Modeling electroencephalography waveforms with semi-supervised deep belief nets: fast

classification and anomaly measurement

Modeling electroencephalography waveforms with semi-supervised deep belief nets: fast classification and anomaly measurement

D F Wulsin[1]**, J R Gupta**[1]**, R Mani**[2]**, J A Blanco**[1] **and B Litt**[1,2]

[1] Department of Bioengineering, University of Pennsylvania, Philadelphia, PA, USA
[2] Department of Neurology, University of Pennsylvania, Philadelphia, PA, USA

E-mail: wulsin@seas.upenn.edu

Abstract

Clinical electroencephalography (EEG) records vast amounts of human complex data yet is still reviewed primarily by human readers. Deep belief nets (DBNs) are a relatively new type of multi-layer neural network commonly tested on two-dimensional image data but are rarely applied to times-series data such as EEG. We apply DBNs in a semi-supervised paradigm to model EEG waveforms for classification and anomaly detection. DBN performance was comparable to standard classifiers on our EEG dataset, and classification time was found to be 1.7–103.7 times faster than the other high-performing classifiers. We demonstrate how the unsupervised step of DBN learning produces an autoencoder that can naturally be used in anomaly measurement. We compare the use of raw, unprocessed data—a rarity in automated physiological waveform analysis—with hand-chosen features and find that raw data produce comparable classification and better anomaly measurement performance. These results indicate that DBNs and raw data inputs may be more effective for online automated EEG waveform recognition than other common techniques.

S Online supplementary data available from stacks.iop.org/JNE/8/036015/mmedia

1. Introduction

Clinical scalp EEG is used to diagnose and guide therapy for a variety of neurological conditions, including acute seizures and brain ischemia after stroke and cardiac arrest [1]. Clinical EEG monitoring often employees automated algorithms to detect epileptiform discharges and seizure-like activity, but most of these tools are plagued by poor performance and high false-positive rates [2], which limit their clinical usefulness. Most current automated algorithms detect a small group of isolated waveform patterns, such as spikes, seizures and eye blink artifacts.

Despite over two decades of research in automated classifiers, neurophysiologists still analyze EEG almost exclusively 'by hand'. Reliable and accurate detectors are very limited, due to their narrow focus, and waveforms of interest are similar enough to each other and to background activity to elude detection strategies that rely only on simple thresholding of the time or frequency domain EEG signal [3]. This is a particular problem of real-time EEG monitoring, especially in critically-ill patients in the intensive care unit (ICU) setting. Doctors are often put in the unenviable position of reading patient data retrospectively and finding disasters long after they happen. Developing robust detectors that have fast enough execution time to operate in real time is thus of great clinical importance, since they could be incorporated into an early-warning system that allows doctors to identify problems sooner.

In this study, we consider the problem of classifying individual second-long waveforms from individual channels into one of five clinically significant [1] EEG classes: (1) spike and sharp wave, (2) generalized periodic epileptiform
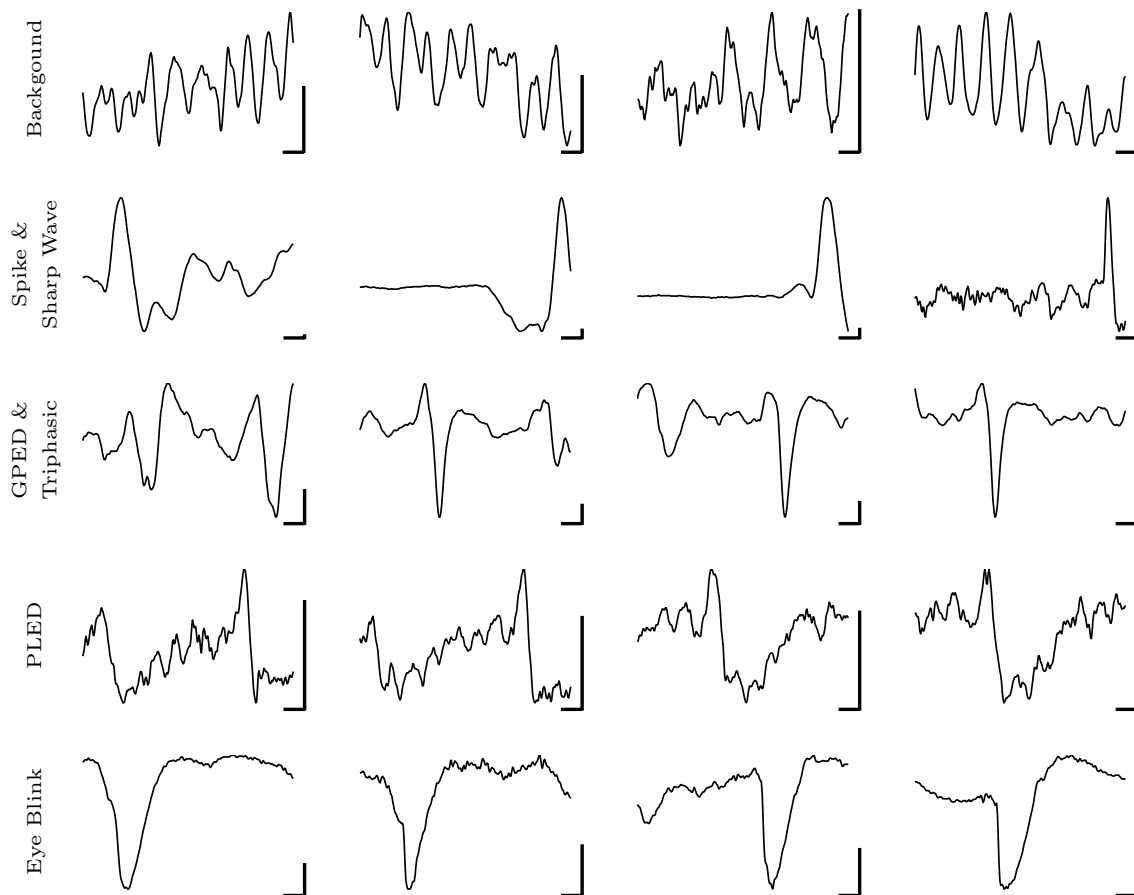
Figure 1. Four representative samples from each class of our EEG waveforms dataset. Abbreviations: GPED, generalized periodic epileptiform discharge; PLED, periodic lateralized epileptiform discharge. Horizontal scale bars show 100 ms, and vertical scale bars show 100 μV.

discharge (GPED) and triphasic waves, (3) periodic lateralized epileptiform discharge (PLEDs), (4) eye blink artifact and (5) background activity, defined as everything not belonging to one of the other classes. Figure 1 shows representative examples from each waveform class. Our aim is to develop a classifier that may be used to produce estimates of specific waveform rates and general waveform spatial and temporal frequency metrics that doctors may use in acute patient care, quantify in their clinical reports, and in reliably quantifying patient neurophysiological state—amount and type of epileptiform activity—for larger clinical studies.

Along with classifying EEG waveforms, we considered the more general task of determining how unusual, or anomalous, a specific signal is. From a clinical perspective, such a measure coupled to a visualization tool would be extremely useful for data visualization of 'abnormal' activity by technicians, nurses and neurophysiologists during continuous real-time monitoring of patient EEG signals. High levels of unusual signals may prompt readers to alert physicians caring for the patient. While detecting anomalies is a mature and active field of research [4], we explore an anomaly metric with potential clinical utility that naturally falls out of the classification deep belief net (DBN) training paradigm.

In the study below, we first compare the performance of four different classifiers using raw data and extracted features

as input to an EEG waveform pattern-recognition task. We then compare the execution times of each of these classifiers on the same volume of EEG data that would arrive in 1 s epochs of 'real-time' monitoring. Finally, we compare DBN anomaly detection performance using raw data versus that using extracted features as input.

