

A34
By 1 1

1 DEEP LEARNING APPROACHES FOR AUTOMATIC ANALYSIS OF EEGS

Meysam Golmohammadi^a, Vinit Shah^b, Iyad Obeid^b, and Joseph Picone^b

^a Internet Brands, El Segundo, California, USA

42 ^b The Neural Engineering Data Consortium, Temple University, Philadelphia, Pennsylvania, USA

Abstract

1 Scalp electroencephalograms (EEGs) are the primary means by which physicians diagnose brain-related illnesses such as epilepsy and seizures. Automated seizure detection using clinical EEGs is a very difficult machine learning problem due to the low fidelity of a scalp EEG signal. Nevertheless, despite the poor signal quality, clinicians can reliably diagnose illnesses from visual inspection of the signal waveform. Commercially available automated seizure detection systems, however, suffer from unacceptably high false alarm rates. Deep learning algorithms that require large amounts of training data have not previously been effective on this task due to the lack of big data resources necessary for building such models and the complexity of the signals involved. The evolution of big data science, most notably the release of the Temple University EEG (TUEG) Corpus, has motivated renewed interest in this problem.

132 In this chapter, we discuss the application of a variety of deep learning architectures to automated seizure detection. Architectures explored include multilayer perceptrons, convolutional neural networks (CNNs), long short-term memory networks (LSTMs), gated recurrent units and residual neural networks. We use the TUEG Corpus, supplemented with data from Duke University, to evaluate the performance of these hybrid deep structures. Since TUEG contains a significant amount of unlabeled data, we also discuss unsupervised pre-training methods used prior to training these complex recurrent networks.

1 Exploiting spatial and temporal context is critical for accurate disambiguation of seizures from artifacts. We explore how effectively several conventional architectures are able to model context and introduce a hybrid system that integrates CNNs and LSTMs. The primary error modalities observed by this state-of-the-art system were false alarms generated during brief delta range slowing patterns such as intermittent rhythmic delta activity. A variety of these types of events have been observed during inter-ictal and post-ictal stages. Training models on such events with diverse morphologies has the potential to significantly reduce the remaining false alarms. This is one reason we are continuing our efforts to annotate a larger portion of TUEG. Increasing the data set size significantly allows us to leverage more advanced machine learning methodologies.

52 **Keywords**— Deep Learning, Convolutional Neural Networks, CNN, Electroencephalography, EEG, Generative Adversarial Networks, GAN, Long Short-Term Networks, LSTM, Recurrent Networks

2 **Corresponding Author:** Joseph Picone, The Neural Engineering Data Consortium, ENGR 703A, Temple University, 1947 North 12th Street, Philadelphia, Pennsylvania, 19122, USA, Tel: 215-204-4841, Fax: 215-204-5960, Email: joseph.picone@gmail.com.

1. INTRODUCTION

An EEG records the electrical activity along the scalp and measures spontaneous electrical activity of the brain. The signals measured along the scalp can be correlated with brain activity, which makes it a primary tool for diagnosis of brain-related illnesses (Ilmoniemi, & Sarvas, J., 2019; Ebersole et al., 2014). Electroencephalograms (EEGs) are used in a broad range of health care institutions to monitor and record electrical activity in the brain. EEGs are essential in the diagnosis of clinical conditions such as epilepsy, depth of anesthesia, coma, encephalopathy, brain death and even in the progression of Alzheimer's disease (Yamada et al., 2017; Ercegovac & Berisavac, 2015).

Manual interpretation of EEGs is time-consuming since these recordings may last hours or days. It is also an expensive process as it requires highly trained experts. Therefore, high performance automated analysis of EEGs can reduce time of diagnosis and enhance real-time applications by flagging sections of the signal that need further review. Many methods have been developed over the years (Ney et al., 2016) including time-frequency digital signal processing techniques (Boashas, 2015; Gotman, 1999), autoregressive spectral analysis (Li et al., 2015), wavelet analysis (Li et al., 2016), nonlinear dynamical analysis (Rodriguez-Bermudez & Garcia-Laencina, 2015), multivariate techniques based on simulated leaky integrate-and-fire neurons (Eichler et al., 2017; Schadt et al., 2013; Schindler et al., 2001) and expert systems that attempt to mimic a human observer (Deburchgraeve et al., 2008). In spite of recent research progress in this field, the transition of automated EEG analysis technology to commercial products in operational use in clinical settings has been limited, mainly because of unacceptably high false alarm rates (Baumgartner & Koren, 2018; Haider et al., 2016; Varsavsky & Mareels, 2006).

In recent years, progress in machine learning and big data resources has enabled a new generation of technology that is approaching acceptable levels of performance for clinical applications. The main challenge in this task is to operate with an extremely low false alarm rate because even 5 false alarms per 24 hours per patient means healthcare staff will be overwhelmed servicing these events in a typical critical care unit with 12 to 47 beds, especially when one considers the amount of other equipment that frequently trigger alerts (Bridi et al., 2014). In this chapter, we discuss the application of deep learning technology to the automated EEG interpretation problem and introduce several promising architectures that deliver performance close to the requirements for operational use in clinical settings.

1.1. Leveraging Recent Advances in Deep Learning

Machine learning has made tremendous progress over the past three decades due to rapid advances in low-cost highly-parallel computational infrastructure, powerful machine learning algorithms, and, most importantly, big data. Although contemporary approaches for automatic interpretation of EEGs have employed modern machine learning approaches such as neural networks (Ahmed et al., 2019; Ramgopal et al., 2011) and support vector machines (Alotaiby et al., 2014), state-of-the-art machine learning algorithms have not previously been utilized in EEG analysis because of a lack of big data sources. A significant big data resource, known as the TUH EEG Corpus (TUEG) is now available creating a unique opportunity to evaluate high performance deep learning approaches (Obeid & Picone, 2016). This database includes detailed physician reports and patient medical histories, which are critical to the application of deep learning. However, transforming physicians' reports into meaningful information that can be exploited by deep learning paradigms is proving to be challenging because the mapping of reports to underlying EEG events is nontrivial.

