

Applying Machine Learning for Human Seizure Prediction

Ehsan Dadgar-Kiani
Bioengineering
Stanford University

dadgarki@stanford.edu

Cagan Alkan
Electrical Engineering
Stanford University

calkan@stanford.edu

Ali Shameli
Management Science and Engineering
Stanford University

shameli@stanford.edu

Abstract

We explore the use of both statistical and deep learning methods for the task of classifying iEEG signals as pre-ictal or inter-ictal in order to perform human seizure prediction. Features were extracted that captured both the frequency and temporal properties of the raw data. The classifier performed adequately well by several performance metrics, especially given that the data set suffered from a significant class imbalance, with the majority of samples representing non-seizure events. Thus, this paper demonstrates the feasibility of using a machine learning algorithm to accurately predict rare seizure events from a large set of EEG data. Such a model could have a significant impact on medicine, for its implementation on a medical device could predict and inform patients of seizure onset with a very high recall rate.

1. Introduction

Epilepsy is a neurological condition associated with sporadic abnormal electrical activity ('seizures') in the brain. It affects around 1% of the world's population. Due to its risk of creating serious injuries, it is important to identify the periods when the patients are more likely to have seizures. If computational methods can reliably predict these seizure occurrences, patients can avoid dangerous activities or take medications only when it is necessary. Successful prediction of seizures can have a positive impact on epilepsy patients' lives. Our proposed method is to use several machine learning methods to train models for predicting seizures from human iEEG (Intracranial Electroencephalography) recordings. More specifically, given a 10 minute recording, our purpose is to classify it as either 'preictal' (pre-seizure) or 'inter-ictal' (non-seizure).

Deep learning has recently gained popularity for model training tasks with large amounts of data. Since it is likely a sequence of ordered physiological events that are actually causing the seizure phenotype in patients, it is fruitful to experiment with recurrent neural networks. Recurrent neural

networks, such as LSTMs, are very well-suited for tasks involving time-series data where you want to keep track of a hidden state throughout a full sequence of inputs. The effectiveness of this deep learning technique can then be compared to statistical machine learning techniques such as Logistic Regression and Support Vector Machines (SVM).

2. Related Work

The research on seizure prediction methods has accelerated in the recent years. Mormann et al. [1] described a wide range of measures characterizing the EEG signals in their review paper about seizure prediction. The measures include statistical moments of the time series, frequency domain properties and features obtained by state space representation of EEG signals motivated by linear dynamical systems. Among these features, spectral power levels in different EEG bands are commonly used in many seizure prediction studies. The main advantage of spectral power based features is that their significance in EEG studies is well established by the neuroscience community. Typically, each of the features are obtained by breaking the several minutes long EEG recording into segments of 10-60 seconds. In some studies, several preprocessing techniques such as undersampling the data, filtering the time series with a filter that has a desired frequency response or windowing each segment with an appropriate function have been applied. In a recent study on forecasting canine seizures, Generalized Linear Models (GLM), Support Vector Machines (SVM), Random Forests and Convolutional Neural Networks (CNN) were trained using a subset of the features listed above and they are shown to have area under the receiver operating characteristics curve (AUROC) scores in the 0.82-0.86 range [2].

3. Data Description

Data was recorded in the form of a 10 minute long iEEG, which is obtained by positioning electrodes on the surface of cerebral cortex and measuring electrical signals sampled at 400 Hz. The data was available as part of an online com-

petition through Kaggle.com [3]. We used data from a single patient (out of three possible patients). There are 16 channels corresponding to different locations of the electrodes, hence each 10 minute recording can be represented by a matrix where the row represents time, and column represents electrode channel (out of 16), and is also labeled as one of the two classes, preictal (1) or interictal (0). Number of samples in each recording is therefore $400 \text{ Hz} \times 600 \text{ sec} \times 16 \text{ channels} \approx 4 \text{ million}$.

The patient we selected has 2058 recordings in total. We should note that there is a significant imbalance between pre-ictal (150) and inter-ictal (1908) examples.

