Enhanced Onset Detection of EEG for Self-paced Brain-Computer Interface using Deep Oversampling

Noura Al Moubayed
School of Engineering and Computer Sciences
University of Durham
Durham, UK
noura.al-moubayed@durham.ac.uk

Bashar Awwad Shiekh Hasan
Institute of Neuroscience
School of Medical Sciences
Newcastle University
Newcastle Upon Tyne, UK
bashar.awwad-sheikh-hasan@newcastle.ac.uk

Andrew Stephen McGough
School of Engineering and Computer Sciences
University of Durham
Durham, UK
stephen.mcgough@durham.ac.uk

***Abstract*—A deep learning approach for oversampling of electroencephalography (EEG) recorded during self-paced hand movement is investigated for the purpose of improving EEG classification in general and onset detection in particular. Oversampling of the movement class significantly enhances the overall accuracy of an onset detection system tested on 12 participants. Modelling the data using a deep neural network not only helps oversampling the movement class but also can help build a subject independent model of movement independent of the subject. In this work we present initial results on the applicability of this model.**

I. Introduction

Brain-Computer Interface (BCI) is an alternative communication medium between human and the machine where direct brain signals are used to control devices in the surrounding environment [1], [2]. This technology has a wide range of applications from assistive living [3], [4], communicating with locked-in patients [5], to car control [6], and gaming [7].

A BCI user can perform several well-studied mental tasks (e.g. imagining a limb movement) to induce changes in brain activity detectable via non-invasive imaging technique such as electroencephalography (EEG). Such a system must be able to distinguish EEG patterns produced by these tasks within a time frame suitable for control. One approach is based on motor-imagery, where the user imagines moving their limbs [8], [9], [10] . Motor imagery tasks are commonly applied in BCI due to their spatial separability and widespread understanding of their underlying physiological properties. Event-related desynchronization/synchronization (ERD/ERS) studies [9], [10] demonstrated that motor imagery tasks within a synchronous paradigm (i.e. the timing is controlled by the system) go through three consecutive phases: preparation, execution and after execution [11].

Previous research on event related ERD/ERS showed that during real movements relevant EEG activity can be found in both contralateral and ipsilateral hemispheres, but in the case of imagined movements only contralateral hemisphere gets activated [10]. This justifies the use of real movements to test new methods, because the experiments are easier to conduct and the labelling is much more reliable in the self-paced configuration (i.e. when the timing of the system is controlled by the user).

Early in BCI research, it faced a challenging problem of knowing when to switch on/off the system and how to detect the idle from active states. In [12] a brain-actuated switch was presented for self-paced BCIs. An unsupervised approach to onset detection was presented in [13] using Gaussian mixture models. An onset detection system was used in [14] to predict intention of reaching movement for use with a prosthetic arm. For onset detection to be practical, the false positive rate must be as low as possible, to increase the reliability of the system especially when safety is an issue. This is particularly difficult due to the highly imbalance nature of self-paced data. To overcome this issue, researchers either use a synchronous, cued, protocol to record training data where equal time windows are given for both baseline and motor activity [14], [15]. The downside of this approach is that downsampling will inherently reduce the information available for learning the baseline. Alternatively, in [16] we tried to model the temporal information as a means to better represent the dynamics of the self-paced data. However, even with the enhanced accuracy with temporal modelling the problem persists.

To keep our terms consistent. We refer to the recorded EEG data as "samples", while "events" are the time windows when movement happens. In self-paced onset detection we are interested in the accuracy of detecting these events with minimum false positive, i.e. instances when an onset event is wrongly detected. Hence the assessment is based on the performance of the system in detecting events, rather than the accuracy of classifying samples. In Section II-E we discuss how the predicted samples are processed to detect events.

A. Learning from imbalanced data

Imbalanced datasets are those where one class is over-represented in relation to the other class(es). This is usually due to intrinsic factors of the dataset [17](e.g. rare medical conditions, difficult and expensive acquisition of data from one class, etc...). In [18], the authors argued that the dataset complexity is the major factor behind the deterioration of classification accuracy, but it is exasperated by the inter-class imbalance. Data complexity is a loosely defined term that comprises: inter-class overlapping, lack of representative