Though modern deep learning algorithms have generated significant improvements in performance in fields such as speech and image recognition, it is far from trivial to apply these approaches to new domains, especially applications such as EEG analysis that rely on waveform interpretation. Deep learning approaches can be viewed as a broad family of neural network algorithms that use a large number of layers of nonlinear processing units to learn a mapping between inputs and outputs. These algorithms are usually trained using a combination of supervised and unsupervised learning. The best overall approach is often

determined empirically and requires extensive experimentation for optimization. There is no universal theory on how to arrive at the best architecture, and the results are almost always heavily data dependent. Therefore, in this chapter we will present a variety of approaches and establish some well-calibrated benchmarks of performance. We explore two general classes of deep neural networks in detail.

The first class is a Convolutional Neural Network (CNN), which is a class of deep neural networks that have revolutionized fields like image and video recognition, recommender systems, image classification, medical image analysis, and natural language processing through end to end learning from raw data (LeCun et al., 2015). An interesting characteristic of CNNs that was leveraged in these applications is their ability in learning local patterns in data by using convolutions, more precisely cross-correlation, as their key component. This property makes them a powerful candidate for modeling EEGs which are inherently multichannel signals. Each channel in an EEG possesses some spatial significance with respect to the type and locality of a seizure event (Shah et al., 2017). EEGs also have an extremely low signal to noise ratio and events of interest such as seizures are easily confused with signal artifacts (e.g., eye movements) or benign variants (e.g., slowing) (von Weltin et al., 2017). The spatial property of the signal is an important cue for disambiguating these types of artifacts from seizures. These properties make modeling EEGs more challenging compared to more conventional applications like image recognition of static images or speech recognition using a single microphone. In this study, we adapt well-known CNN architectures to be more suitable for automatic seizure detection. Leveraging a high-performance time-synchronous system that provides accurate segmentation of the signal is also crucial to the development of these kinds of systems. Hence, we use a hidden Markov model (HMM) based approach (Picone, 1990) as a non-deep learning baseline system (Golmohammadi et al., 2018).

Optimizing the depth of a CNN is crucial to achieving state-of-the-art performance. Best results are achieved on most tasks by exploiting very deep structures (e.g., thirteen layers are common) (Szegedy et al., 2015; Simonyan et al., 2014). However, training deeper CNN structures is more difficult since they are prone to degradation in performance with respect to generalization and suffer from convergence problems. Increasing the depth of a CNN incrementally often saturates sensitivity and also results in a rapid decrease in sensitivity. Often increasing the number of layers also increases the error on the training data due to convergence issues, indicating that the degradation in performance is not created by overfitting. We address such degradations in performance by designing deeper CNNs using a deep residual learning framework (ResNet) (He et al., 2016).

We also extend the CNN approach by introducing an alternate structure, a deep convolutional generative adversarial network (DCGAN) (Radford et al., 2015) to allow unsupervised training. Generative adversarial networks (GANs) (Goodfellow et al., 2014) have emerged as powerful techniques for learning generative models based on game theory. Generative models use an analysis by synthesis approach to learn the essential features of data required for high performance classification using an unsupervised approach. We introduce techniques to stabilize the training of DCGAN for spatio-temporal modeling of EEGs.

The second class of network that we discuss is a Long Short-Term Memory (LSTM) network. LSTMs are a special kind of recurrent neural network (RNN) architecture that can learn long-term dependencies. This is achieved by introducing a new structure called a memory cell and by adding multiplicative gate units that learn to open and close access to the constant error flow (Hochreiter et al., 1997). It has been shown that LSTMs are capable of learning to bridge minimal time lags in excess of 1,000 discrete time steps. To overcome the problem of learning long-term dependencies in modeling of EEGs, we describe a few hybrid systems composed of LSTMs that model both spatial relationships (e.g., cross-channel dependencies) and temporal dynamics (e.g., spikes). In an alternative approach for sequence learning of EEGs, we propose a structure based on gated recurrent units (GRUs) (Chung et al., 2014). A GRU is a gating mechanism for RNNs that is similar in concept to what LSTMs attempt to accomplish. Chung et al. (2014) demonstrated that GRUs can outperform many other RNNs, including LSTM, in several datasets.

1.2. Big Data Enables Deep Learning Research

Recognizing that deep learning algorithms require large amounts of data to train complex models, especially when one attempts to process clinical data with a significant number of artifacts using specialized models, we have developed a large corpus of EEG data to support this kind of technology development. The TUEG Corpus is the largest publicly available corpus of clinical EEG recordings in the world. The most recent release, v1.1.0, includes data from 2002 – 2015 and contains over 23,000 sessions from over 13,500 patients – over 1.8 years of multichannel signal data in total (Obeid & Picone, 2016). This dataset was collected at the Department of Neurology at Temple University Hospital. The data includes sessions taken from outpatient treatments, Intensive Care Units (ICU) and Epilepsy Monitoring Units (EMU), Emergency Rooms (ER) as well as several other locations within the hospital. Since TUEG consists entirely of clinical data, it contains many real-world artifacts (e.g., eye blinking, muscle artifacts, head movements). This makes it an extremely challenging task for machine learning systems and differentiates it from most research corpora currently available in this area. Each of the sessions contains at least one EDF file and one physician report. These reports are generated by a board-certified neurologist and are the official hospital record. These reports are comprised of unstructured text that describes the patient, relevant history, medications, and clinical impression. The corpus is publicly available from the Neural Engineering Data Consortium (www.nedcdata.org).

EEG signals in TUEG were recorded using several generations of Natus Medical Incorporated's Nicolet™ EEG recording technology (Natus, 2019). The raw signals consist of multichannel recordings in which the number of channels varies between 20 and 128 channels (Shah et al., 2017; Harati et al., 2014;). A 16-bit A/D converter was used to digitize the data. The sample frequency varies from 250 Hz to 1024 Hz. In our work, we resample all EEGs to a sample frequency of 250 Hz. The Natus system stores the data in a proprietary format that has been exported to EDF with the use of NicVue v5.71.4.2530. The original EEG records are split into multiple EDF files depending on how the session was annotated by the attending technician. For our studies, we use the 19 channels associated with a standard 10/20 EEG configuration and apply a Transverse Central Parasagittal (TCP) montage (Lopez et al., 2016; American Clinical Neurophysiology Society, 2012).