2. Methods

We first briefly review aspects of the three standard classifiers and then discuss DBNs, feature selection, dataset collection and classifier training. We then describe the details of our three experiments.

2.1. Classifiers

We describe the implementations and parameters used for each classifier in appendix A.

Decision trees (DTs). Also known as classification trees, DTs have the advantages of intuitive decision rules and fast classification that involves only traversing the learned tree. They have been used in EEG [5] and other clinical classification tasks [6, 7]. We used a DT splitting on the Gini diversity index with pruning [8].
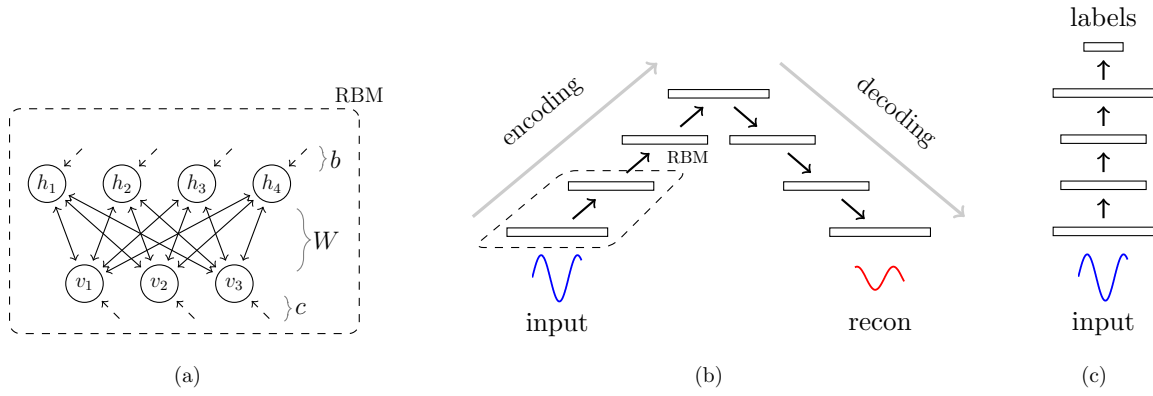
Figure 2. (a) A RBM contains hidden layer units h_j connected to the visible layer units v_i with symmetric weights W along with hidden layer biases b and visible layer biases c. (b) A DBN autoencoder can be initialized by stacking sequentially-trained RBMs on top of each other and then 'unrolling' the weights to form a feed-forward network. Here, the first three hidden layers encode successive representations of the data, and the last three decode previous representation to form a reconstruction of the input. (c) A DBN classifier is initialized from either stacked RBMs or the first half of a DBN autoencoder to form a feed-forward network. A label layer is stacked above the top hidden layer to produce the label output.

Support vector machines (SVMs).   SVMs are a very popular type of classifier and have also previously been used in EEG classification tasks [9]. SVMs maximize the margin of the separating hyperplane by solving a quadratic optimization problem [10]. We used a SVM with a radial basis function kernel.

k-nearest neighbors (KNNs).   KNNs are a nonparametric classifier that determines a testing sample's class by the majority class of the k closest training samples. Many different distance measures have been used with KNNs as well as different weights on the classes of the neighbors, but we use one of the most common versions that employs Euclidean distance and equal neighbor weighting.

Deep belief nets (DBNs).   DBNs are a relatively new type of multi-layer neural network that are capable of learning high-dimensional manifolds of the data. A thorough description of DBN varieties and their training is available elsewhere [11, 12]. We consider a DBN composed of logistic restricted Boltzmann machines (RBMs)—a generative model—with symmetric weights $\mathbf{W}$ between binary visible units $\mathbf{v}$ and binary hidden units $\mathbf{h}$ as well as biases $\mathbf{b}$ and $\mathbf{c}$ to the hidden layer and visible layer (figure 2(a)). In both the RBM and DBN, the training algorithm learns the weights and biases between adjacent layers of the network.

An RBM has a joint distribution

$$P(\mathbf{v}, \mathbf{h}) = \frac{1}{z} e^{-\mathbf{h}^T \mathbf{W} \mathbf{v} - \mathbf{b}^T \mathbf{h} - \mathbf{c}^T \mathbf{v}} \tag{1}$$

with the normalization constant z and thus has an energy function

$$E(\mathbf{v}, \mathbf{h}) = \mathbf{h}^T \mathbf{W} \mathbf{v} + \mathbf{b}^T \mathbf{h} + \mathbf{c}^T \mathbf{v}. \tag{2}$$

The binary units of the hidden layer are Bernoulli random variables, where each hidden unit h_j is activated, here with the logistic sigmoid function, based on each visible unit v_i with probability

$$P(h_j = 1) = \text{Sigm}\left(b_j + \sum_i v_i W_{ij} \right). \tag{3}$$

Calculating the gradient of the log likelihood of $\mathbf{v}$ is intractable and so contrastive divergence after k iterations of Gibbs sampling (often $k = 1$) [13] is usually used to approximate it:

$$\frac{\partial \log P(\mathbf{v})}{\partial W_{ij}} \approx \langle v_i h_j \rangle^0 - \langle v_i h_j \rangle^k \tag{4}$$

where $\langle \cdot \rangle^m$ represents the average value at a contrastive divergence iteration m. In practice, we also use momentum in the weight update to prevent getting stuck in local minima and standard ℓ_2 regularization to prevent the weights from getting too large. This regularization also prevents hidden layers with more units than their input layer from learning trivial (one-to-one mapping) features of their inputs.

To form a DBN, RBMs are individually trained one after another and then stacked on top of each other, where the visible layers of higher RBMs are the hidden states of the previous RBM. In unsupervised DBNs, the n RBMs are 'unrolled' to form a $2n - 1$ directed encoder–decoder network that can be fine-tuned with backpropagation [14]. Figure 2(b) shows a diagram of an 'unrolled' autoencoding DBN. Each unit (node) in the network can be viewed as describing a learned feature of the input it receives. In the first hidden layer, these are features of the data, but in higher layers, they are features of features. Training this deep autoencoder is an attempt to learn the weights and biases between each of the layers such that the reconstruction and the input sample are as close to each other as possible.

For supervised DBNs, we add a label layer to the highest encoding layer and ignore the decoding layers (figure 2(c)). Hinton *et al* use the contrastive wake-sleep algorithm for supervised learning [11], which has the added benefit of also making the DBN generative. We found that in tasks where the generative properties of the DBN are unnecessary, standard backpropagation has a lower classification error and is faster since we eliminate both fine-tuning the generative weights and the longer Gibbs sampling between the top layers.

Table 1. Hand-chosen features considered.

EEG feature
Area
Normalized decay
Frequency band power
Line length
Mean energy
Average peak/valley amplitude
Normalized peak number
Peak variation
Root mean square
Wavelet energy
Zero crossings

Table 2. Prevalence of labeled EEG waveform samples of each class in the dataset.

Class	Prevalence	Number of samples
Spike and sharp wave	2.32%	2357
GPED and triphasic	4.41%	4486
PLED	1.40%	1422
Eye blink	0.33%	332
Background	91.6%	93 203
Total	100%	101 800

2.2. Anomaly measurement using DBNs

The autoencoding DBN previously described (and shown in figure 2(b)) produces a reconstructed signal as close as possible to the input signal given a fixed number of layers. We hypothesize that the DBN learns types of signals that are more prevalent in the training data, producing better reconstructions of them. Similarly, 'unusual' (or anomalous) signals will occur rarely in the training data, preventing the DBN from learning (and reconstructing) those as well. While some aspects of even common signals seem harder for the DBN to learn (e.g. higher frequency, lower amplitude components), we have found that the DBN generally learns most aspects of the common signals better than those of the uncommon signals.