4. Methods

4.1. Feature Extraction

After reviewing the literature on seizure prediction, we found out that the most important measure that characterizes EEG signals is the spectral power in 6 physiological frequency bands [1, 2, 4]. These bands are delta (δ : 0.5-4 Hz), theta (θ : 4-8 Hz), alpha (α : 8-12 Hz), beta (β : 12-30 Hz) and gamma (γ : 30-100 Hz). We first divided each 10 minute recording into 1 minute consecutive windows, then calculated the spectral power for each band on each 1 minute segment. Since each recording consists of data from 16 channels, and each power calculation represents one feature, this results in a feature size of $6 \text{ bands} \times 10 \text{ segments} \times 16 \text{ channels} = 960$ features per recording. We stacked the features from every 1 minute segment into one big feature vector for each 10 minute recording, resulting in a feature size of $96 \text{ features} \times 10 \text{ segments} = 960$ features per recording.

This is a reduction from 4 million time points in the raw data. Power in each frequency band is calculated by taking the Fast Fourier Transform (FFT) and summing up squares of the magnitudes of FFT coefficients corresponding to each frequency band mentioned above (Figure 1). The main advantage of this feature extraction scheme is that it gives information about both the time course and the frequency spectrum of the iEEG signals.

4.2. Models

After extracting our features from each recording, we used the models described below to do classification on human seizure data. These were implemented in Python using the Scikit-Learn and Keras packages. All models were trained using 70-30 hold-out cross validation.

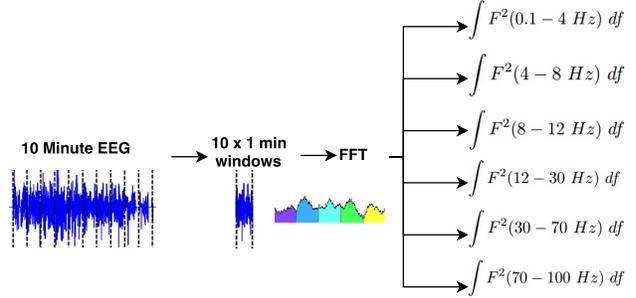


Figure 1: Feature extraction scheme. First, a 10 minute EEG is split into 10 one minute chunks. We then do a FFT on each chunk and obtain 6 features corresponding to 6 different frequency bands.

4.2.1 Logistic Regression

In logistic regression, the hypothesis has the form of a logistic function:

$$h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

We used L2 regularized logistic regression to have a control over the bias-variance tradeoff. The cost function and the corresponding minimization problem for logistic regression is then

$$\min_{\theta} \frac{1}{2} \theta^T \theta + C \sum_{i=1}^m \log(1 + \exp(-y^{(i)} x_i^T \theta))$$

where θ is the parameter vector to be optimized, m is the number of samples, x_i is the vector of features for the i^{th} sample, y is the vector of labels and $C > 0$ is the regularization parameter. Note that the expression above uses the convention that $y^{(i)} = \pm 1$ although our labels are 1 and 0. We tuned regularization parameter C to increase the performance.

4.2.2 Linear Support Vector Machines (SVM)

Support vector machines use hinge loss to find the optimal parameters.

$$\varphi_{hinge}(z) = [1 - z]_+ = \max\{0, 1 - z\}$$

We used linear SVM with L2 regularization which tries to minimize

$$\min_{\theta} \frac{1}{2} \theta^T \theta + C \sum_{i=1}^m \max\{0, 1 - y^{(i)} x_i^T \theta\}$$

where $C > 0$ is the regularization parameter we try to tune to increase performance.

4.2.3 SVM with Radial Basis Function (RBF) Kernel

The kernel function used in RBF SVM is the Gaussian kernel

$$K(x, z) = \exp(-\gamma \|x - z\|_2^2)$$

where $\gamma > 0$ controls the bandwidth of the kernel. We used L2 regularization to generalize better to unseen data. The cost function and the corresponding minimization problem for RBF SVM with L2 regularization is

$$\min_{\alpha} \frac{1}{2} \alpha^T K \alpha + C \sum_{i=1}^m \max \left\{ 0, 1 - y^{(i)} K^{(i)T} \alpha \right\}$$

where C is the regularization parameter and the matrix $K = [K^{(1)} \dots K^{(m)}]$ is defined by $K_{ij} = K(x^{(i)}, x^{(j)})$. Note again that in the expression above, we use the convention that $y^{(i)} = \pm 1$ and the optimization variable α is the coefficient vector used in Representer Theorem. For RBF SVM, both γ and C control the bias-variance trade-off, hence we tune them to increase the performance.

