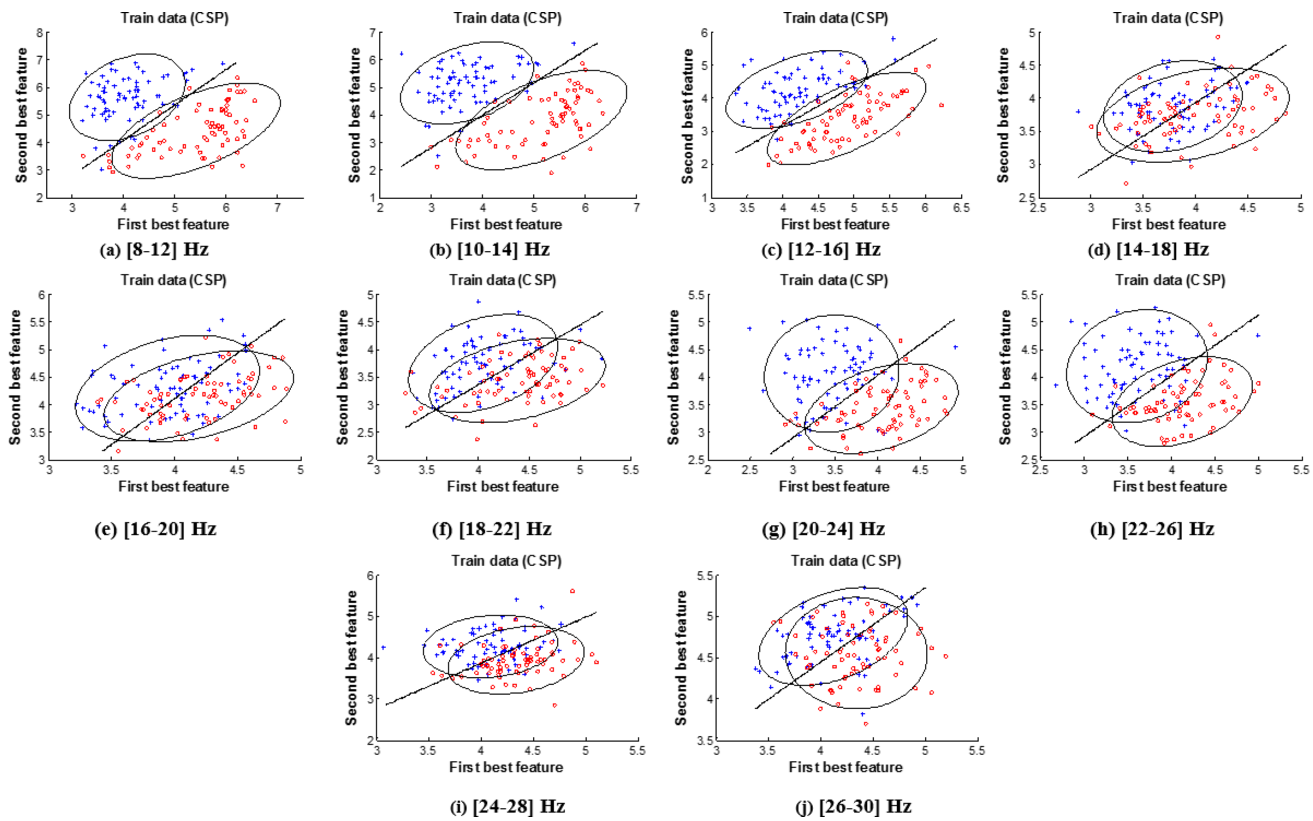
Bootstrap aggregation commonly termed bootstrapping [27] is a method for decreasing the variance of classifiers, without increasing the variance. This is especially important in small datasets, that are very prone to overfitting. The method works by creating h “Bootstraps” of the data, by sampling with replacement. Each Bootstrap set of input-output pairs $(X_b^0, y_b^0)...(X_b^{n-1}, y_b^{n-1})$ is used to train a classifier. In our case, we train the binary neural network presented in Fig.3. When the bootstraps are trained, the outcomes are used as:

As an ensembling method, bagging is mostly used with Trees in random forests - though it has been applied in the past to neural networks[28], however, the long training times make it less popular. The dataset we are using here is relatively small, but training time is still an issue due to implementation inefficiencies.