A portion of TUEG was annotated manually known as the TUH EEG Seizure Detection Corpus (TUSZ) (Shah et al., 2018). TUSZ is also the world's largest publicly available corpus of annotated data for seizure detection that is unencumbered. No data sharing or IRB agreements are needed to access the data. TUSZ contains a rich variety of seizure morphologies. Variation in onset and termination, frequency and amplitude, and locality and focality protect the corpus from a bias towards one type of seizure morphology. TUSZ, which reflects a seizure detection task, will be the focus of the experiments presented in this chapter. For related work on six-way classification of EEG events, see Golmohammadi et al. (2018), Shah et al. (2018) and Harati et al. (2015).

We have also included an evaluation on a held-out data set based on the Duke University Seizure Corpus (DUSZ) (Swisher et al., 2015). The DUSZ database is collected solely from the adult ICU patients exhibiting non-convulsive seizures. These are continuous EEG (cEEG) records (Kubota et al., 2018) where most seizures are very focal and slower in frequency. TUSZ in contrast contains records from a much broader range of patients and morphologies. A comparison of these two corpora is shown in Table 1. The evaluation sets are comparable in terms of the number of patients and total amount of data, but TUSZ contains many more sessions collected from each patient.

It is important to note that TUSZ was collected using several generations of Natus Incorporated EEG equipment (Natus, 2019), while DUSZ was collected at a different hospital, Duke University, using a Nihon Kohden system (Nihon Kohden, 2019). Hence, using DUSZ as a held-out evaluation set is an important benchmark because it tests the robustness of the models to variations in the recording conditions. Deep learning systems are notoriously prone to overtraining, so this second data set represents important evidence that the results presented here are generalizable and reproducible on other tasks.

2. TEMPORAL MODELING OF SEQUENTIAL SIGNALS

The classic approach to machine learning, shown in Figure 1, involves an iterative process that begins with the collection and annotation of data and ends with an open set, or blind, evaluation. Data is usually sorted into training, development test set and evaluation. Evaluations on the development set, or dev_test data as it is often referred to, is used to guide system development. One cannot adjust system parameters based on the outcome of the open set evaluations but can use these results to assess overall system performance. We typically iterate on all aspects of this approach, including expansion and repartitioning of the training and dev_test data, until overall system performance is optimized.

We often leverage previous stages of technology development to seed, or initialize, models used in a new round of development. Further, there is often a need to temporally segment the data, for example automatically labeling events of interest, to support further explorations of the problem space. Therefore, it is common when exploring new applications to begin with a familiar technology. As previously mentioned, EEG signals have a strong temporal component. Hence, a likely candidate for establishing good baseline results is an HMM approach, since this algorithm is particularly strong at automatically segmenting the data and localizing events of interest.

HMM systems typically operate on a sequence of vectors referred to as features. In this section, we briefly introduce the feature extraction process we have used, and we describe a baseline system that integrates hidden Markov models for sequential decoding of EEG events with deep learning for decision-making based on temporal and spatial context.

2.1. A Linear Frequency Cepstral Coefficient Approach to Feature Extraction

The first step in our machine learning systems consists of converting the signal to a sequence of feature vectors (Picone, 1993). Common EEG feature extraction methods include temporal, spatial and spectral analysis (Thodoroff et al., 2016; Mirowski et al., 2009). A variety of methodologies have been broadly applied for extracting features from EEG signals including a wavelet transform, independent component analysis and autoregressive modeling (Subasi et al., 2007; Jahankhani et al., 2006). In this study, we use a methodology based on mel-frequency cepstral coefficients (MFCC) which have been successfully applied to many signal processing applications including speech recognition (Picone, 1993). In our systems, we use linear frequency cepstral coefficients (LFCCs) since a linear frequency scale provides some slight advantages over the mel scale for EEG signals (Harati et al., 2015). A block diagram summarizing the feature extraction process used in this work is presented in Figure 2. Though it is increasingly popular to operate directly from sampled data in a deep learning system, and let the system learn the best set of features automatically, for applications in which there is limited annotated data, it is often more beneficial to begin with a specific feature extraction algorithm. Experiments with different types of features (Da Rocha Garrit et al., 2015) or with using sampled data directly (Xiong et al., 2017) have not shown a significant improvement in performance.

Harati et al. (2015) did an extensive exploration of many of the common parameters associated with feature extraction and optimized the process for six-way event classification. We have found this approach, which leverages a popular technique in speech recognition, is remarkably robust across many types of machine learning applications. The LFCCs are computed by dividing raw EEG signals into shorter frames using a standard overlapping window approach. A high resolution discrete fast Fourier Transform is computed next. The spectrum is downsampled with a filter bank composed of an array of overlapping bandpass filters. Finally, the cepstral coefficients are derived by computing a discrete cosine transform of the filter bank's output (Picone, 1993). In our experiments, we discarded the zeroth-order cepstral coefficient, and replaced it with a frequency domain energy term which is calculated by adding the output of the oversampled filter bank after they are downsampled:

$$E_f = \log(\sum_{k=0}^{N-1} |X(k)|^2) . \quad (1)$$

22

We also introduce a new feature, called differential energy, that is based on the long-term differentiation of energy. Differential energy can significantly improve the results of spike detection, which is a critical part of seizure detection, because it amplifies the differences between transient pulse shape patterns and stationary background noise. To compute the differential energy term, we compute the energy of a set of consecutive frames, which we refer to as a window, for each channel of an EEG:

$$E_d = \max_m (E_f(m)) - \min_m (E_f(m)). \quad (2)$$

We have used a window of 9 frames which are each 0.1 secs in duration, corresponding to a total duration of 0.9 secs. We calculate differential energy term. Even though this term is a relatively simple feature, it resulted in a statistically significant improvement in spike detection performance (Harati et al., 2015).