We quantify how well the DBN transforms an input sample **x** to a reconstruction **z** by the root mean-squared-error over the dimensions of the samples.

2.3. Features

It is common in EEG classification tasks to preprocess the raw signal by computing features selected either directly [9] or algorithmically from a larger pool of candidates that are predetermined by humans [15]. These 'hand-chosen' features are then used as inputs to the classifier. Not only does this process allow designers to incorporate domain knowledge, often seen as useful for improving the physiological interpretability of results, but it can also greatly reduce the computational burden and alleviate the curse of dimensionality. Classifiers working with hand-chosen features extracted from raw data often use inputs that the designer believes should separate the data well or are in some way related to the target outcome of the classifier [16]. Ideally they are also independent of one another. The fundamental question when using such hand-chosen features is which ones to use.

We compared using raw, unprocessed data with user-defined features as inputs to our classifier. We considered the features listed in table 1, all of which have some hypothesized relevance to the task, and have been previously used in EEG signal processing. A more thorough description of each feature is given in appendix B.

We evaluated the classification performance of each potential feature individually as a first step in deciding which to include in our final feature set. We tested each of the 11

candidate features[3] as input to a KNN classifier ($k = 3$) and ranked the features on how well they predicted the class of samples in any of the four non-background classes. Thus, a feature that had high performance for only a single non-background class was still ranked well among the features.

Using this ranking of the 11 features, we formed 11 groups of features, where the first group contained only the top ranked feature, the second group the top two features, the third group the top three features, and so on[4]. We then looked at the classification performance of each of these groups, again using the KNN ($k = 3$). The best group was that which included every feature shown in table 1 except zero crossings.

2.4. EEG dataset collection

Referential scalp EEG sampled at 256 Hz was recorded from 11 patients undergoing continuous EEG monitoring while receiving therapeutic hypothermia treatment comatose after cardiac arrest. Since the waveforms of interest (i.e. classes 1–4) occurred very sparsely in time, we selected thirteen 2 h blocks of all channels based on the clinical reports of where these events were prevalent, generated after viewing the data in bipolar montage [19]. We then randomly subsampled one thousand 2 min segments across all channels from these 2 h blocks. A clinical epileptologist (RM) labeled 1 s long samples for each channel (which we refer to as channel-seconds) in 50 random 2 min segments containing all channels, often labeling individual channels in the same second differently since the EEG waveforms of interest may have only been present in one or a few channels at a given time. The reviewer had access to his clinical notes for the patient being marked. Table 2 shows the prevalence of classes in this dataset.

Although our classifiers accept only data from each individual channel-second, the human marker was given as much context data as possible surrounding each 2 min segment during marking in an effort to maximize marking accuracy.

We divided the labeled data (50 random 2 min segments) and unlabeled data (950 random 2 min segments) into individual channel-seconds. Since these individual channel-seconds were the only data used by the classifiers, they have

---

[3] Three of the features, average peak/valley amplitude, frequency band power and wavelet energy, contained more than one value but were treated as one feature.

[4] A more global method for selecting which features to include as classifier inputs would be necessary as the number of features examined increases. Genetic algorithms have been used for this type of feature selection in the past [17], although we could also explore applying principal components analysis to the feature themselves [18].

Table 3. Number of samples in each part of the EEG waveform dataset.

	Number of samples	
	Unlabeled	Labeled
Training	500 000	72 800
Validation	100 000	14 500
Testing	100 000	14 500
Total	700 000	101 800

no information from other channels or prior data from the same channel in their learning and classification. Labeled and unlabeled data were randomly subsampled to form ten training, validation and testing sets with the number of samples in each shown in table 3. We created three separate datasets from these samples to be used by the classifiers tested. In the first dataset, which we call the *raw256* dataset, each sample was individually scaled so that its 256 data points had values between zero and 1. Scaling parameters used for each sample were also encoded as [0, 1] values[5] and prepended to each sample so that the original signal voltage information would not be lost. In the second dataset, which we call the *feat16* dataset, we extracted 16 hand-chosen features (with selection criteria described in section 2.2) from each sample. To limit the influence of potential outliers in each feature, the bottom and top 5% values of each feature were truncated to zero and 1, respectively, while the rest of the feature space was scaled between zero and 1. In the third dataset, which we call the *pca20* dataset, we performed principal component analysis (PCA) on the raw (260-dimensional) samples and used the coefficients corresponding to the first 20 eigenvectors, accounting for an average of 92.75% of the variance across the ten partitions. As in the *feat16* samples, the *pca20* samples were normalized to have a minimum of zero and a maximum of 1.

2.5. Classifier training

We selected common parameters for each of the four classifiers and used cross-validation over the ten partitions to pick the parameters that led to the highest average F_1 in the validation sets. Appendix A describes in detail the parameter space searched for each classifier and the optimal values found. Below we describe some of the training details for the DBN. For training details of the decision tree, support vector machines and KNN classifiers, we refer readers to the implementations given in appendix A.

Since the architecture of a DBN can greatly influence its performance, the number of units in each layer[6] of the DBN was our main search parameter. Hinton *et al* used a three-layer DBN (500-500-2000) for MNIST digit classification [11]. Since adding layers can only improve a DBN's modeling power [20], we used four-layer DBNs in this study, as the added computational cost was reasonable given our available

resources. Training was done on a Mac OS X 10.5 array of 36 dual-core Intel Xeon CPUs (2.26–2.8 GHz).

An advantage of DBNs is that they can learn most of their representation on unlabeled data, which is abundant in our case. We first performed unsupervised layer-wise RBM training and then fine-tuned the DBN with backpropagation on the unlabeled data. The resulting DBN was then used to initialize the weights for supervised fine-tuning with backpropagation. We found that this three-step semi-supervised process offered higher classification performance than using only the labeled data in training or skipping from unsupervised layer-wise pretraining to supervised fine-tuning. Since we have previously found the DBNs to be more sensitive to class imbalance than the other classifiers, we also used a post-training step that shifts the DBN label distributions in order to improve the sensitivity of minority classes.

2.6. Experiment 1: classification performance

One common measure of detection performance, incorporating both a classifier's sensitivity and precision, is the F_1 measure

$$F_1 = 2\frac{\text{sensitivity} \cdot \text{precision}}{\text{sensitivity} + \text{precision}}. \tag{5}$$

For our multi-class problem, we reduced each class's recall and precision calculations to a one-against-others problem. In this study, we first find the mean F_1 for each class over all the partitions and then take the mean of the class F_1s as our single metric of classifier performance.

We measured the mean F_1 values for each of the four classifiers and over each of the three datasets.

2.7. Experiment 2: classification times

With an eye toward real-time monitoring, we measured the classification time of 1 s of multi-channel EEG (17 channels with our dataset) for each classifier. We ran 100 trials and calculated the median time each for each classifier over each of the three datasets.

2.8. Experiment 3: raw data versus features in DBN anomaly measurement

We evaluated how suitable *raw256* data were compared with *feat16* data for measuring the degree of anomaly using an autoencoding DBN. Since the non-background classes are all relatively rare (see table 2), we aggregate them to form our 'anomaly' class. We make no assumptions about how difficult these anomaly waveforms are to learn versus the background waveforms.