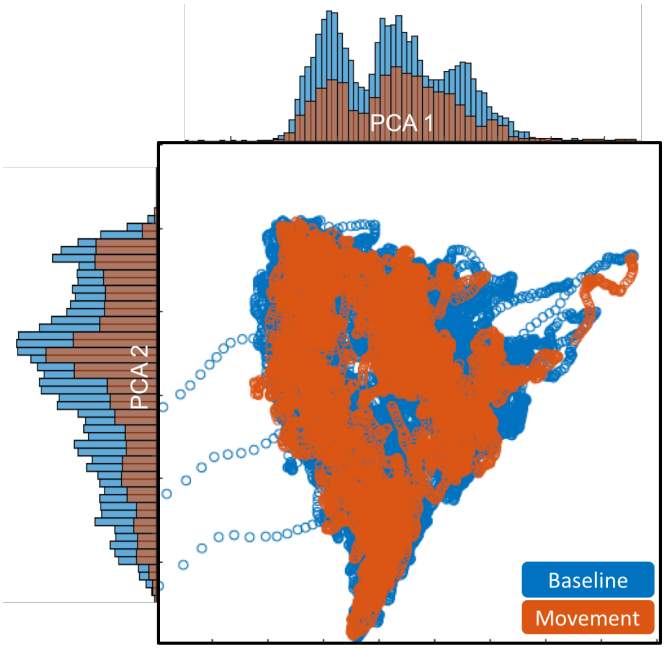

Fig. 1. Sample of self-paced EEG data projected using PCA on a two dimensional space. The figure demonstrates two challenges of classifying self-paced BCIs: I) the overlap between the two classes II) the imbalance in number of samples between the classes apparent in the histograms.

data, non-linear boundaries, time-variant data, and others. EEG driven BCI data are notorious for having the said characteristics of complexity [19], [20]. The problem is especially challenging when operating in a self-paced paradigm where controlling equal number of action (e.g. imagery movements) and baseline windows is almost impossible to achieve. Figure 1 demonstrates the challenge of classifying self-paced BCI data with overlapping imbalanced classes.

An added challenge to the BCI data classification is the high dimensionality of the extracted features (up to hundreds of features) in comparison to the available samples especially from the minority class (usually tens of events). This leads to poor generalisation of the learning algorithm especially when it is presented with imbalanced data sets leading to over-fitting. Feature selection and dimensionality reduction can be used to mitigate the effect of high dimensionality [21].

Tackling the problem of imbalanced data is a growing research field within machine learning [22], [18]. Intuitively speaking the problem can be solved either by finding a way to equalise the number of samples of all the classes or by introducing a new cost function of the learning algorithm that takes the imbalance of the data into consideration. Cost sensitive methods include AdaBoost motivated methods [23], Decision Trees [24], and neural networks [25], or using feature selection with an imbalance sensitive cost measure [26], [27]. Sampling, however, is the arguably the most commonly used method to enhance accuracy with imbalanced data [28], [29]. Sampling can be by either randomly over-sample /under-sample the minority or the majority classes accordingly. In [15] the baseline was under-sampled by taking a window of data of equal size to proceeding the movement window. Synthetic Minority Over-sampling Technique (SMOTE) [30], [31] and its variants are one of the commonly used methods in the literature and is briefly described in the next section.

In this work we address the imbalance of self-paced data using oversampling of the active class. To achieve this goal a Generative moment matching networks (GMMN) [32] is used. A deep generative model of the data is built and utilised to generate independent samples via a single feedforward pass through the layers of the neural network. The use of GMMN is advantageous not only for oversampling but also as a tool to build subject independent model of the data. In this work we present tentative results of this approach and we discuss its future use. The methods are tested on self-paced real finger EEG data collected from 12 participants. Electromyography (EMG) data which records the muscle activity are used as accurate labels to better quantify the performance of the different methods.

Next section briefly describes of the two oversampling and classification methods used here. The experimental design and data pre-processing are described in Section III. The results are presented in SectionIV, while Section V concludes the paper.

II. Methods

To circumvent the problem of imbalanced data and before classifying the data into baseline and movement, the movement data is oversampled using an unsupervised deep generative neural network. To compare with a non-generative oversampling model, we use SMOTE . To compare with a cost sensitive method, we use a feature selection based approach. All the methods use the same linear discriminant analysis (LDA) based classifier, and are described in the following.