4.2.4 Long Short-Term Memory (LSTM)

In this section we describe Recurrent Neural Networks (RNN) and how we incorporate them into our predictive model. Traditional neural networks, in general, don't have the ability to maintain the information and reasoning obtained in the past to make more effectively predictions in the future. This seems like a major shortcoming and RNNs address this issue. They are essentially normal networks with loops in them which allows information to persist in them. A RNN can be thought of as multiple copies of the same network, each passing information to a successor. RNNs have been very successful in various tasks such as speech recognition, language modeling, image captioning, etc. Essential to these successes is the use of Long Short Term Memory (LSTM) networks, a very special kind of RNN that works, for many tasks, much better than the standard version. LSTMs are capable of learning long term dependencies. They were first introduced by [5] and were refined and improved by many successive works. LSTMs, like standard RNNs, have a chain like structure. The repeating module, however, has a different structure and instead of having a single neural network layer, there are four, interacting in a special way. The key to LSTMs is the cell state, the horizontal line running through the top of the network. The information can flow throughout the network along the cell state. It also has the ability to add or remove information to the cell state using the structures called gates. It looks at h_{t-1} and x_t and outputs a number between 0 and 1 representing how much of the data we are willing to keep (a 1 means completely keep the data). We have:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f).$$

The next step is to decide what cell information are we willing to store in the next step. This process consists of two stages. First, a sigmoid layer called the "input gate layer" decides which values should be updated. Next, a tanh layer creates a vector of new candidate values, \tilde{C}_t , that could be added to the state. We have:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i),$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c).$$

Next step would be to update the value of the old state C_{t-1} into the new state based on what we have already decided do in the previous step. The update rule would be:

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t.$$

Finally, the last step would be to produce an output. We first run a sigmoid layer to decide what parts of the cell state should go to the output. We then put the cell state through a tanh gate to push the values between -1 and 1 and multiply it by the output of the sigmoid gate to make sure we only output the parts we have decided to. The output function is:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o),$$

$$h_t = o_t \times \tanh(C_t).$$

The LSTM architecture that was chosen for this project consisted of a sequence of ten LSTM cells. Each cell, which corresponds to a 1-minute time window, accepts a 64 dimensional input which represents the spectral power across the 16 channels and 6 frequency bands. The final LSTM cell's hidden state is then connected to a fully connected layer with a subsequent softmax function that outputs a single score for classification. Both the internal hidden layer dimensionality of the LSTM cell and the size of the fully connected layer are hyperparameters that were optimized by a grid search using several performance metrics (see Results section for more information). This resulted in a final design with a hidden state size of 100 and fully-connected size of 200. In terms of training this model, stochastic gradient descent with a batch-size of 100 was used to optimize a binary cross-entropy objective.

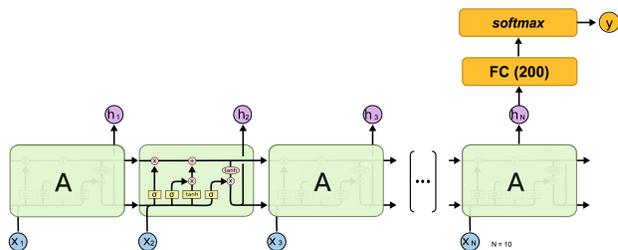


Figure 2: Full LSTM Architecture for Seizure Prediction

Model	Training Acc.	Test Acc.	AUROC	F-Score	AUPRC	Recall
LSTM	0.852	0.854	0.942	0.489	0.534	0.860
Logistic Regression	0.909	0.872	0.938	0.533	0.589	0.818
SVM (RBF)	0.874	0.861	0.923	0.538	0.434	0.909
SVM (Linear)	0.899	0.867	0.923	0.445	0.506	0.764

Table 1: Performance comparison of different models.

5. Results

5.1. Performance Metrics

Due to the strong imbalance between inter-ictal and pre-ictal examples, testing and training accuracy are not the main criteria we used to evaluate the performance of the classifiers. For our problem, it is critical not to miss seizure events (pre-ictal). Therefore, we tried to maximize the True Positive Rate (TPR) or Recall. However, focusing only on Recall may decrease the performance of classification of non-seizure (inter-ictal) events. Therefore, we aim having high Recall values while keeping the False Positive Rate (FPR) reasonably low. For that reason, we selected receiver operating characteristics (ROC) curve and the area under ROC curve (AUROC) as the main metric to evaluate the performance. ROC curve shows the tradeoff between FPR and TPR for different threshold values. Having a higher AUROC indicates better performance. In addition to the relationship between FPR and TPR, we investigated the relationship between sensitivity and recall by plotting the precision-recall curves (PRC) for each classifier and calculating the area under each curve (AUPRC). Finally, we use the F_1 Score, which is the harmonic mean of precision and recall. To summarize, we evaluate our generalization performance by primarily considering AUROC, while trying to achieve higher values in Recall, AUPRC and F_1 Score.