In comparing the DBN's ability to measure degree of anomaly using the *feat16* and *raw256* datasets, we looked at the differences between the medians of each class conditional distribution (i.e. anomaly or background) of RMSE values. We denote these differences as γ_{feat16} and γ_{raw256}. We calculated the DBN RMSE for *feat16* and *raw256* and found γ_{feat16} and γ_{raw256} over 1000 trials, each time resampling half of the validation samples from the first partition. We used a

---

[5] The minimum and maximum values of each sample are the scaling parameters. If a parameter value x is in the range [0, 1], we encode it as $[1, \frac{1}{2} + \frac{x}{2}]$. If x is not in the range [0, 1], we encode it as $[0, \frac{1}{2} + \frac{1}{2x}]$.

[6] We exclude the input layer and the classification output layer when referring to the number of DBN layers.
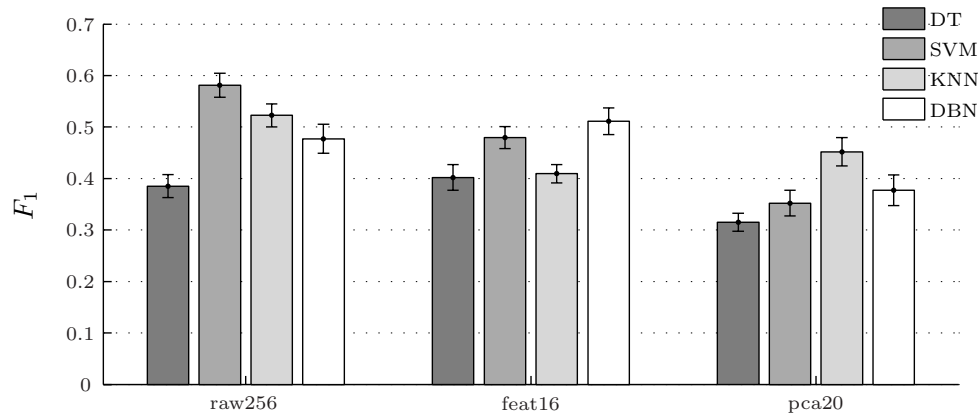
Figure 3. Average F_1 classification performance on the *raw256*, *feat16* and *pca20* datasets for each classifier with standard deviation error bars.
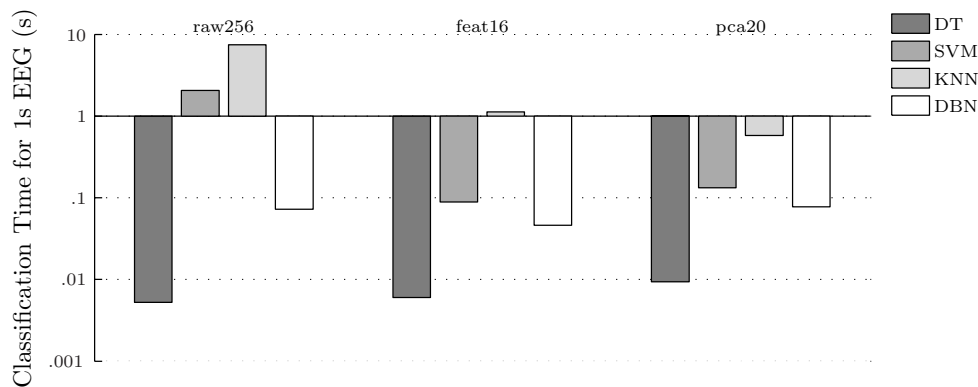
Figure 4. Median time for each classifier to test 1 s of EEG data (17 channels) for the *raw256*, *feat16* and *pca20* datasets. Note that the *y*-axis scale is in powers of 10.

Wilcoxon–Mann–Whitney test to test the null hypothesis that the values of γ_{feat16} and γ_{raw256} were drawn from the same distribution.

We visualized the anomaly measurements for samples in a larger (10 s, 10 channels) segment of EEG by first reconstructing each channel-second sample in the *raw256* form using a DBN autoencoder. Successive channel-second reconstructions of each channel were concatenated to form a 10 s long reconstruction of the 10 s long original signal in each channel. We then computed the sliding RMSE between the original and reconstructed signals in each channel using a symmetric 1 s sliding window with 62.5 ms (eight data points) overlap. The RMSE values were then superimposed onto the original EEG clip using a heatmap, which modifies the color behind each sample point based on its RMSE value.

3. Results

3.1. Experiment 1: classification performance

Figure 3 shows the average F_1 measures across the classes and partitions for each of the four classifiers in each of the three datasets. The DT consistently performs worse than the other three classifiers, but among SVM, KNN and DBN, there seems no clear standout, as each performs best on one of the three test datasets. For the comparison of the *raw256* to *feat16*

and *pca20* datasets, the performance of the four classifiers in *raw256* seems at least as good as that in the *feat16* and better in the *pca20*. The classifier that performed best in each of the three datasets has consistently high performance across the classes of each dataset (see figure 1 in the supplementary material, available at stacks.iop.org/JNE/8/036015/mmedia). Of the four non-background waveform classes, the SVM, KNN and DBN classifiers on average performed best on the eye blink class and worst on the spike and sharp wave class.

3.2. Experiment 2: classification times

Figure 4 shows the median classification time for each of the four classifiers on each of the three datasets for 1 s of EEG with all channels (17 individual channels). As expected, the DT is much faster than the other three, but the DBN is consistently faster than the KNN (7.5× to 103.7×) and the SVM (1.7× to 28.5×) across the three datasets. When the dimensionality of the data is low (as in the *feat16* and *pca20* datasets), the SVM is fairly fast, but in the *raw256* data it becomes quite slow. We also note how little the dimensionality of the input data seems to affect the DBN classification time. Of the three highest performing classifiers, the DBN has consistently faster classification time, especially in the higher-dimensional *raw256* data, where the time difference is between one and two orders of magnitude.
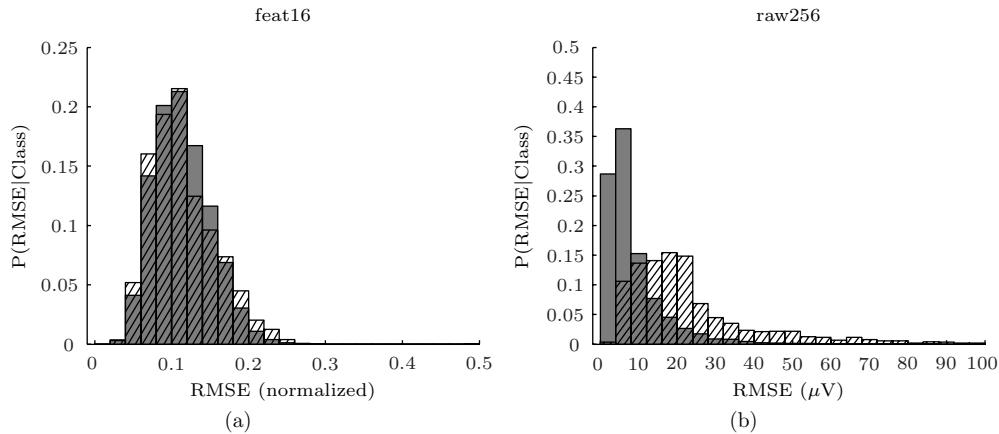
Figure 5. Histogram estimates of the class-conditional probability density functions of the DBN reconstruction error (RMSE) for the background (solid) and non-background (hatched) classes in the (a) *feat16* and (b) *raw256* datasets.

3.3. Experiment 3: raw data versus features in DBN anomaly measurement

With the Wilcoxan–Mann–Whitney test, we found that we can reject the null hypothesis that the differences γ_{feat16} and γ_{raw256} between the median RMSE values of the background and non-background samples of the *feat16* and *raw256* datasets is not significant ($p \ll 0.001$). We observe this difference qualitatively in figure 5, which shows one data partition's RMSE conditional probability density function estimates for the background and non-background classes in the *feat16* and *raw256* datasets.