A. GMMN

The motivation behind using deep learning is to be able to build a model of the minority class that could be used to synthesis minority data. To be able to build such a model, unsupervised deep learning would be used as it is capable of learning manifolds where there is high density of the data rather than maximising the margin among classes [33]. Generative models has the ability to evaluate the generalisation in the feature space. In [32] a generative network for unsupervised deep learning, generative moment matching network (GMMN) was proposed. GMMN uses a feedforward neural network to create a mapping from an easy to sample distribution space to the data space. GMMN starts by a simple prior of the parameters of the neural network making it easy to draw samples. The priors are propagated through the network in a deterministic manner to produce a sample of the data as the output of the network. In contrast to the complicated Markov Chain Monte Carlo (MCMC) methods required by Restricted Boltzmann Machines (RBM) [34], [35], samples can easily be drawn from a GMMN network. Also unlike the recently developed generative adversarial networks (GAN) [36], GMMNs are trained on a straightforward loss function using backpropagation.

For GMMN to work, it depends on a statistical hypothesis testing framework, maximum mean discrepancy (MMD) [37]. By training the model, and minimising the discrepancy we would be matching all moments of the model distribution to the distribution of the modelled data. A kernel is used to simplify the loss function keeping the training efficient.

The top hidden layer $h \in R^H$ contains H hidden units with a simple prior, e.g. uniform, on each unit independently,

$$p(h) = \prod_{j=1}^{H} U(h_j) \tag{1}$$

, where $U(h_j)$ is a uniform distribution. h is then passed through the neural network and then deterministically mapped to a vector $d \in R^D$ in the data space.

$$d = f(h, w) \tag{2}$$

, where f is the mapping function representing the neural network and w is the network parameters. The network can contain a number of nonlinear layers (e.g. ReLu, sigmoid, ...). Given the prior $p(h)$ and the mapping $f(h, w)$ a new sampled set in the data space can be generated.

The advantage of GMMN is that training the parameters of the network can be done using a standard backpropagation to minimise MMD as an objective. Using a Gaussian kernel the objective function can be written as:

$$\begin{aligned}\mathcal{L}_{MMD} = &\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} k(x_i, x_j) - \frac{2}{NM}\sum_{i=1}^{N}\sum_{l=1}^{M} k(x_i, y_l) \\ &+ \frac{1}{M^2}\sum_{l=1}^{M}\sum_{v=1}^{M} k(y_l, y_v)\end{aligned} \tag{3}$$

, where x_i are the generated sampled data, y_l are the original training data. N is the number of generated samples and M is the number of original data. k is the Gaussian kernel:

$$k(x, y) = exp(-\frac{1}{2\sigma}|x - y|^2) \tag{4}$$

, and σ is the bandwidth parameter. The gradient of the objective function can easily be calculated analytically and hence can easily be back propagated to update the weights of the network.

In this study a two-layer ReLU network was built. First Layer contained 200 nodes, while the second layer had 150. σ was set to 3 and 5 for the first and second layers respectively. Maximum iterations was set to 10000 and 100 mini batch size was used.

B. SMOTE

Synthetic minority over sampling Technique (SMOTE) is a simply and very effective approach of over-sampling which has proven to be superior in many applications [18], [30], [31]. The minority class is over-sampled by creating samples in the feature space between each minority class sample and a k nearest neighbour samples of the same class along the line segments joining any/all of the neighbours. Depending on the desired amount of over-sampling a subset of the neighbours are randomly selected, e.g. to achieve 300% oversampling 3 nearest neighbours are randomly chosen. Synthetic samples are generated as following: Take the difference between the feature vector (sample) under consideration and its selected nearest neighbour. This difference is multiplied by a random number between 0 and 1, and then added to the feature vector, i.e. interpolate a sample point between the sample point and its neighbour. This causes the selection of a random point along the line segment between two related samples. The effectiveness of this approach is credited to the fact it forces the decision region of the minority class, within the decision trees framework, to become more general. The synthetic samples allows the classifier to create larger and less specific decision regions [30].

C. Feature Selection

Sequential Forward Floating Search (SFFS) was used to select up to 10 features [21]. The method starts by using only one feature and selecting the feature that results in the highest value of F1-measure (see II-F). Once this feature is selected the method is repeated to find the second one which in combination with the previously selected produces the best F1. Then a pruning step is performed where a feature is removed sequentially from the selected features to check if the evaluation measure is enhanced. Expansion and pruning goes into iterations until a maximum number of features is selected or a finite number of cycles are executed.

D. Classification

The data are assumed independent in time during the training of an LDA classifier, but the data sequence is maintained during testing. Over-sampling is only applied on the movement data during training. The generated samples are added to the original data and a 10-fold cross validation is performed with the condition of having samples of both classes in each fold. LDA is used as it is one of the most commonly used classifiers for BCI [7], [5].