D. Data Processing and Feature Extraction

The first stage of signal processing employs a filter bank that decomposes the EEG signals into multiple frequency bands [29]. A total of 10 band-pass filters are used, namely [8-12], [10-14], [12-16], [14-18], [16-20], [18-22], [20-24], [22-26], [24-28], [26-30] Hz. This set of frequency bands is used because it covers the expected frequency range of motor imagery response. In the next sections, we consider a time segment of 3 s after the cue onsets.

The second stage employs common spatial filters (CSP), a set of spatial filters that maximize the variance of spatially filtered signals under one condition while minimizing it for



(a) [8-12] Hz (b) [10-14] Hz (c) [12-16] Hz (d) [14-18] Hz

(e) [16-20] Hz (f) [18-22] Hz (g) [20-24] Hz (h) [22-26] Hz

(i) [24-28] Hz (j) [26-30] Hz

Fig. 2: Distribution of the two best features obtained by CSP for Subject A03. The plots (a-j) represent the CSP features for each band. The red circles denote the features of the left hand MI and blue crosses denote the features of the right hand MI. The black line represents the separation plane for illustration purpose only.

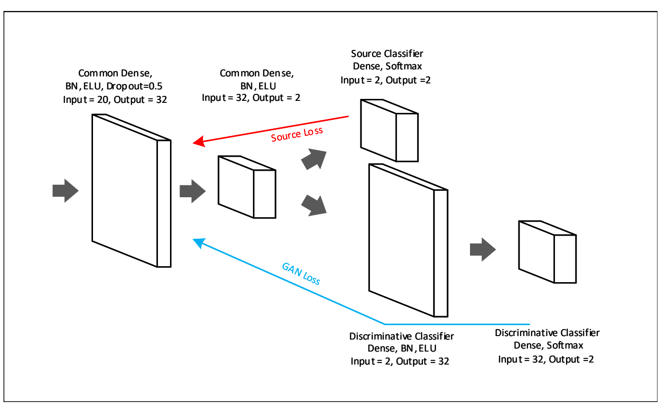


Fig. 3: Depiction of the neural network used. The source classifier (SC) part of the network is the normal binary classifier, while the discriminative classifier (DC) part of the network is tries to discriminate between train and test data. They both share the common layers, where 'BN' is batch normalization, 'ELU' is Exponential Linear Unit, and 'Softmax' is Softmax activation function

the other condition. Raw EEG scalp potentials are known to have poor spatial resolution due to volume conduction. If the signal of interest is weak while other sources produce strong signals in the same frequency range, then it is difficult to classify two classes of EEG measurements [3].