Figure 6 shows three representative clips of EEG with the anomaly measurement for each signal superimposed as a heatmap. As expected, samples the human marker independently labeled as belonging to one of the four non-background classes generally have higher RMSE values. In the bottom plot after the second burst of spikes, we also see examples of signals that were not classified as one of the clinically relevant classes but are clearly unusual compared to the low-amplitude background. While in the top and bottom clips the anomalous signals seem to have a clear amplitude difference from the background, the middle clip shows samples where non-background has amplitude and wave morphologies fairly similar (to the untrained eye) to background. The DBN's RMSE metric nevertheless still distinguishes between these seemingly similar samples in a way that seems consistent with many of the human labels.

In our previous work [38], we have shown that this DBN anomaly metric can be turned into a positive detection by learning an optimal threshold value. Such anomaly detection outperformed a state-of-the-art one-class SVM at the same task.

4. Discussion

Our experiments are motivated by the clinical need for high-performance classification and anomaly measurement of EEG signals in an online, continuous-monitoring environment. The field of EEG review has no accepted minimum standards both for sensitivity and precision of event detections as well

as anomaly measurement. We see this study as an initial investigation into multi-class epileptiform discharge detection. We did not use sensitivity and precision as benchmarks for performance because which of these measures is optimized depends greatly on the application. For example, detecting epileptic seizures might require high sensitivity and lower precision so as to avoid missing any seizures, even at the expense of a higher false positive rate. Other features of interest, such as episode rates of frontal slowing as a measure of dissipation of sedating medications, might emphasize a higher precision at the cost of sensitivity. We chose to use a balance of these two performance measures, F1, as an objective way to compare classifier performance without emphasizing a specific application. We have as much as possible avoided the preselection and filtering that often occurs in EEG event detection literature, again with an eye toward online monitoring, where channel and 'clean signal' preselection is impossible. That said, we have not yet implemented our methods in a clinical trial, and so can only estimate the standards we must meet in order for them to be practical and used.

4.1. Comparing classifiers

Human labeling of specific waveforms is itself a very difficult task, and the inter-rater reliability among board-certified neurophysiologists is usually low [18, 21, 22]. For computational simplicity, we constrained our classifiers to only examine individual channel-seconds, when in reality experts incorporate much greater spatial and temporal information in their markings. These factors kept our expectations modest for the ultimate recall and precision of our classifiers.

In our classification performance experiment, a different classifier was the highest performing in each of the three datasets with only the DT performing worst in all. We therefore argue that DBNs have classification performance competitive with the two other high-performing classifiers, SVMs and KNNs, for our EEG data. To our knowledge, the only other studies comparing DBNs to SVMs on a head-to-head classification task have been in handwritten digit
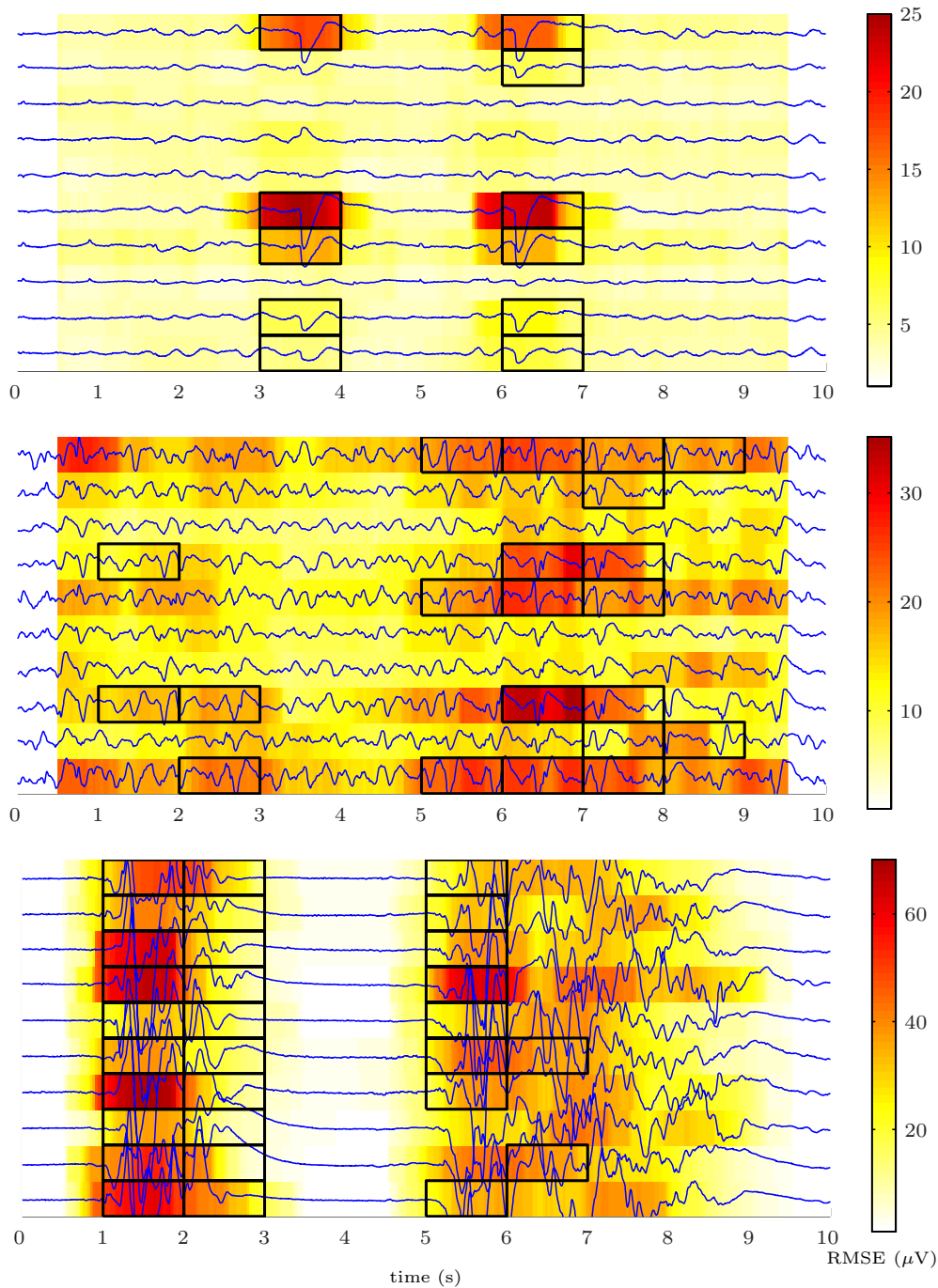
Figure 6. Anomaly visualizations using DBN RMSE on three representative example 10 s clips of ten EEG channels. The color behind a point on a given channel represents the windowed (1 s, 62.5 ms window overlap) RMSE centered around that point. More anomalous areas of the signal have higher RMSE and are redder in color. Samples that a human reviewer independently labeled as non-background are boxed. The top sample shows examples of eye-blink artifact; the middle shows examples of GPEDs and triphasic waves; and the bottom shows examples of high-amplitude spikes. The height of the black boxes represents 90 μV.

recognition [11, 23], and in those studies DBNs slightly outperformed SVMs overall.

As previously mentioned, we have found that DBNs can be sensitive to heavy class-imbalance, which occurs in our dataset, and have used a simple post-training technique that improves the sensitivity of the minority classes. In some ways this step is unfair to the other classifiers that did not receive such a post-training step. Similar post-training steps for the

other three classifiers are not entirely straightforward, so we did not undertake them in this study. Nevertheless, we see this problem as a necessary area of future work for our group and others.