E. Post Processing

Regardless of the sampling algorithm, or lack thereof, the output of the LDA classifier is smoothed using a 5-sample temporal window. The class of the window is selected using majority voting. To detect onset events, i.e. moving from baseline to movement, another larger overlapping decision window is selected. Due to the variability of movement continuous times among subjects, these decision windows were optimised per subject to increase the number of available events. An onset is detected if within one decision window at least 40% continuous of the window is classified as baseline followed by a 40% continuously classified as movement. If an onset is detected a 2 seconds debounce/refractory window is applied complying with the nature of EEG and our understanding of the neuro motor system, which therefore reduces the false positives.

F. Evaluation

The evaluation was conducted by 10-fold cross-validation. Number of training/testing events varied depending on the number of all events per participant, however the overall number of samples is the same.

To take the imbalance of the data into consideration on the level of events, we use the standard F1-measure and true-false difference (TF) [38].

Given (E) is the number of onsets, the number of true-positive (TP) detections, the number of false-positive (FP) detections, and the number of false-negative (FN) combined from all the folds. F1-measure is defined as:

$$F1 = 2.\frac{Precision * Recall}{Precision + Recall} \tag{5}$$

, where

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

TF is defined as:

$$TF = (\frac{TP}{E} - \frac{FP}{E + FP}) * 100 \tag{8}$$

III. Data Collection

A. Subjects and Motor Task

Data were recorded from fifteen right handed subjects, three subjects were female, ages ranged from 23 to 28. Subjects 3 and 8 were experienced users of a BCI system based on self-paced movement. Subjects 6, 9, and 11 had previous experience in online BCI experiments, the remaining subjects were naive to BCI systems. As the protocol used here was un-cued the number of trials performed within each run was variable. Each subject performed three runs in a single session. A run lasted 610 seconds. After a five second waiting period a fixation cross appeared on the screen. The fixation cross remained on the screen for 10 minutes during which EEG data were acquired. A five second post waiting period was used, to give the user some time to relax. Each subject performed 4 sessions (12 runs).

Within each run subjects were instructed to perform self-paced flexion /extension of the left index finger whilst the fixation cross was visible. Subjects were requested to perform the movement for between 5 and 10 seconds and to rest for at least 10 seconds between movements. Instructions were given to concentrate on the fixation cross as much as possible during each run. After each run EMG recordings were assessed to ensure subjects understood requirements and could moderate actions accordingly.

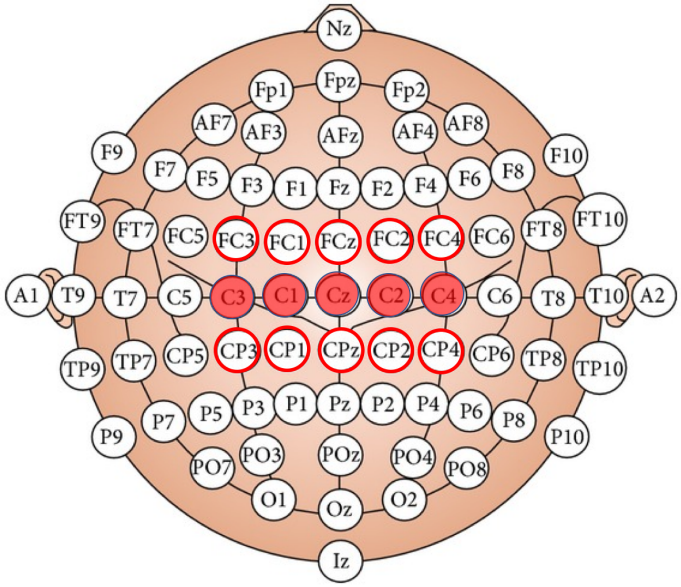

Fig. 2. Layout of the electrodes used to record the data using the standard 10-20 system. Image adapted from [40].

B. Data Acquisition

Five bipolar EEG channels were recorded over the motor cortex at locations C3, C1, Cz, C2 and C4. EMG was recorded from the flexors of the right forearm. A right mastoid reference channel was used. Signals were acquired using a Guger Technologies g.BSamp, Figure 2. EMG and EEG were acquired at 256 Hz and later down sampled to 25Hz. EMG was used to record muscle activity for establishing correct onset and offset time points of self-paced movements. This allows training data to be correctly labeled according to the real movement activities.

No artifact rejection or EOG correction was employed as visual inspection did not find significant artifacts in the recorded EEG signals. In addition, the filtering applied before feature extraction (common average reference and band-pass filtering) can play a role in removing some artifacts.