5.2. Hyperparameter Tuning

Figure 5 depicts the range of various model hyperparameters that were searched through to maximize different performance metrics. For Linear SVM, the optimal value was 0.01, while for Logistic Regression it was 0.1. For radial SVM (RBF kernel), a grid search was performed over the Cartesian grid of the model’s regularization parameter and its inverse kernel bandwidth (γ). The optimal values were found to be $\gamma = 0.001$ and $C = 10$. For LSTM, the optimal hyperparameters were an LSTM cell internal state size of 100, and a fully connected layer size of 200.

5.3. Classifier Performance Comparison

Performance of classifiers according to the metrics we have chosen is summarized in Table 1. We also generated ROC and PRC curves for all classifiers in Figure 3 and Fig-

ure 4, respectively. From the table, we see that the training and testing errors for each classifier converged approximately to the same value, which indicates that we don’t have an overfitting problem. When we compare the AUROC values, LSTM results has the highest AUROC, although all of the classifiers have AUROC values higher than 0.9. On the other hand, RBF SVM outperforms the other classifiers in terms of recall value.

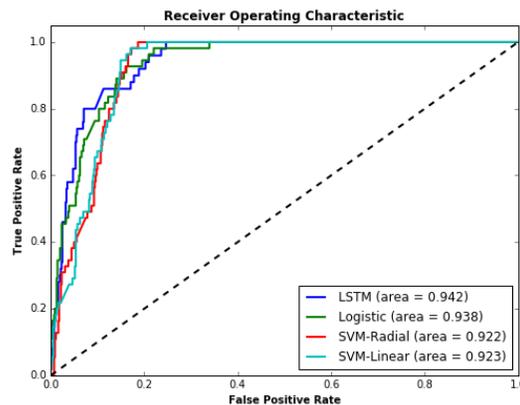


Figure 3: Receiver Operating Characteristics Curves of different classification methods.

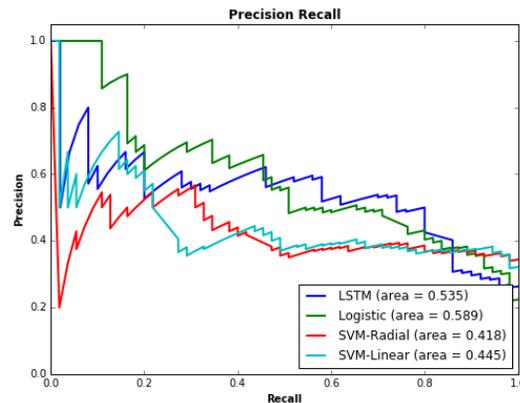


Figure 4: Precision-Recall Curves of different classification methods.