Our experiments have also shown that using regression-based derivatives of features, which is a popular method in speech recognition (Picone, 1993), is effective in the classification of EEG events. We use the following definition for the derivative:

$$d_t = \frac{\sum_{n=1}^N n(c_{t+n} - c_{t-n})}{2 \sum_{n=1}^N n^2}. \quad (3)$$

Eq. (3) is applied to the cepstral coefficients, c_t , to compute the first derivatives, which are referred to as delta coefficients. Eq. (3) is then reapplied to the first derivatives to compute the second derivatives, which are referred to as delta-delta coefficients. Again, we use a window length of 9 frames (0.9 secs) for the first derivative and a window length of 3 (0.3 secs) for the second derivative. The introduction of derivatives helps the system discriminate between steady-state behavior, such as that found in a PLED event, and impulsive or nonstationary signals, such as that found in spikes (SPSW) and eye movements (EYEM).

Through experiments designed to optimize feature extraction, we found best performance can be achieved using a feature vector length of 26. This vector includes nine absolute features consisting of seven cepstral coefficients, one frequency-domain energy term, and one differential energy term. Nine deltas are added for these nine absolute features. Eight delta-deltas are added because we exclude the delta-delta term for differential energy (Harati et al., 2015).

2.2. Temporal and Spatial Context Modeling Using Hybrid HMMs and Deep Learning

HMMs are among the most powerful statistical modeling tools available today for signals that have both a time and frequency domain component (Picone, 1990). HMMs have been used quite successfully in sequential decoding tasks like speech recognition (Huang et al., 2002), cough detection (Parker et al., 2013) and gesture recognition (Lu et al., 2012) to model signals that have sequential properties such as temporal or spatial evolution. Automated interpretation of EEGs is a problem like speech recognition since both time domain (e.g., spikes) and frequency domain information (e.g., alpha waves) are used to identify critical events (Obeid and Picone, 2018). EEGs also have a spatial component as well.

A left-to-right channel-independent GMM-HMM, as illustrated in Figure 3, was used as a baseline system for sequential decoding (Golmohammadi et al., 2018). HMMs are attractive because training is much faster than comparable deep learning systems, and HMMs tend to work well when moderate amounts of annotated data are available. We divide each channel of an EEG into 1 sec epochs, and further subdivide these epochs into a sequence of 0.1 sec frames. Each epoch is classified using an HMM trained on the subdivided epoch, and then these epoch-based decisions are postprocessed by additional statistical models in a process that parallels the language modeling component of a speech recognizer. Standard three state left-to-right HMMs (Picone, 1990) with 8 Gaussian mixture components per state were used. The covariance matrix for each mixture component was assumed to be a diagonal matrix – a common assumption for cepstral-based features. Though we evaluated both channel-dependent and channel-independent models, channel-independent models were ultimately used because channel-dependent models did not provide any improvement in performance.

Supervised training based on the Baum-Welch reestimation algorithm was used to train two models – seizure and background. Models were trained on segments of data containing seizures based on manual annotations. Since seizures comprise a small percentage of the overall data (3% in the training set; 8% in the evaluation set), the amount of non-seizure data was limited to be comparable to the amount of seizure data, and non-seizure data was selected to include a rich variety of artifacts such as muscle and eye movements. Twenty iterations of Baum-Welch were used though performance is not very sensitive to this value. Standard Viterbi decoding (no beam search) was used in recognition to estimate the model likelihoods for every epoch of data. The entire file was not decoded as one stream because of the imbalance between the seizure and background classes – decoding was restarted for each epoch.

The output of the epoch-based decisions was postprocessed by a deep learning system. Our baseline system used a Stacked denoising Autoencoder (SdA) (Vincent et al., 2010; Vincent et al., 2008) as shown in Figure 3. SdAs are an extension of stacked autoencoders and are a class of deep learning algorithms well-suited to learning knowledge representations that are organized hierarchically (Bengio et al., 2007). They also lend themselves to problems involving training data that is sparse, ambiguous or incomplete. Since inter-rater agreement is relatively low for seizure detection (Haider et al., 2016), it made sense to evaluate this type of algorithm as part of a baseline approach.

An N-channel EEG was transformed into N independent feature streams. The hypotheses generated by the HMMs were postprocessed using a second stage of processing that examines temporal and spatial context. We apply a third pass of postprocessing that uses a stochastic language model to smooth hypotheses involving sequences of events so that we can suppress spurious outputs. This third stage of postprocessing provides a moderate reduction in false alarms.

Training of SdA networks are done in two steps: (1) pre-training in a greedy layer-wise approach (Bengio et al., 2007) and (2) fine-tuning by adding a logistic regression layer on top of the network (Hinton et al., 2006). The output of the first stage of processing is a vector of two likelihoods for each channel at each epoch. Therefore, if we have 22 channels and 2 classes (seizure and background), we will have a vector of dimension $2 \times 22 = 44$ for each epoch.

Each of these scores is independent of the spatial context (other EEG channels) or temporal context (past or future epochs). To incorporate context, we form a supervector consisting of N epochs in time using a sliding window approach. We benefit to making N large – typically 41. This results in a vector of dimension $41 \times 44 = 1,804$ that needs to be processed each epoch. The input dimensionality is too high considering the amount of manually labeled data available for training and the computational requirements. To deal with this problem we used Principal Components Analysis (PCA) (van der Maaten et al., 2009; Ross et al., 2008) to reduce the dimensionality to 20 before applying the SdA postprocessing.

The parameters of the SdA model are optimized to minimize the average reconstruction error using a cross-entropy loss function. In the optimization process, a variant of stochastic gradient descent is used called “Minibatch stochastic gradient descent” (MSGD) (Zinkevich et al., 2010). MSGD works identically to stochastic gradient descent, except that we use more than one training example to make each estimate of the gradient. This technique reduces variance in the estimate of the gradient, and often makes better use of the hierarchical memory organization in modern computers.

The SdA network has three hidden layers with corruption levels of 0.3 for each layer. The number of nodes in each layer are: 1st layer (connected to the input layer) = 800, 2nd layer = 500, 3rd layer (connected to the output layer) = 300. The parameters for pre-training are: learning rate = 0.5, number of epochs = 150, batch size = 300. The parameters for fine-tuning are: learning rate = 0.1, number of epochs = 300, batch size = 100. The overall result of the second stage is a probability vector of dimension two containing a likelihood that each label could have occurred in the epoch. A soft decision paradigm is used rather than a hard decision paradigm because this output is smoothed in the third stage of processing. A more detailed explanation about the third pass of processing is presented in (Harati et al., 2016).