C. Feature Extraction

A common average reference is used to reduce the common noise. Similar to previous work [39] narrow power band features were extracted per channel. The mu, beta, and lower gamma bands are divided into even finer bands, so that feature selection method can be applied more efficiently. 90 features were used in total. For SMOTE and GMMN all the features are used, while feature selection is applied for comparison as described in Section II-C.

IV. Results

Figure 3 presents the precision vs recall results of the three compared methods applied on the 12 dataset as discussed above. If the method is performing well for both classes precision and recall should have comparable values,

TABLE I
DATA STATISTICS: NUMBER OF ONSETS FROM BASELINE TO MOVEMENT. PERCENT OF BASELINE DATA. PERCENT OF MOVEMENT DATA.

Subject	No. Onsets	Baseline	Movement
Subject 01	99	67.35	32.65
Subject 02	232	59.36	40.64
Subject 03	55	57.81	42.19
Subject 04	58	67.95	32.05
Subject 05	109	79.58	20.42
Subject 06	102	87.56	12.44
Subject 07	185	61.15	38.85
Subject 08	81	49.62	50.38
Subject 09	93	64.83	35.17
Subject 10	108	78.10	21.9
Subject 11	128	60.79	39.21
Subject 12	156	61.95	38.05

i.e. lying around the diagonal line. Each participant has a unique shape so the results of applying the method to their data can be compared. Results of oversampling using GMMN is represented in red, LDA with feature selection in blue, and Green for oversampling with Smote. The results show high correlation between precision and recall for GMMN (0.9798 with $p < 0.05$), and Smote (0.9448 with $p < 0.05$), and LDA (0.9343, $p < 0.05$). The F1 results in Figure 4 show a relative advantage of oversampling methods compared to LDA. Most importantly to onset detection, TF shows a significant improvement of oversampling over LDA with t-test resulting of a p-value ¡0.05 for both GMMN and Smote against LDA.

The results suggest that the advantage of oversampling is in its ability to help sustaining a continuous quality of the output, which results in higher onset detection accuracy after temporal smoothing as described above. This is further clarified in Figure 5. The figure shows the accuracy of movement and baseline classes. If the method is performing similarly to both classes the symbols would be expected on the diagonal line. Any deviation from it is interpreted as a bias to either class. It is clear that LDA has a strong bias to the majority baseline class compared to the over-sampling methods. The dotted lines represent the chance level.

V. DISCUSSION AND CONCLUSION

The work here presents a novel approach to solve the problem of imbalanced data in onset detection from real hand movement as a tool to enhance the onset detection in self-paced brain-computer interfaces. Unsupervised deep learning using a generative model is used to model the minority class, movement, and then synthetic samples were generated. The samples are then used to build an LDA classifier from now balanced data set allowing for higher classification accuracy.

The results are compared with those obtained using a non-generative over-sampling method, SMOTE, with comparable accuracies. Another alternative to over-sampling is to perform feature selection with an F1 measure as a cost function, termed LDA in the figures above. Feature selection performed worse than over-sampling especially when using TF, a custom

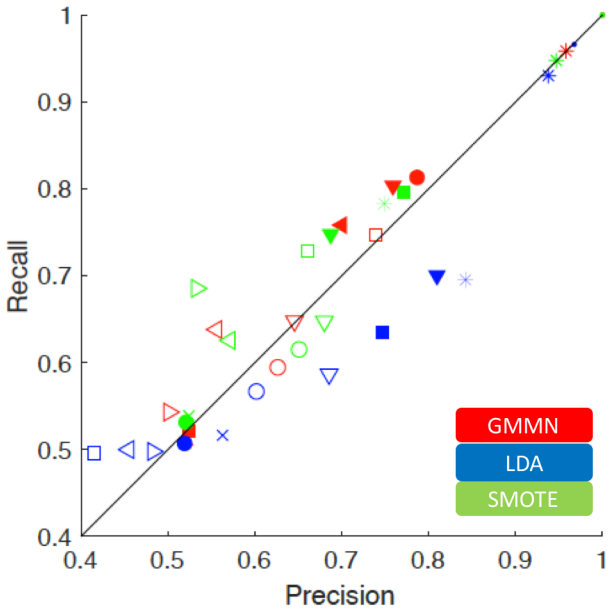

Fig. 3. Precision and Recall values for the three methods under comparison. Results from each subject are plotted with a unique shape. Colors represent the methods.

designed metric for onset detection which account for any bias to either classes. Statistical t-tests confirm these conclusions.