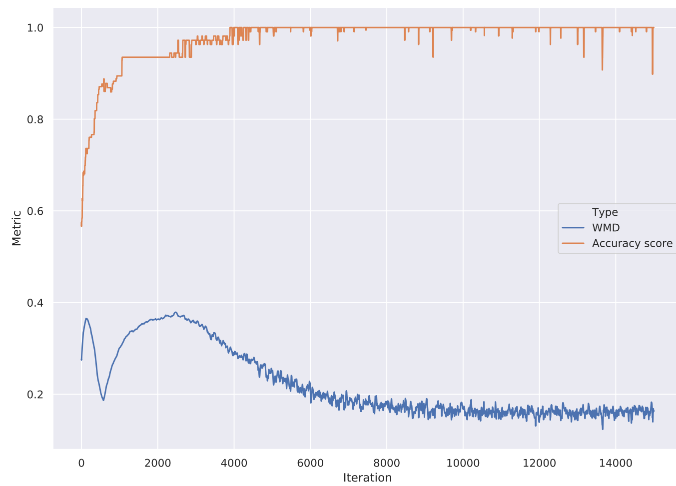
E. Evaluation

In order to evaluate the performance of the system, we have considered the classification accuracy as the measure of the performance index. The accuracy is given in percentage (%). We did not perform any hyper-parameter tuning on the data. All hyperparameters were tested on a dummy dataset of trying to separate two Gaussian under covariate shift (i.e. the centre's change). We train the network using Adam with a learning rate of 0.0003 for 2000 iterations for "DANN = False" and 1000 iterations when "DANN = True", where DANN stand from Domain Adversarial Neural Network and signifies whether or not the adversarial loss was used. Due to the inherent instability of smaller sample sizes, n number of neural networks are trained (more on this below).

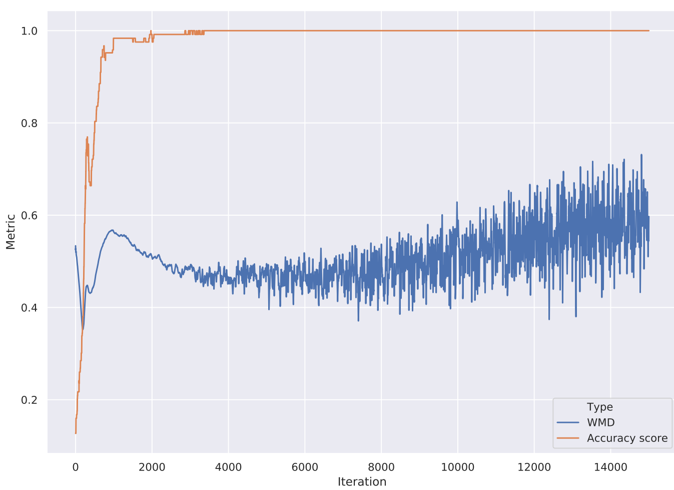
IV. Results

The features obtained with CSP are depicted for subject A03 in Fig. 2. Each of sub-figures (a)-(j), represents a CSP feature corresponding to a frequency band. The blue crosses and red circles denote the features of the left hand and right hand MI, respectively. The black line represents a possible linear separation between the features of the two classes obtained from each frequency band. The hyperplane is plotted for illustration purposes only. We performed two

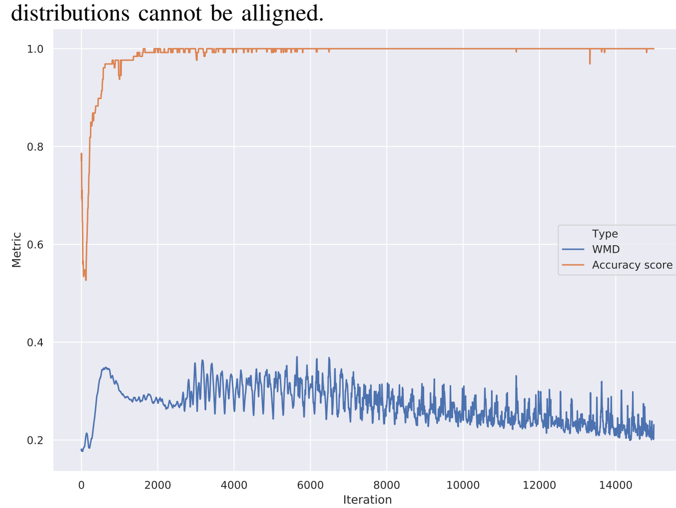
(a) Dataset 2A [19], Subject 1 measurements with DANN = False. The distance betwen the features learned oscillates



(b) Dataset 2A [19], Subject 1 training example with DANN = True - observe the drop in WMD after accuracy reaches one in the training set; there is very little improvement compared to DANN=False - the distributions cannot be alligned.



(c) Dataset 2A [19], Subject 3 measurements with DANN = False. WMD increases as the system keeps training.