3. IMPROVED SPATIAL MODELING USING CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks (CNNs) have delivered state of the art performance on highly challenging tasks such as speech (Saon et al., 2016) and image recognition (Simonyan et al., 2014), and these early successes played a vital role in stimulating interest in deep learning approaches. In this section we explore modeling of spatial information in the multichannel EEG signal to exploit our knowledge that seizures occur on a subset of channels (Ebersole, 2014). The identity of these channels also plays an important role localizing the seizure and identifying the type of seizure (Lopez et al., 2017).

3.1. Deep Two-Dimensional Convolutional Neural Networks

CNN networks are usually composed of convolutional layers, subsampling layers followed by one or more fully connected layers. Consider an image of dimension $W \times H \times N$, where W and H are the width and height of the image in pixels, and N is the number of channels (e.g. in an RGB image, $N = 3$ since there are three colors). Two-dimensional (2D) CNNs are only used in sequential decoding problems such as speech or image recognition typically consist of a convolutional layer that will have K filters (or kernels) of size $M \times N \times Q$ where M and N are smaller than the dimension of the data and Q is smaller than the number of channels. The image can be subsampled by skipping samples as you convolve the kernel over the image. This is known as the stride, which is essentially a decimation factor. CNNs have a large learning capacity that can be controlled by varying their depth and breadth to produce K feature maps of size $(W-M+1) \times (H-N+1)$ for a stride of 1, and proportionally smaller for larger strides. Each map is then subsampled using a technique known as max pooling (LeCun et al., 2015), in which a filter is applied to reduce the dimensionality of the map. An activation function, such as a rectified linear unit (ReLU), is applied to each feature map either before or after the subsampling layer to introduce nonlinear properties to the network. Nonlinear activation functions are necessary for learning complex functional mappings.

In Figure 14, a system that combines CNN and a multi-layer perceptron (MLP) (Simonyan et al., 2014) is shown. Drawing on our image classification analogy, each image is a signal where the width of the image (W) is the window length multiplied by the number of samples per second, the height of the image (H) is the number of EEG channels and the number of image channels (N) is the length of the feature vector. This architecture includes six convolutional layers, three max pooling layers and two fully-connected layers. A rectified linear unit (ReLU) nonlinearity is applied to the output of every convolutional and fully-connected layer (Nair et al., 2010).

In our optimized version of this architecture, a window duration of 7 secs is used. The first convolutional layer filters the input of size $70 \times 22 \times 26$ using 16 kernels of size 3×3 with a stride of 1. The input feature vectors have a dimension of 26, while there are 22 EEG channels. The window length is 70 because the features are computed every 0.1 secs, or 10 times per second, and the window duration is 7 sec. These kernel sizes and strides were experimentally optimized (Golmohammadi et. al., 2018).

The second convolutional layer filters its input using 16 kernels of size 3×3 with a stride of 1. The first max pooling layer takes as input the output of the second convolutional layer and applies a pooling size of 2×2 . This process is repeated two times with kernels of size 32 and 64. Next, a fully-connected layer with 512 neurons is applied and the output is fed to a 2-way sigmoid function which produces a two-class decision. This two-class decision is the final label for the given epoch, which is 1 sec in duration. Neurologists usually review EEGs using 10 sec windows, so we attempt to use a similar amount of context in this system. Pattern recognition systems often subdivide the signal into small segments during which the signal can be considered quasi-stationary. A simple set of preliminary experiments determined that a reasonable tradeoff between computational complexity and performance was to split a 10 sec window, which is what neurologists use to view the data, into 1 sec epochs (Farati et al., 2015).

In our experiments, we found structures that are composed of two consecutive convolutional layers before a pooling layer perform better than structures with one convolutional layer before a pooling layer. Pooling

layers decrease the dimensions of the data and thereby can result in a loss of information. Using two convolutional layers before pooling mitigates the loss of information. We find that using a very small fields throughout the architecture (e.g., 3×3) performs better than larger fields (e.g. 5×5 or 7×7) in the first convolutional layer.

3.2. Augmenting CNNs with Deep Residual Learning

The depth of a CNN plays an instrumental role in its ability to achieve high performance (Simonyan et al., 2014; Szegedy et al., 2015). As many as thirteen layers are used for challenging problems such as speech and image recognition. However, training deeper CNN structures is more difficult since convergence and generalization become issues. Increasing the depth of CNNs, in our experience, tends to increase the error on evaluation dataset. As we add more convolutional layers, sensitivity first saturates and then degrades quickly. We also see an increase in the error on the training data when increasing the depth of a CNN, indicating that overfitting is actually not occurring. Such degradations in performance can be addressed by using a deep residual learning framework known as a ResNet (He et al., 2016). ResNets introduce an “identity shortcut connection” that skips layers. Denoting the desired underlying mapping as $H(x)$, we map the stacked nonlinear layers using $F(x) = H(x) - x$, where x is the input. The original mapping is recast into $F(x) + x$. It can be shown that it is easier to optimize the residual mapping than to optimize the original, unreferenced mapping (He et al., 2016).

The deep residual learning structure mitigates two important problems: vanishing/exploding gradients and saturation of accuracy when the number of layers is increased. As the gradient is backpropagated to earlier layers, repeated multiplication of numbers less than one often makes the gradient infinitively small. Performance saturates and can rapidly degrade due to numerical precision issues. Our structure addresses these problems by reformulating the layers as learning residual functions with reference to the layer inputs instead of learning unreferenced functions.

An architecture for our ResNet approach is illustrated in Figure 5. The shortcut connections between the convolutional layers make training of the model tractable by allowing information to propagate effectively through this very structure. The network consists of 6 residual blocks with two 2D convolutional layers per block. These convolutional layers are followed by a fully connected layer and a single dense neuron as the last layer. This brings the total number of layers in this modified CNN structure to 130. The 2D convolutional layers all have a filter length of (3, 3). The first 7 layers of this architecture have 32 filters while the last 128s have 64 filters. We increment the number of filters from 32 to 64, since the initial layers represent generic features, while the deeper layers represent more detailed features. In other words, the richness of the data representations increases because each additional layer forms new kernels using combinations of the features from the previous layer.