In our classification time experiment, the DBN was significantly faster than the SVM and the KNN though slower than the DT. Notably, DBN and DT classification times are relatively consistent as the dimensionality of the data changes

between the three datasets. We see this experiment as a first attempt to find which classifiers are more likely to efficiently process the continuous streams of multi-channel EEG data that can sometimes reach hundreds of channels at once.

An important goal of this research was to develop a classification technique that can operate in real time on raw data in a clinical environment. We found that the DBN was the only classification technique we explored that was able to do this. While DBN performance was slightly worse than both the SVM and KNN classifiers on raw data, neither of these techniques are implementable at present on raw data streams in a real-time clinical environment. Of note, the DBN performed better on features we extracted in this experiment than either the SVM or KNN. While DBN performance did not substantially reduce the high false positive rates that limit existing detectors, we did not optimize detectors for this task, something we will be investigate in future work.

Given that DBNs can also learn most of their model in the unsupervised phase of learning, the large amounts of unlabeled EEG data may also be conducive to the semi-supervised DBN training paradigm and allow it to learn more sophisticated models than otherwise possible with traditional supervised learners. In addition to the DBN's fast execution time and ability to learn from unlabeled data, the anomaly measurement capability that naturally falls out of the DBN training paradigm is another advantage of DBNs over SVMs and KNN.

4.2. Comparing data representations

In our exploration of using raw data instead of hand-chosen features, we found that in classification F_1, the raw data yielded results comparable to the feature's data. We made our best attempt to be as rigorous and thorough as possible in feature selection. Nevertheless, it is certainly possible that in spite of our best efforts, the features we selected were not optimal for this specific classification task. That said, optimizing features based on very specific domain knowledge is precisely what we would like to move away from. If using raw data with sophisticated learning algorithms gives just as good performance as using features, we believe this move is an improvement, both in increased methodological elegance and a reduced reliance on ultra-specific domain knowledge.

It appears that the top classifiers did not fall victim to the curse of dimensionality with the *raw256* data, perhaps since they had a relatively large amount of labeled training data (72 800 samples, see table 3). In classification time, the SVM and KNN were much slower than with the lower dimensional *feat16* and *pca20* data, but the DBN maintained relatively constant speed regardless of the input dimensionality. This result makes sense given that DBN classification consists almost exclusively of a number of matrix multiplications, and the first matrix is the only one that changes with the dimensionality of the input data for a given model.

In the DBN anomaly measurement, we showed with the Wilcoxon–Mann–Whitney test and also in figure 5 that the raw data have significantly better separation between the background and non-background (anomaly) classes versus the features data. This result, combined with the previous finding that raw data seems on average no worse for the classification task, indicates that using raw data instead of features for EEG processing may have some advantages in both methodological elegance and better anomaly measurement.

4.3. Novel multi-class EEG waveform classification

We have focused throughout this study on using different classifiers and different data representations for our EEG waveform classification task. We note that—to the best of our knowledge—no other classification paradigm exists for the five clinically significant EEG waveform classes we used in this study. The most recent variant [2] of the commonly used spike-detection algorithm originally proposed by Gotman *et al* [24] has a reported recall range of 0.09–0.34 and a false positive rate range of 4.2–48.6 per hour [25]. While the authors of this study do not report their results in terms of F_1, we believe that our DBN classifier on the *raw256* data, which has 0.2271 recall and 0.1920 precision in the spike and sharp waves class, is competitive with this existing clinical standard, especially since the DBN also detects three other types of clinically important signals at the same time. To our knowledge, two of the classes we detect, GPEDs and triphasic and PLEDS, have never before been examined for automated classification despite their published significance and use in clinical EEG review. Part of this study's novelty also stems from its being the first to perform automated waveform classification and anomaly measurement in continuous EEG of critically-ill patients. Other small sample studies have restricted themselves to seizure detection and changes in time–frequency parameters of EEG [26].

4.4. Limitations and future work

We did not thoroughly examine the training time of the different models in this paper, but it deserves a mention. The training time of the KNN and SVM models took anywhere from a few hours to a few days, depending on the dataset, and that of the DBNs took anywhere from a few days to more than a week[7]. DBN training is an important consideration when applying it to practical applications in the clinical arena, such as the ICU. Since the waveforms of interest in the ICU, such as epileptiform discharges and frontal slowing, can have great similarity from patient to patient, training on one representative data set, one time, would likely be adequate for application to many patients. In cases where the detector would need to be trained to a rare pattern in one individual, the incremental training times will be much lower because the model has already been initialized, which avoids its having to redo the first two training stages that take up the bulk of the total

---

[7] While it would be interesting to look at the variation of training time between patients, because of the relatively low number of patients in this study necessitated by time intensive physician data marking, we looked at training and testing on an aggregate basis only.

training time. These techniques were not investigated for this study, but are fertile areas for future work. The field of DBN learning is also still very young, and the training regime is still very un-optimized. With optimization improvements and the use of relatively cheap GPU processors (which have already been shown to decrease RBM training by up to a factor of 72 [27]), RBM and DBN training times will certainly decrease.

Each of the four classification algorithms has variants that may have faster testing time given certain assumptions about the data[8]. We could not implement and test all of these in our initial experiments but plan to explore them in future ones.

Epileptiform discharges such as GPEDs and PLEDs are usually characterized by their spatial patterns across one or both sides of the brain. In this initial work, we explored how well classifiers are able to separate these signals using only information from one second of one channel and found that they are able to distinguish them reasonably well[9]. Nevertheless, we are confident that increasing the amount of information in space (channels) and time (more than one second) would improve classification performance. We are currently pursuing this next iteration for improving the classification performance.

This study finds that DBNs and raw data may be most appropriate for both online and retrospective clinical EEG monitoring and data mining tasks, but we need to apply these methods in the clinical setting to truly understand the classification performance and test time requirements. Our future work will involve training these models on a larger patient population as well as incorporating more spatial and temporal area of the EEG.

While our results indicate that the DBN computational performance is such that it could be applied in a 'real' clinical context, more research will be necessary to optimize this process. Using raw data as input to the DBN makes incorporating new patient data fairly easy, as little preprocessing is necessary. For applications on extracted features, computational overhead will increase as the number and/or complexity of the features increases. Regarding supervision, for some applications a single training set may be applicable to multiple patients, though rare, patient-specific patterns may require more individualized training, which will require some optimization. We will also explore online DBN learning, in which new, unlabeled patient data can be incorporated into the DBN model. One common online learning paradigm is to use the existing classifier to assign labels to new data and then use those samples labeled with high confidence to further train the model. Finally, we expect to work on more specific applications of this method that require fine-tuned sensitivity and precision metrics. We expect to

work closely with practicing clinicians to better understand and test the requirements of these systems.

5. Conclusion

In this paper, we explored new classification and anomaly measurement techniques for multiple classes of clinical EEG waveforms. We showed that a relatively new type of neural network, the deep belief net, has comparable performance with SVM and KNN classifiers and has fast test time, even on high-dimensional raw data. We also showed that using unpreprocessed, raw input data instead of features can yield comparable classification performance with greatly increased methodological elegance. Finally, we described how DBNs can be used for signal anomaly measurement and show that raw data are significantly better than features for this task. These experiments result in a five-class EEG waveform classification system that, to our knowledge, is the first automated classifier for two of the four clinically significant waveform classes considered. It is also the first to measure performance of an automated waveform classification and anomaly measurement algorithms in continuous EEG of critically-ill patients.

These experiments show that fast classification and anomaly measurement of EEG waveforms are possible with sophisticated machine learning methods such as deep belief nets. As the amount of clinical monitoring continues to increase throughout the world, the necessity of such fast, powerful algorithms to process and interpret these data becomes evermore acute.