(d) Dataset 2A [19], Subject 3 training example with DANN = True - observe the drop in WMD after accuracy reaches one. In this subject the drop is reflected in imporoved results.

Fig. 4: Comparisons between subjects of Dataset 2A [19] - Subject‘s 1 performance is not impacted by domain adaptation, while in subject 3 we can see an increase. Notice the drop in WSD score in the second case when DANN=True, indicating progress in aligning the distributions of the features.

sets of experiments, one with DANN enabled and another one with DANN disabled. The results can be seen in Table I for both the training and the test dataset. Note that due to the nature of neural networks and the size of the dataset, getting good performance on the training data is trivial. In all cases enabling the adversarial pathways produces better results. What is also worth noting is that in case of subject A03, DANNs achieve almost perfect results. Following this, we performed another experiment where we varied the number of bootstraps - the results are plotted in Fig. 5. It can be seen from the experiments that an increased number of bootstraps almost always increases performance, though there are cases where that is not true. Theoretically, the higher the number of bootstraps, the better results should be achieved, though this is not always the case as the resulting classifiers might be correlated. In order to better understand the behaviour of our algorithm, we plot the sum of Wasserstein distances (WMD) of each neuron in the last feature layer (a total of two tanh neurons) - the results for two sample subjects can be seen in Fig.4. The adversarial process (DANN=True) keeps does achieve its goal of minimizing the distance between different features in an unsupervised fashion.

V. Discussion

We have shown that the unsupervised domain adaptation, in combination with bagging, can be a very successful method in attacking problems of domain adaptation in EEG small datasets. An inherent problem of our results (probably owing to the very small size of the data presented) is the

TABLE I: Classification accuracy (in %) results from BCI competition IV Dataset 2A.

Subject	Training DANN=True	CTIL [17]	CSE-UAEL (Passive) [18]	CSE-UAEL (Active) [18]	DANN= False	DANN= True	DANN= False	DANN= True
					Run 1		Run 2	
		Test	Test	Test	Test	Test	Test	Test
A01	97.14	93.75	91.67	91.67	93.75	**93.75**	91.66	92.86
A02	95.71	61.11	63.89	63.89	62.50	**63.89**	62.87	**63.89**
A03	100	93.75	94.44	94.44	93.06	**99.31**	91.66	98.66
A04	97.85	69.44	70.80	72.22	66.67	**74.06**	67.36	73.53
A05	96.42	76.39	77.78	77.08	70.83	**77.61**	72.91	74.52
A06	94.28	66.67	73.61	75.69	63.89	**74.89**	62.50	73.68
A07	95.71	68.75	72.92	73.61	77.08	79.86	77.77	**81.25**
A08	99.28	92.36	93.75	94.44	93.06	**94.67**	93.05	91.66
A09	91.68	90.28	88.89	90.28	89.58	**92.89**	87.50	91.68
Mean	96.45	79.17	80.86	81.48	78.93	**83.43**	78.50	82.41
Std	2.52	13.30	11.44	11.33	13.45	12.06	12.72	11.76

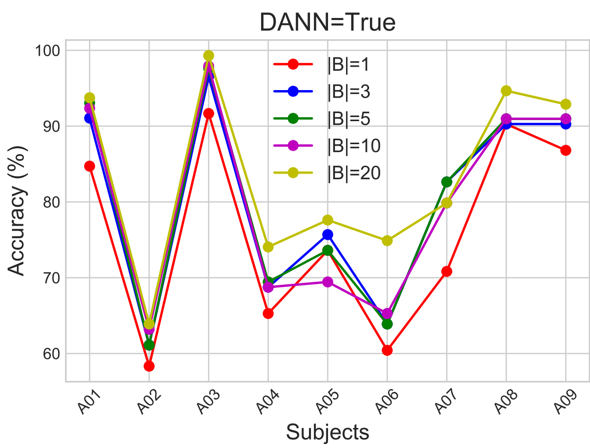


Fig. 5: Comparison of the classification accuracy with different bootstrap iterations with DANN = True, where $|B|$ is number of bootstrap iterations for Run 1.

instability of the classifier produced; increasing the number of bootstraps should stabilise this, but maybe with the added issues of increasing bias, due to classifier correlations. A possible way out is to use stacking [30] methods, instead of simple bagging algorithms; or alternatively, any modification of bagging that can help with stabilising classifiers, e.g. Bragging [31]. A further limitation of our study is that it was conducted in a single dataset - we will seek to combine multiple datasets in the future.