Except for the first and last layers of the network, before each convolutional layer we apply a Rectified Linear Unit (ReLU) as an activation function (Maas et al., 2013). ReLU is the most commonly used activation function in deep learning models. This function returns 0 if it receives any negative input, but for any positive value it returns that value (e.g., $f(x) = \max(0, x)$). To overcome the problem of overfitting in deep learning structures with a large number of parameters, we use dropout (Srivastava et al., 2010) as our regularization method between the convolutional layers and after ReLU. Dropout is a regularization technique for addressing overfitting by randomly dropping units along with their connections from the deep learning structures during training. We use the Adam optimizer (Kingma et al., 2015) which is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. After parameter tuning, we apply Adam optimization (Kingma et al., 2015) using the following parameters (according to the notation in their original paper): $\alpha = 0.00005$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and $decay = 0.0001$.

The deep learning systems described thus far have incorporated fully supervised training and discriminative models. Next, we introduce a generative deep learning structure based on convolutional neural networks (74) leverages unsupervised learning techniques. These are important for biomedical applications where large amounts of fully annotated data are difficult to find.

3.3. Unsupervised Learning Using Deep Convolutional Generative Adversarial Networks

Machine learning algorithms can generally be split into two categories: generative and discriminative. A generative model learns the joint probability distribution of $P(X, Y)$ where X is an observable variable and Y is the target variable. These models learn the statistical distributions of the input data rather than simply classifying the data as one of C output classes. Hence the name, generative, since these methods learn to replicate the underlying statistics of the data. GMMs trained using a greedy clustering algorithm or HMMs trained using the Expectation Maximization (EM) algorithm (Jelinek, 1997) are well-known examples of generative models. A discriminative model, on the other hand, learns the conditional probability of the target Y , given an observation X which we denote $P(Y|X)$ (Bishop, 2011). Support Vector Machines (Cortes & Vapnik, 1995) and Maximum Mutual Information Estimation (MMIE) (Bahl et al., 1986) are two well-known discriminative models.

Generative adversarial networks (GANs) (Goodfellow et al., 2014) have emerged as a powerful learning paradigm technique for learning generative models for high-dimensional unstructured data. GANs use a game theory approach to find the Nash equilibrium between a generator and discriminator network (Pandey, 2018). A basic GAN structure consists of two neural networks: a generative model G that captures the data distribution, and a discriminative model D estimates the probability that a sample came from the training data rather than G . These two networks are trained simultaneously via an adversarial process. In this process, the generative network, G , transforms the input noise vector z to generate synthetic data $G(z)$. The training objective for G is to maximize the probability of D making a mistake about the source of the data.

The output of the generator is a synthetic EEG – data that is statistically consistent with a real EEG but is fabricated entirely by the network. The second network, which is the discriminator, D , takes as input either the output of G or samples from real world data. The output of D is a probability distribution over possible input sources. The output of the discriminator in GAN determines if the signal is a sample from real world data or synthetic data from the generator.

The generative model, G , and the discriminative model, D , compete in a two-player minimax game with a value function, $V(G; D)$, in a way that D is trained to maximize the probability of assigning the correct label to both the synthetic and real data from G (Goodfellow et al., 2014). The generative model G is trained to fool the discriminator by minimizing $\log(1 - D(G(z)))$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] . \quad (4)$$

During the training process, our goal is to find a Nash equilibrium of a non-convex two-player game that minimizes both the generator and discriminator's cost functions (Pandey, 2018).

A deep convolutional generative adversarial network (DCGAN) is shown in Figure 6. The generative model takes 100 random inputs and maps them to a matrix with size of $[21, 22, 250]$, where 21 is the window length (corresponding to a 21 sec duration), 22 is number of EEG channels and 250 is number of samples per sec. Recall, in our study, we resample all EEGs to a sample frequency of 250 Hz (Harati et al., 2015). The generator is composed of transposed CNNs with upsamplers. Transposed convolution, also known as fractionally-strided convolution, can be implemented by swapping the forward and backward passes of a regular convolution (Goodfellow et al., 2014). We need transposed convolutions in the generators since we want to go in the opposite direction of a normal convolution. For example, in this case we want to compose

the vector of [21, 22, 250] from 100 random inputs. Using transposed convolutional layers, we can transform feature maps to a higher-dimensional space. Leaky ReLUs (Maas et al., 2013) are used for the activation function and dropout layers are used for regularization. Adam is used as the optimizer and binary cross-entropy (Goodfellow et al., 2016) is used as the loss function.

In this architecture, the discriminative model accepts vectors from two sources: synthetic data generators and real data (raw EEGs in this case). It is composed of strided convolutional neural networks (Goodfellow et al., 2016). Strided convolutional neural networks are like regular CNNs but with a stride greater than one. In the discriminator we replace the usual approach of convolutional layers with max pooling layers with strided convolutional neural networks. This is based on our observations in pilot experiments that using convolutional layers with max pooling makes the training of DCGAN unstable. This is due to the fact that using strided convolutional layers, the network learns its own spatial downsampling, and convolutional layers with max pooling tend to conflict with striding.

Finding the Nash equilibrium, which is a key part of the GAN approach, is a challenging problem that impacts convergence during training. Several recent studies address the instability of GANs and suggest techniques to increase the training stability of GANs (Salimans et al., 2016; Radford et al., 2015). We conducted a number of preliminary experiments and determined that these techniques were appropriate:

In the discriminator:

- pretraining of the discriminator;
- one-sided label smoothing;
- eliminating fully connected layers on top of convolutional features;
- replacing deterministic spatial pooling functions (such as max pooling) with strided convolutions.

In the generator:

- using an ReLU activation for all layers except for the output;
- normalizing the input to [-1, 1] for the discriminator;
- using a $\tanh()$ activation in the last layer except for the output;
- using leaky ReLU activations in the discriminator for all layers except for the output;
- freezing the weights of discriminator during adversarial training process;
- unfreezing weights during discriminative training;
- eliminating batch normalization in all the layers of both the generator and discriminator.