Although SMOTE and GMMN perform similarly on the data, however GMMN has the advantage of building a model of the data. SMOTE on the other hand is only interested in local topography. This gives GMMN the advantage of building subject-independent models, which is referred to as BCI illiteracy [41]. Enabling people to use BCI with minimum or no training is one of the biggest challenges of the wide adaptation of BCI. Only a few studies tried to tackle this issue though the build of ”feature” bank that are used to reduce the amount of training necessary for a new user [42]. As a proof of concept we present here some preliminary results of using GMMN to build subject independent model. The model is trained on data combined from 11 subjects and tested on the remaining subject in a cross-validation scheme. Over-sampling, classification, and post-processing is carried out similar to what has been described above. Figure 6 compares the subject independent GMMN results to those using a bank of band power features and an LDA classifier. The results clearly show that without the GMMN model the TF accuracy is well below 50% for most subjects, while GMMN consistently performs significantly above chance (t-test, p¡0.05). More work would be necessary to better explore the subject-independent model and test it on an online system, however the results provide a strong incentive for the use of deep generative models in BCI.

ACKNOWLEDGMENT

The authors are grateful to the EPSRC for funding this work.

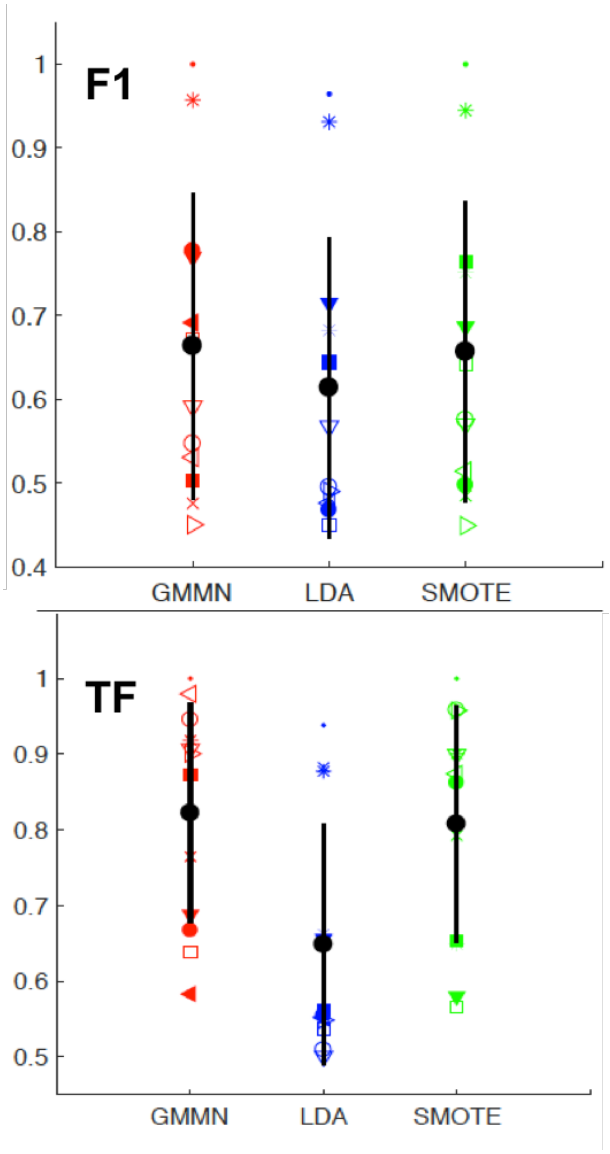

Fig. 4. F1 and TF values for the three methods under comparison. Results from each subject are plotted with a unique shape. Colors represent the methods.

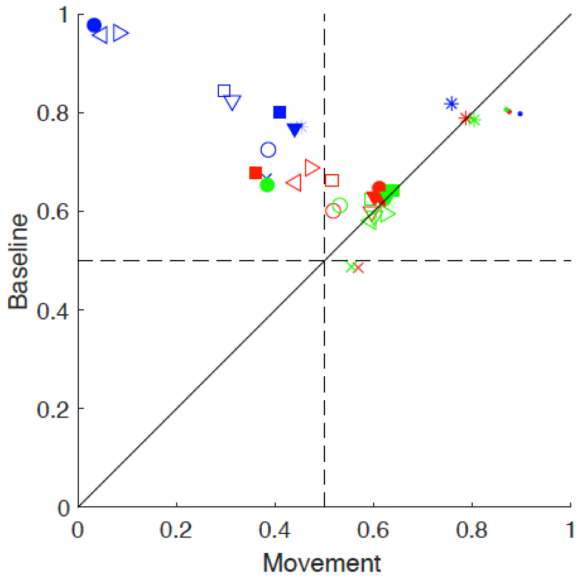

Fig. 5. Cross-validated classification results of both movement and baseline classes without the post-processing steps.