A possible area of exploration, this time however more from an EEG perspective, is why some subjects are far more amicable to method; our initial conclusions, which are in line with the data, point out to a situation where some subject are in a completely different part of the search space during testing, which makes most domain adaptation methods futile. Domain adaptation can help when one is transferring to distributions that share some common properties - it might be the case that there is a fundamental shift in the distributions for each subject. This also points towards the direction of capturing features that are amicable to domain adaptation methods; though no covariate shift should be present (we are in theory measuring the same responses), it might be there in practice.

The third avenue of future research is improving hyper-parameters; this has proved difficult in this study, as doing cross-validation with so small datasets which include covariate shift is problematic. In the future, we aim at developing methods that can help towards this direction.

Finally, another avenue worth taking is detecting if the domain adaptation process was successful. We have measured the sum of Wasserstein distance for each individual neuron in the final feature extraction layer, but this does not accurately reflect if the distributions are the same - we will need to use the multivariate version of the distance to do this. Given that batch normalisation turn the distribution of the activation into a normal distribution, we could potentially create methods hold this assumption and make the distance measurements more precise.

Finally, we aim at improving the training time of the network. The implementation is currently very inefficient and can be trivially increased in speed. This will allow easier stability testing.

VI. Conclusion

The train and the test distribution are rarely the same in real-world problems, while it is also extremely hard to label data effectively. Algorithms that can adapt online are crucial to the success of machine learning. We have shown that a combination of known methods can vastly improve performance in the challenging setting of EEG. Our future work will focus on stabilising the classifiers while creating measures that can help to detect whether or not the adaptation process was successful.

References

[1] J. G. Moreno-Torres, T. Raeder, R. Alaiz-RodríGuez, N. V. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," *Pattern Recognition*, vol. 45, no. 1, pp. 521–530, 2012.

[2] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain-computer interfaces for communication and control," *Clin. Neurophysiol.*, vol. 113, no. 6, pp. 767–91, Jun. 2002.

[3] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K. R. Müller, "Optimizing spatial filters for robust eeg single-trial analysis," *IEEE Signal Proc. Mag.*, vol. 25, no. 1, pp. 41–56, 2008.

[4] H. Raza, H. Cecotti, Y. Li, and G. Prasad, "Adaptive learning with covariate shift-detection for motor imagery-based brain–computer interface," *Soft Computing*, vol. 20, no. 8, pp. 3085–3096, 2016.

[5] Y. Li, H. Kambara, Y. Koike, and M. Sugiyama, "Application of covariate shift adaptation techniques in brain-computer interfaces," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 6, pp. 1318–1324, Jun. 2010.

[6] M. Arvaneh, C. Guan, K. K. Ang, and C. Quek, "Optimizing spatial filters by minimizing within-class dissimilarities in electroencephalogram-based brain-computer interface," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 24, no. 4, pp. 610–619, 2013.

[7] H. Raza, G. Prasad, and Y. Li, "Ewma model based shift-detection methods for detecting covariate shifts in non-stationary environments," *Pattern Recognition*, vol. 48, no. 3, pp. 659–669, Aug. 2015.

[8] B. Blankertz, G. Curio, and K. R. Müller, "Classifying single trial EEG: Towards brain computer interfacing," in *Advances in Neural Information Processing Systems*, vol. 1, 2002, pp. 157–164.

[9] R. Leeb, F. Lee, C. Keinrath, R. Scherer, H. Bischof, and G. Pfurtscheller, "Brain-computer communication: motivation, aim, and impact of exploring a virtual apartment," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 15, no. 4, pp. 473–82, Dec. 2007.