The GAN approach is attractive for a number of reasons including creating an opportunity for data augmentation. Data augmentation is common in many state-of-the-art deep learning systems today (Yang et al., 2016), allowing the size of the training set to be increased as well as exposing the system to previously unseen patterns during training.

4. LEARNING TEMPORAL DEPENDENCIES USING RECURRENT NETWORKS

The duration of events such as seizures can vary dramatically from a few seconds to minutes. Further, neurologists use significant amounts of temporal context and adaptation in manually interpreting EEGs. They are very familiar with their patients and often can identify the patient by examining the EEG signal, especially when there are certain types of anomalous behaviors. In fact, they routinely use the first minute or so of an EEG to establish baseline signal conditions (Lopez, 2017), or normalize their expectations, so that they can more accurately determine anomalous behavior. Recurrent neural networks (RNN), which date back to the late 1980's (Lang et al., 1989), have been proposed as a way to learn such dependencies. Prior to this, successful systems were often based on approaches such as hidden Markov models, or used heuristics to convert frame-level output into longer-term hypotheses. In this section, we introduce several architectures that model long-term dependencies.

4.1. Integration of Incremental Principal Component Analysis with LSTMs

In the HMM/SdA structure proposed in Section 2.2, PCA was used prior to SdA for dimensionality reduction. Unlike HMM/SdA, applying LSTM networks directly to features requires more memory efficient approaches than PCA, or the memory requirements of the network can easily exceed the available computational resources (e.g., low-cost graphics processing units such as the Nvidia 1080ti have limited amount of memory – typically 8Gbytes). Incremental principal components analysis (IPCA) is an effective technique for dimensionality reduction (Ross et al., 2008; Levy et al., 2000). This algorithm is often more memory efficient than PCA. IPCA has constant memory complexity proportional to the batch size, and it enables use of large datasets without a need to load the entire file or dataset into memory. IPCA builds a low-rank approximation for the input data using an amount of memory which is independent of the number of input data samples. It is still dependent on the dimensionality of the input data features but allows more direct control of memory usage by changing the batch size.

In PCA, the first k dominant principal components, $y_1(n), y_2(n), \dots, y_k(n)$, are computed directly from the input, $x(n)$ as follows:

For $n = 1, 2, \dots$, do the following:

1) $x_1(n) = x(n)$.

2) For $i = 1, 2, \dots, \min(k, n)$, do:

a) if $i = n$, initialize the i principal component as $y_i(n) = x_i(n)$;

b) otherwise compute:

$$y_i(n) = \frac{n-1-p}{n} y_i(n-1) + \frac{1+p}{n} x_i(n) x_i^T(n) \frac{y_i(n-1)}{\|y_i(n-1)\|}, \quad (5)$$

$$x_{i+1}(n) = x_i(n) - x_i^T(n) \frac{y_i(n)}{\|y_i(n)\|} \frac{y_i(n)}{\|y_i(n)\|}, \quad (6)$$

140

where the positive parameter p is called the amnesic parameter. Typically, p ranges from 2 to 4. Then eigenvector and eigenvalues are given by:

$$e_i = \frac{y_i(n)}{\|y_i(n)\|} \text{ and } \lambda_i = \|y_i(n)\|. \quad (7)$$

In Figure 7, we present an architecture that integrates IPCA and LSTM (Golmohammadi et al., 2018). In this system, samples are converted to features and the features are delivered to an IPCA layer that performs spatial context analysis and dimensionality reduction. The output of the IPCA layer is delivered to a one-layer LSTM for seizure classification task. The input to the IPCA layer is a vector whose dimension is the product of the number of channels, the number of features per frame and the number of frames of context. Preliminary experiments have shown that 7 seconds of temporal context performs well. The corresponding dimension of the vector input to IPCA is 22 channels \times 26 features \times 7 seconds \times 10 frames/second, or a total of 4004 elements. A batch size of 50 is used in IPCA and the dimension of its output is 25 elements per frame at 10 frames/second. In order to learn long-term dependencies, one LSTM with a hidden layer size of 128 and batch size of 128 is used along with Adam optimization and a cross-entropy loss function.

4.2. End-to-End Sequence Labeling Using Deep Architectures

In machine learning, sequence labeling is defined as assigning a categorical label to each member of a sequence of observed values. In automatic seizure detection, we assign one of two labels: seizure or non-seizure. This decision is made every epoch, which is typically a 1 sec interval. The proposed structures are trained in an end-to-end fashion, requiring no pre-training and no pre-processing, beyond the feature extraction process that was explained in Section 2.1. For example, for an architecture composed of a combination of CNN and LSTM, we do not train CNN independently from LSTM, but we train both jointly. This is challenging because there are typically convergence issues when attempting this.

In Figure 8, we integrate 2D CNNs, 1-D CNNs and LSTM networks, which we refer to as a CNN/LSTM, to better exploit long-term dependencies (Golmohammadi et al., 2018). Note that the way that we handle data in CNN/LSTM is different from the CNN/MLP system presented in Figure 4. The input EEG features vector sequence can be thought of as being composed of frames distributed in time where each frame is an image of width (W) equal to the length of a feature vector. The height (H) equals the number of EEG channels and the number of image channels (N) equals one. The input to the network consists of T frames where T is equal to the window length multiplied by the number of frames per second. In our optimized system, where features are available 10 times per second, a window duration of 21 seconds is used. The first 2D convolutional layer filters 210 frames ($T = 21 \times 10$) of EEGs distributed in time with a size of $26 \times 22 \times 1$ ($W = 26$, $H = 22$, $N = 1$) using 16 kernels of size 3×3 with a stride of 1. The first 2D max pooling layer takes as input a vector which is 260 frames distributed in time with a size of $26 \times 22 \times 16$ and applies a pooling size of 2×2 . This process is repeated two times with two 2D convolutional layers with 32 and 64 kernels of size 3×3 respectively and two 2D max pooling layers with a pooling size 2×2 .