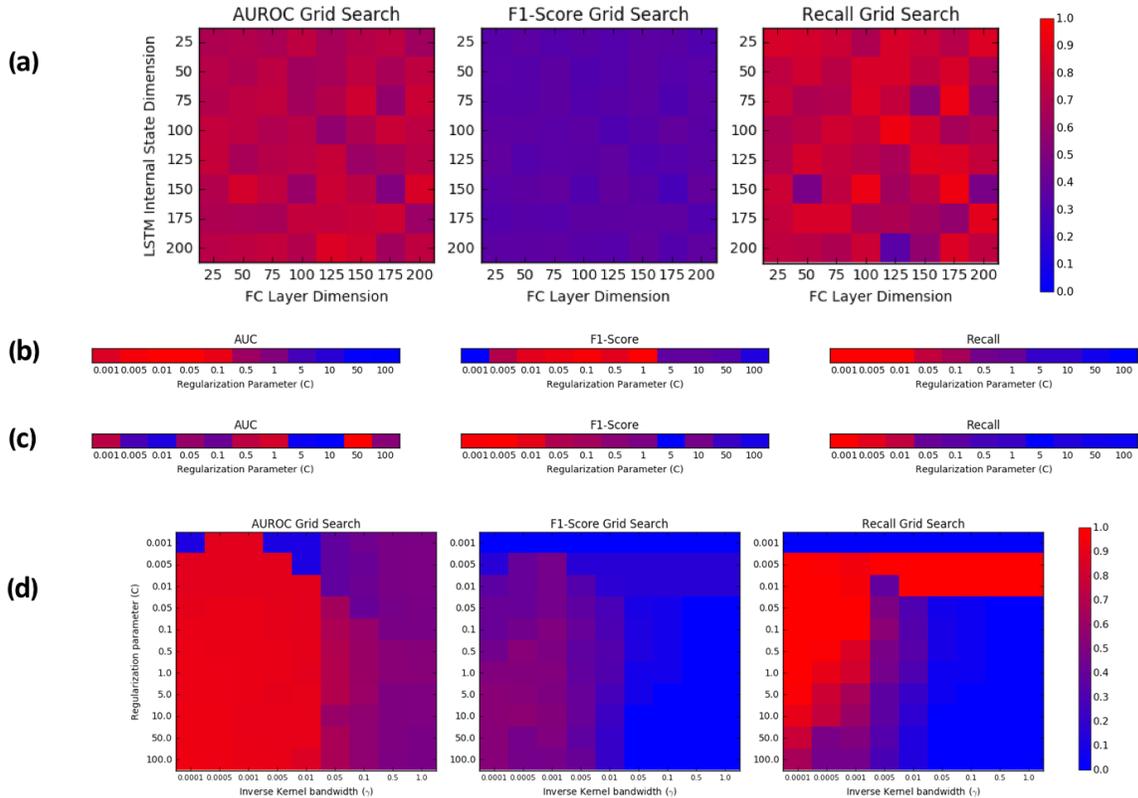


Figure 5: Hyperparameter Optimization: (a) Grid Search for LSTM, (b) Linear SVM, (c) Logistic Regression, (d) SVM with RBF kernel

6. Discussion

Our results demonstrate the feasibility of using spectral power across different bands to forecast a seizure event. For this application, it is vital to not miss a seizure, thus our emphasis on optimizing the recall rate. For most of our models, our recall was high but precision was low (mainly because of the imbalanced dataset), which explains the low F-scores and AUPRC values. All models had high AUROC, with the LSTM model having the highest.

The biggest challenge in this project was to identify the proper metrics that characterize the performance on the imbalanced dataset we have. Another challenge has been to identify the optimal hyperparameters using the multiple metrics we have selected. Although our results show that we achieved good performance, having a dataset that is more balanced would make performance comparison easier.

Unfortunately, we were unable to train a reasonable model when using a data set with multiple patients simultaneously, so such a model would only be customized for a single patient.

7. Future Directions

Although satisfactory results were attained, there are a myriad of ways in which one might increase the performance metrics of our classifiers. One method is to incorporate more features into the feature set, such as discrete wavelet transforms of the original signal, and several time-series statistics. However, too many features could potentially lower the generalization of our model across many subjects. One potential avenue of interesting exploration is the use of convolutional neural networks to first extract meaningful features from the spectrogram (Frequency vs Time) of an iEEG recording, as opposed to the manual feature extraction process mentioned earlier. Another direction would be to use autoencoders for nonlinear dimensionality reduction prior to performing supervised learning. Training an autoencoder on either the extracted feature set or even the raw signal may eliminate both noise and redundant features. Finally, it would be fruitful to also investigate domain adaptation techniques that would allow a model to be trained on one patient and still be applicable to another.

References

- [1] Florian Mormann, Ralph G. Andrzejak, Christian E. Elger, and Klaus Lehnertz. Seizure prediction: the long and winding road. *Brain*, 130(2):314–333, 2007.
- [2] Benjamin H. Brinkmann, Joost Wagenaar, Drew Abbot, Phillip Adkins, Simone C. Bosshard, Min Chen, Quang M. Tieng, Jialune He, F. J. Muñoz-Almaraz, Paloma Botella-Rocamora, Juan Pardo, Francisco Zamora-Martinez, Michael Hills, Wei Wu, Iryna Korshunova, Will Cukierski, Charles Vite, Edward E. Patterson, Brian Litt, and Gregory A. Worrell. Crowdsourcing reproducible seizure forecasting in human and canine epilepsy. *Brain*, 2016.
- [3] Kaggle.com. Melbourne University AES/MathWorks/NIH Seizure Prediction. <https://www.kaggle.com/c/melbourne-university-seizure-prediction>, 2016. [Online; accessed October-22-2016].
- [4] Kais Gadhomi, Jean-Marc Lina, Florian Mormann, and Jean Gotman. Seizure prediction for therapeutic devices: A review. *Journal of Neuroscience Methods*, 260:270 – 282, 2016. *Methods and Models in Epilepsy Research*.
- [5] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.