Appendix A. Classifier implementation details

Software. We used Matlab implementations of the KNN and DT classifiers (`knnclassify` and `classregtree`) and LIBSVM [29] package for our SVM experiments. We used the DBN implementation in our object-oriented DBNToolbox_v1.0[10] in Matlab. All code and experiments were done in Matlab 2010a (although they should work with at least 2009a and 2009b and possibly earlier versions).

---

[8] For example, the nearest neighbor principal axis search tree (PAT) algorithm which partitions the training data into a tree based on dimensions of maximal variance, has been shown to improve testing time [28] with some datasets. We implemented PAT but found that test time was not consistently faster than that of the standard KNN implementation used by Matlab for k values ($k < 5$) offering good performance on our EEG datasets.

[9] The best F_1 value for the GPED & triphasic class was 0.55 and that for the PLED class was 0.41.

[10] available at www.seas.upenn.edu/~wulsin

Table A1. Parameter search and optimal values.

Classifier	Search	Top dataset values		
		raw256	*feat16*	*pca20*
DT	$L = \{2, 4, 6, 8, 10, 12, 14, 16\}$	8	10	8
SVM	$C = 2^{\{3,5,7\}}$	2^3	2^7	2^5
	$\gamma = 2^{\{-3,-1,1\}}$	2^{-3}	2^1	2^{-1}
KNN	$k = \{1, 2, 3, 4, 5, 6, 7, 8\}$	4	2	4
DBN	Layer1 = {250, 500}	500	250	500
	Layer2 = {250, 500}	250	500	250
	Layer3 = {250, 500}	250	250	250

Table A2. Major parameters held constant for DBN training.

Parameter	Value
Layer 4 size	1000
RBM learning rate	0.1
Number of mini batches combined in fine-tuning	10
Number of (unsup) pretraining epochs	50
Number of (unsup) fine-tuning epochs	50
Number of (sup) fine-tuning epochs	200

Parameters. Table A1 summarizes the parameters searched and the optimal values found for each classifier and each dataset. Initial experimentation with different parameter values informed our search for each classifier.

In DT training, we searched the minimum number L of samples in a node required for it to split. In SVM training, we searched the cost C and γ of the Gaussian kernel. In KNN training, we searched the number of neighbors k. In DBN training, we searched the number of hidden units in the first three RBMs, choosing to keep the fourth layer's size constant at 1000. There are too many other tunable parameters to list here (see `DeepNN.m` for a complete list and descriptions), but some of the major ones are given in table A2.

Appendix B. Hand-chosen features

For the features below, we consider samples with N raw dimensions.

Area. Area [30, 31] describes the normalized positive area under the curve:

$$\text{Area} = \frac{1}{N} \sum_{i=1}^{N} |x_i|.$$

Normalized decay. Normalized decay describes the chance-corrected fraction of data that is decreasing or increasing. The function $I(x)$ is the indicator function, which is 1 when the argument is true and 0 when it is false:

$$\text{Normalized decay} = \left| \frac{1}{N-1} \sum_{i=1}^{N-1} I(x_{i+1} - x_i < 0) - 0.5 \right|.$$

Frequency band power. Frequency band power [32, 33] is calculated using Welch's averaged modified periodogram method of power spectral density estimation (`pwelch` function in Matlab). For these calculations, the data were divided into eight segments, each with 50% overlap (default values), and each segment was windowed with a Hamming window. To calculate the band powers from the resultant spectral density vector, the power spectral density values were averaged over the frequency bands 1.5 Hz–7 Hz, 8 Hz–15 Hz and 16 Hz–29 Hz, yielding a feature value for each band.

Line length. Line length [15, 34] describes the sum of the absolute differences between successive data points:

$$\text{Line length} = \sum_{i=2}^{N} |x_i - x_{i-1}|.$$

Mean energy. Mean energy [22] describes mean energy across the data:

$$\text{Mean energy} = \frac{1}{N} \sum_{i=1}^{N} |x_i^2|.$$

Average peak amplitude. Average peak amplitude finds the base-10 logarithm of the mean-squared amplitude of each of the K peaks, where a peak is defined as a change from positive to negative in the signal derivative sign. Let $P(i)$ be the index of the ith peak and $x_{P(i)}$ its value:

$$\text{Peak amplitude} = \log_{10} \left(\frac{1}{K} \sum_{i=1}^{K} x_{P(i)}^2 \right).$$

Average valley amplitude. Average valley amplitude finds the base-10 logarithm of the mean-squared amplitude of each of the K valleys, where a valley is defined as a change from positive to negative in the signal derivative sign. Let $V(i)$ be the index of the ith valley and $x_{V(i)}$ its value:

$$\text{Peak amplitude} = \log_{10} \left(\frac{1}{K} \sum_{i=1}^{K} x_{V(i)}^2 \right).$$

Normalized peak number. Normalized peak number [35, 36] describes the number of peaks (K) present normalized by the average difference between adjacent data point values:

$$\text{Normalized peak number} = K \left(\frac{1}{N-1} \sum_{i=1}^{N-1} |x_{i+1} - x_i| \right)^{-1}.$$

Peak variation. Peak variation [31, 35] describes the variation between peaks and valleys across both time and values of the data. The mean and standard deviation of the indices for the K peaks and valleys are given by

$$\overline{D}_{\mathrm{PV}} = \frac{1}{K} \sum_{i=1}^{K} P(i) - V(i)$$

$$\sigma_{\mathrm{PV}} = \sqrt{\frac{1}{K-1} \sum_{i=1}^{K} (P(i) - V(i) - \overline{D}_{\mathrm{PV}})^2}$$

and for the sequential value difference are

$$\overline{D}_{x_{\mathrm{PV}}} = \frac{1}{K} \sum_{i=1}^{K} x_{P(i)} - x_{V(i)}$$

$$\sigma_{x_{\mathrm{PV}}} = \sqrt{\frac{1}{K-1} \sum_{i=1}^{K} (x_{P(i)} - x_{V(i)} - \overline{D}_{x_{\mathrm{PV}}})^2}.$$

The peak variation is defined as

$$\text{Peak variation} = \frac{1}{\sigma_{\mathrm{PV}} \sigma_{x_{\mathrm{PV}}}}.$$

Root mean square. Root mean square [34] calculates the square root of the mean of the squared data points:

$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} x_i^2}.$$

Wavelet energy. Wavelet energy [37] performs a five-level decomposition of the data using Daubechies' fourth-order wavelets and then calculates the energy across the four frequency bands of 4–8 Hz 8–16 Hz, 16–32 Hz and 32–64 Hz. First, the decomposition high-pass and low-pass filters associated with Daubechies' fourth-order wavelets are obtained using MATLAB's `wfilters` function. The input data (oneDimData) are then convolved with these two filters and downsampled using MATLAB's `dwt` function to obtain approximation and detail coefficients as follows:

Approximation coefficients of oneDimData $[n]$

$$= \sum_{i=-\infty}^{\infty} (x_i h_{2n-i})$$

Detail coefficients of oneDimData $[n] = \sum_{i=-\infty}^{\infty} (x_i g_{2n+1-i})$

where h and g are the impulse responses of the low-pass and high-pass filters, respectively. At each successive level of decomposition, the approximation coefficients from the previous iteration are used as the input to the low-pass and high-pass filters. Since the sampling frequency of our data was 256 Hz, the detailed values from decomposition levels $2-5$ roughly correspond to the frequency bands $4-8$ Hz $8-16$ Hz, $16-32$ Hz and $32-64$ Hz. The energy of each band is calculated using the corresponding detailed values as follows:

$$\text{Wavelet energy of a band} = \frac{\sum_{i=1}^{N} |\text{detail value}_i|^2}{N}.$$

Zero crossings. Zero crossings [36] first detrends the data by subtracting the linear best-fit line from it and then counting how many times the detrended data cross zero.