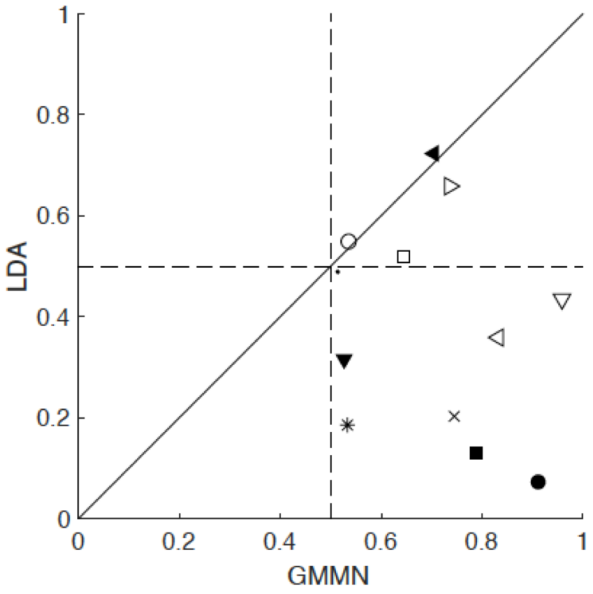

Fig. 6. TF results of using subject-independent model. The x-axis is the results obtained using GMMN and LDA results on the y-axis.

References

[1] J. Wolpaw and E. W. Wolpaw, *Brain-computer interfaces: principles and practice*. OUP USA, 2012.

[2] J. R. Wolpaw, N. Birbaumer, W. J. Heetderks, D. J. McFarland, P. H. Peckham, G. Schalk, E. Donchin, L. A. Quatrano, C. J. Robinson, T. M. Vaughan *et al.*, "Brain-computer interface technology: a review of the first international meeting," *IEEE transactions on rehabilitation engineering*, vol. 8, no. 2, pp. 164–173, 2000.

[3] N. Mora, I. De Munari, P. Ciampolini, and J. d. R. Millán, "Plug&play brain–computer interfaces for effective active and assisted living control," *Medical & Biological Engineering & Computing*, pp. 1–14, 2016.

[4] J. d. R. Millán, F. Galán, D. Vanhooydonck, E. Lew, J. Philips, and M. Nuttin, "Asynchronous non-invasive brain-actuated control of an intelligent wheelchair," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2009, pp. 3361–3364.

[5] U. Chaudhary, N. Birbaumer, and A. Ramos-Murguialday, "Brain–computer interfaces in the completely locked-in state and chronic stroke," *Progress in Brain Research*, vol. 228, pp. 131–161, 2016.

[6] Y. Yu, Z. Zhou, E. Yin, J. Jiang, J. Tang, Y. Liu, and D. Hu, "Toward brain-actuated car applications: Self-paced control with a motor imagery-based brain-computer interface," *Computers in biology and medicine*, vol. 77, pp. 148–155, 2016.

[7] B. A. S. Hasan and J. Q. Gan, "Hangman bci: An unsupervised adaptive self-paced brain–computer interface for playing games," *Computers in biology and medicine*, vol. 42, no. 5, pp. 598–606, 2012.

[8] A. Riehle and E. Vaadia, *Motor cortex in voluntary movements: a distributed system for distributed functions*. CRC Press, 2004.

[9] G. Pfurtscheller and F. L. Da Silva, "Event-related eeg/meg synchronization and desynchronization: basic principles," *Clinical neurophysiology*, vol. 110, no. 11, pp. 1842–1857, 1999.

[10] C. Neuper, M. Wörtz, and G. Pfurtscheller, "Erd/ers patterns reflecting sensorimotor activation and deactivation," *Progress in brain research*,