[10] M. Sugiyama, M. Krauledat, and M. K. R., "Covariate shift adaptation by importance weighted cross validation," *J. Mach. Learn. Res.*, vol. 8, pp. 985–1005, 2007.

[11] A. Chowdhury, H. Raza, Y. K. Meena, A. Dutta, and G. Prasad, "An eeg-emg correlation-based brain-computer interface for hand orthosis supported neuro-rehabilitation," *Journal of neuroscience methods*, vol. 312, pp. 1–11, 2019.

[12] P. Shenoy, M. Krauledat, B. Blankertz, R. P. N. Rao, and K.-R. Müller, "Towards adaptive classification for bci," *J. Neural Eng.*, vol. 3, no. 1, pp. R13–R23, Mar. 2006.

[13] C. Vidaurre, A. Schlögl, R. Cabeza, R. Scherer, and G. Pfurtscheller, "A fully on-line adaptive bci," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 6, pp. 1214–1219, 2006.

[14] H. Raza, H. Cecotti, Y. Li, and G. Prasad, "Learning with covariate shift-detection and adaptation in non-stationary environments: Application to brain-computer interface," in *2015 International Joint Conference on Neural Networks (IJCNN)*, July 2015, pp. 1–8.

[15] H. Raza, H. Cecotti, and G. Prasad, "Optimising frequency band selection with forward-addition and backward-elimination algorithms in eeg-based brain-computer interfaces," in *2015 International Joint Conference on Neural Networks (IJCNN)*, July 2015, pp. 1–7.

[16] A. Chowdhury, H. Raza, Y. K. Meena, A. Dutta, and G. Prasad, "Online covariate shift detection based adaptive brain-computer interface to trigger hand exoskeleton feedback for neuro-rehabilitation," *IEEE Transactions on Cognitive and Developmental Systems*, pp. 1–1, 2018.

[17] H. Raza, H. Cecotti, and G. Prasad, "A combination of transductive and inductive learning for handling non-stationarities in motor imagery classification," in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 763–770.

[18] H. Raza, D. Rathee, S. Zhou, H. Cecotti, and G. Prasad, "Covariate shift estimation based adaptive ensemble learning for handling non-stationarity in motor imagery related eeg-based brain-computer interface," *arXiv preprint arXiv:1805.01044*, 2018.

[19] M. Tangermann, K. R. Müller, A. Aertsen, N. Birbaumer, C. Braun, C. Brunner, R. Leeb, C. Mehring, K. J. Miller, G. R. Müller-Putz, G. Nolte, G. Pfurtscheller, H. Preissl, G. Schalk, A. Schlögl, C. Vidaurre, S. Waldert, and B. Blankertz, "Review of the bci competition iv," *Front. Neurosci.*, vol. 6, p. 55, 2012.

[20] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[21] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[24] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," *arXiv preprint arXiv:1409.7495*, 2014.

[25] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Computer Vision and Pattern Recognition (CVPR)*, vol. 1, no. 2, 2017, p. 4.

[26] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, "Revisiting batch normalization for practical domain adaptation," *arXiv preprint arXiv:1603.04779*, 2016.

[27] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[28] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: many could be better than all," *Artificial intelligence*, vol. 137, no. 1-2, pp. 239–263, 2002.

[29] H. Raza, H. Cecotti, and G. Prasad, "Optimising frequency band selection with forward-addition and backward-elimination algorithms in EEG-based brain-computer interfaces," in *Proc. of the Int. Joint Conf. on Neural Net.*, 2015, pp. 1–7.

[30] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.

[31] P. L. Bühlmann, "Bagging, subagging and bragging for improving some prediction algorithms," in *Research report/Seminar für Statistik, Eidgenössische Technische Hochschule (ETH)*, vol. 113. Seminar für Statistik, Eidgenössische Technische Hochschule (ETH), Zürich, 2003.