References

[1] Chong D J and Hirsch L J 2005 Which EEG patterns warrant treatment in the critically ill? Reviewing the evidence for treatment of periodic epileptiform discharges and related patterns *J. Clin. Neurophysiol.* **22** 79–91

[2] Flanagan D, Agarwal R, Wang Y H and Gotman J 2003 Improvement in the performance of automated spike detection using dipole source features for artefact rejection *Clin. Neurophysiol.* **114** 38–49

[3] Wilson S B and Emerson R 2002 Spike detection: a review and comparison of algorithms *Clin. Neurophysiol.* **113** 1873–81

[4] Hodge V and Austin J 2004 A survey of outlier detection methodologies *Artif. Intell. Rev.* **22** 85–126

[5] Polat K and Gunes S 2007 Classification of epileptiform EEG using a hybrid system based on decision tree classifier and fast Fourier transform *Appl. Math. Comput.* **187** 1017–26

[6] Abuhanna A and Dekeizer N 2003 Integrating classification trees with local logistic regression in intensive care prognosis *Artif. Intell. Med.* **29** 5–23

[7] Demsar J 2001 Feature mining and predictive model construction from severe trauma patient's data *Int. J. Med. Inform.* **63** 41–50

[8] Breiman L, Friedman J H, Olshen R A and Stone C J 1984 *Classification and Regression Trees* (*The Wadsworth Statistics/Probability Series*) (Boca Raton, FL: Wadsworth)

[9] Gardner A B, Krieger A M, Vachtsevanos G and Litt B 2006 One-class novelty detection for seizure analysis from intracranial EEG *J. Mach. Learn. Res.* **7** 1044

[10] Bishop C M 2006 *Pattern Recognition and Machine Learning* (*Information Science and Statistics* vol 16) (Berlin: Springer)

[11] Hinton G E, Osindero S and Teh Y W 2006 A fast learning algorithm for deep belief nets *Neural Comput.* **18** 1527–54

[12] Bengio Y and LeCun Y 2007 Scaling learning algorithms towards AI *Large-Scale Kernel Machines* vol 1 (Cambridge, MA: MIT Press) pp 1–41

[13] Hinton G E 2002 Training products of experts by minimizing contrastive divergence *Neural Comput.* **14** 1771–800

[14] Hinton G E and Salakhutdinov R R 2006 Reducing the dimensionality of data with neural networks *Science* **313** 504–7

[15] Esteller R, Echauz J, Tcheng T, Litt B and Pless B 2001 Line length: an efficient feature for seizure onset detection *Proc. 23rd EMBS Conf.* vol 3, 1707–10

[16] Löfhede J, Thordstein M, Löfgren N, Flisberg A, Rosa-Zurera M, Kjellmer I and Lindecrantz K 2010 Automatic classification of background EEG activity in healthy and sick neonates *J. Neural Eng.* 7 16007

[17] D'Alessandro M, Vachtsevanos G, Esteller R, Echauz J, Cranstoun S, Worrell G, Parish L and Litt B 2005 A multi-feature and multi-channel univariate selection process for seizure prediction *Clin. Neurophysiol.* **116** 506–16

[18] Blanco J A, Stead M, Krieger A, Viventi J, Richard Marsh W, Lee K H, Worrell G A and Litt B 2010 Unsupervised classification of high-frequency oscillations in human neocortical epilepsy and control patients *J. Neurophysiol.* **104** 2900–12

[19] Ebersole J S and Pedley T A 2003 *Current Practice of Clinical Electroencephalography* 3rd edn (Baltimore, MD: Williams & Wilkins)

[20] Le Roux N and Bengio Y 2008 Representational power of restricted Boltzmann machines and deep belief networks *Neural Comput.* **20** 1631–49

[21] Gerber P A, Chapman K E, Chung S S, Drees C, Maganti R K, Ng Y-T, Treiman D M, Little A S and Kerrigan J F 2008 Interobserver agreement in the interpretation of EEG patterns in critically ill adults *J. Clin. Neurophysiol.* **25** 241–9

[22] Gardner A B, Worrell G A, Marsh E, Dlugos D and Litt B 2007 Human and automated detection of high-frequency oscillations in clinical intracranial EEG recordings *Clin. Neurophysiol.* **118** 1134–43

[23] Larochelle H, Erhan D, Courville A, Bergstra J and Bengio Y 2007 An empirical evaluation of deep architectures on problems with many factors of variation *Proc. 24th Int. Conf. on Machine Learning*, number 2006 (ACM) p 480

[24] Gotman J and Gloor P 1976 Automatic recognition and quantification of interictal epileptic activity in the human scalp EEG *Electroencephalogr. Clin. Neurophysiol.* **41** 513–29

[25] Ver Hoef L, Elgavish R and Knowlton R C 2010 Effect of detection parameters on automated electroencephalography spike detection sensitivity and false-positive rate *J. Clin. Neurophysiol.* **27** 12–6

[26] Shah A K, Agarwal R, Carhuapoma J R and Loeb J A 2006 Compressed EEG pattern analysis for critically ill neurological-neurosurgical patients *Neurocritical Care* **5** 124–33

[27] Raina R, Madhavan A and Ng A Y 2009 Large-scale deep unsupervised learning using graphics processors *ICML 2009* pp 1–8

[28] McNames J 2001 A fast nearest-neighbor algorithm based on a principal axis search tree *IEEE Trans. Pattern Anal. Mach. Intell.* **23** 964–76

[29] Chang C C and Lin C J 2001 LIBSVM: a library for support vector machines

[30] Agarwal R, Gotman J, Flanagan D and Rosenblatt B 1998 Automatic EEG analysis during long-term monitoring in the ICU *Electroencephalogr. Clin. Neurophysiol.* **107** 44–58

[31] Qu H and Gotman J 1997 A patient-specific algorithm for the detection of seizure onset in long-term EEG monitoring: possible use as a warning device *IEEE Trans. Biomed. Eng.* **44** 115–22

[32] Liu A, Hahn J S, Heldt G P and Coen R W 1992 Detection of neonatal seizures through computerized EEG analysis *Electroencephalogr. Clin. Neurophysiol.* **82** 30–7

[33] Alarcon G, Binnie C D, Elwes R D and Polkey C E 1995 Power spectrum and intracranial EEG patterns at seizure onset in partial epilepsy *Electroencephalogr. Clin. Neurophysiol.* **94** 326–37

[34] Greene B R, Faul S, Marnane W P, Lightbody G, Korotchikova I and Boylan G B 2008 A comparison of quantitative EEG features for neonatal seizure detection *Clin. Neurophysiol.* **119** 1248–61

[35] Pietilä T, Vapaakoski S, Nousiainen U, Värri A, Frey H, Häkkinen V and Neuvo Y 1994 Evaluation of a computerized system for recognition of epileptic activity during long-term EEG recording *Electroencephalogr. Clin. Neurophysiol.* **90** 438–43

[36] van Putten M J A M, Kind T, Visser F and Lagerburg V 2005 Detecting temporal lobe seizures from scalp EEG recordings: a comparison of various features *Clin. Neurophysiol.* **116** 2480–9

[37] Osorio I, Frei M G and Wilkinson S B 1998 Real-time automated detection and quantitative analysis of seizures and short-term prediction of clinical onset *Epilepsia* **39** 615–27

[38] Wulsin D, Blanco J, Ram M and Litt B 2010 Semi-supervised anomaly detection for EEG waveforms using deep belief nets *9th Int. Conf. Machine Learning and Applications* **9** 436–41