vol. 159, pp. 211–222, 2006.
[11] B. A. S. Hasan and J. Q. Gan, "Temporal modeling of eeg during self-paced hand movement and its application in onset detection," *Journal of neural engineering*, vol. 8, no. 5, p. 056015, 2011.
[12] S. G. Mason and G. E. Birch, "A brain-controlled switch for asynchronous control applications," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 10, pp. 1297–1307, 2000.
[13] B. A. S. Hasan and J. Q. Gan, "Unsupervised movement onset detection from eeg recorded during self-paced real hand movement," *Medical & biological engineering & computing*, vol. 48, no. 3, pp. 245–253, 2010.
[14] E. Lew, R. Chavarriaga, S. Silvoni, and J. d. R. Millán, "Detection of self-paced reaching movement intention from eeg signals," *Front. Neuroeng*, vol. 5, no. 13, 2012.
[15] C. Tsui, A. Vučković, R. Palaniappan, F. Sepulveda, and J. Gan, "Narrow band spectral analysis for movement onset detection in asynchronous bci," in *The 3rd international workshop on braincomputer interfaces*, 2006.
[16] B. A. S. Hasan, "On the temporal behavior of eeg recorded during real finger movement," in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer Berlin Heidelberg, 2011, pp. 335–347.
[17] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Information Sciences*, vol. 250, pp. 113–141, 2013.
[18] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.
[19] T. Burns and R. Rajan, "Combining complexity measures of eeg data: multiplying measures reveal previously hidden information," *F1000Research*, vol. 4, 2015.
[20] J. R. Millan, "On the need for on-line learning in brain-computer interfaces," in *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, vol. 4. IEEE, 2004, pp. 2877–2882.
[21] J. Q. Gan, B. A. S. Hasan, and C. S. L. Tsui, "A filter-dominating hybrid sequential forward floating search method for feature subset selection in high-dimensional space," *International Journal of Machine Learning and Cybernetics*, vol. 5, no. 3, pp. 413–423, 2014.
[22] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: special issue on learning from imbalanced data sets," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 1–6, 2004.
[23] Y. Sun, M. S. Kamel, A. K. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognition*, vol. 40, no. 12, pp. 3358–3378, 2007.
[24] M. A. Maloof, "Learning when data sets are imbalanced and when costs are unequal and unknown," in *ICML-2003 workshop on learning from imbalanced data sets II*, vol. 2, 2003, pp. 2–1.
[25] M. Kukar, I. Kononenko *et al.*, "Cost-sensitive learning with neural networks." in *ECAI*, 1998, pp. 445–449.
[26] Z. Zheng, X. Wu, and R. Srihari, "Feature selection for text categorization on imbalanced data," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 80–89, 2004.
[27] M. Wasikowski and X.-w. Chen, "Combating the small sample class imbalance problem using feature selection," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1388–1400, 2010.
[28] G. M. Weiss and F. Provost, "The effect of class distribution on classifier learning: an empirical study," *Rutgers Univ*, 2001.
[29] A. Estabrooks, T. Jo, and N. Japkowicz, "A multiple resampling method for learning from imbalanced data sets," *Computational intelligence*, vol. 20, no. 1, pp. 18–36, 2004.
[30] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.
[31] J. Luengo, A. Fernández, S. García, and F. Herrera, "Addressing data complexity for imbalanced data sets: analysis of smote-based oversampling and evolutionary undersampling," *Soft Computing*, vol. 15, no. 10, pp. 1909–1936, 2011.
[32] Y. Li, K. Swersky, and R. Zemel, "Generative moment matching networks," in *International Conference on Machine Learning*, 2015, pp. 1718–1727.
[33] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *Journal of Machine Learning Research*, vol. 11, no. Feb, pp. 625–660, 2010.
[34] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 791–798.
[35] R. Salakhutdinov and G. E. Hinton, "Deep boltzmann machines." in *AISTATS*, vol. 1, 2009, p. 3.
[36] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.
[37] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Advances in neural information processing systems*, 2006, pp. 513–520.
[38] G. Townsend, B. Graimann, and G. Pfurtscheller, "Continuous eeg classification during motor imagery-simulation of an asynchronous bci," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 12, no. 2, pp. 258–265, 2004.
[39] C. S. L. Tsui, "Adaptive self-paced brain-actuated control of mobility devices," Ph.D. dissertation, The University of Essex, 2009.
[40] S. Jirayucharoensak, S. Pan-Ngum, and P. Israsena, "Eeg-based emotion recognition using deep learning network with principal component based covariate shift adaptation," *The Scientific World Journal*, vol. 2014, 2014.
[41] C. Vidaurre and B. Blankertz, "Towards a cure for bci illiteracy," *Brain topography*, vol. 23, no. 2, pp. 194–198, 2010.
[42] S. Fazli, F. Popescu, M. Danóczy, B. Blankertz, K.-R. Müller, and C. Grozea, "Subject-independent mental state classification in single trials," *Neural networks*, vol. 22, no. 9, pp. 1305–1312, 2009.