The output of the third max pooling layer is flattened to 210 frames with a size of 384×1 . Then a 1D convolutional layer filters the output of the flattening layer using 16 kernels of size 3 which decreases the dimensionality in space to 210×16 . Next, we apply a 1D max pooling layer with a size of 8 to decrease the dimensionality to 26×16 . This is the input to a deep bidirectional LSTM network where the dimensionality of the output space is 128 and 256. The output of the last bidirectional LSTM layer is fed to a 2-way sigmoid function which produces a final classification of an epoch. To overcome the problem of overfitting and force the system to learn more robust features, dropout and Gaussian noise layers are used between layers (Srivastava et al., 2010). To increase nonlinearity, Exponential Linear Units (ELU) are used (Clevert et al., 2017). Adam is used in the optimization process along with a mean squared error loss function.

More recently, Cho et al. (2014) proposed another type of recurrent neural network, known as a gated recurrent unit (GRU). A GRU architecture is similar to an LSTM but without a separate memory cell. Unlike LSTM, a GRU does not include output activation functions and peep hole connections. It also integrates the input and forget gates into an update gate to balance between the previous activation and the candidate activation. The reset gate allows it to forget the previous state (Wu et al., 2016). It has been shown that the performance of a GRU is on par with an LSTM, but a GRU can be trained faster (Golmohammadi et al., 2017). The architecture is similar to that in Figure 8, but we simply replace LSTM with GRU, in a way that the output of 1D max pooling is the input to a GRU where the dimensionality of the output space is 128 and 256. The output of the last GRU is fed to a 2-way sigmoid function which produces final classification of an epoch. These two approaches, LSTM and GRU, are evaluated as part of a hybrid architecture that integrates CNNs with RNNs (Golmohammadi et al., 2017).

4.3. Temporal Event Modeling Using LSTMs

Final architecture we wish to consider is a relatively straightforward variation of an LSTM network. LSTMs are a special type of recurrent neural network which contains forget and output gates to control the information flow during its recurrent passes. LSTM networks have proven to outperform conventional HMMs, HMMs and other sequence learning methods in numerous applications such as speech recognition and handwriting recognition (Hermans et al., 2013; Graves et al., 2013). Our first implementation of LSTM was a hybrid network of both HMM and LSTM networks. A block diagram of HMM/LSTM system is shown in Figure 9. Similar to the HMM/SdA model discussed before, the input to the second layer of the system, which is the first layer of LSTMs, is a vector of dimension $2 \times 22 \times \text{window length}$. We use PCA to reduce the dimensionality of the input vector to 20 and pass it to the LSTM model. A window size of 41 secs (41 epochs at 1 sec per epoch) is used for a 32-node single hidden layer LSTM network. The final layer uses a dense neuron with a sigmoid activation function. The parameters of the models are optimized to minimize the error using a cross-entropy loss function and Adam (Kingma et al., 2015).

Next, we use a 3-layer LSTM network model. Identification of a seizure event is done based on an observation of a specific type of epileptiform activity called “spike and wave discharges” (Krauss et al., 2011). The evolution of these activities across time helps identify a seizure event. These events can be observed on individual channels. Once observed, seizures can be confirmed based on their focality, signal energy and its polarity across spatially close channels. The architecture is shown in Figure 10.

In the preprocessing step, we extract a 26-dimensional feature vector for an 11-frame context centered around the current frame. The output dimensionality for each frame is 10×26 (left) + 26 (center) + 10×26 (right) = 40. The static LSTM cells are used with a fixed batch size of 64 and a window size of 7 seconds. The data is randomly split into subsets where 80% is used for training and 20% is used for cross-validation during optimization. The features are normalized and scaled down to a range of [0, 1] on a file basis, which helps the gradient descent algorithm (and its variants) to converge much faster (Ioffe & Szegedy, 2015). Shuffling was performed on batches to avoid training biases.

The network includes 3 LSTM layers with (256, 64, 16) hidden layers followed by a 2-cell dense layer. The activation function used for all LSTM layers is a hyperbolic tangent function, $\tanh()$, except for the final layer, which uses a softmax function to compress the range of output value [0,1] so they resemble posterior probabilities. A cross-entropy function is used for calculating loss. Stochastic gradient descent with Nesterov momentum is used for optimization. Nesterov momentum attempts to increase the speed of training by introducing a momentum term based on accumulated gradients of its previous steps and a correction term in the direction of the current gradient (Sutskever et al., 2013). This tends to reduce the amount of overshoot during optimization.

The optimization is performed on the training data at a very high learning rate of 1.0 for the first five epochs. Cross-validation is performed after each epoch. After five epochs, if the cross-validation loss stagnates for three consecutive epochs (referred to as “patience = 3”), learning rates are halved after each iteration until it anneals to zero. If the model fails to show consistent performance on a cross-validation set, then it reverts to the previous epoch’s weights and restarts training until convergence. This method helps models avoid overfitting on the training data as long as the training and cross-validation sets are equally diverse.

The outputs of the models are fed to a postprocessor which is described in more detail in Section 5.2. This postprocessor is designed based on domain knowledge and observed system behavior to remove spurious and misleading detections. This is implemented to incorporate spatial context. The postprocessor sets a threshold for hypothesis confidence, the minimum number of channels for target event detection and a duration constraint which must be satisfied for detection. For example, if multiple channels consistently detected spike and wave discharges in the same 9-second interval, this event would be permitted as a valid output. Outputs from a fewer number of channels or over a smaller duration of time would be suppressed.

We have now presented a considerable variety of deep learning architectures. It is difficult to predict which architecture performs best on a given task without extensive experimentation. Hence, in the following section, we review a wide-ranging study of how these architectures perform on the TUSZ seizure detection task.

5. EXPERIMENTATION

Machine learning is at its core an experimental science when addressing real-world problems of scale. Real world data is complex and poses many challenges that require a wide variety of technologies to solve and can mask the benefits of one specific algorithm. Therefore, it is important that a rigorous evaluation paradigm be used to guide architecture decisions. In this chapter, we are focusing on the TUSZ Corpus because it is a very comprehensive dataset and it offers a very challenging task.

The evaluation of machine learning algorithms in biomedical fields for applications involving sequential data lacks standardization. Common quantitative scalar evaluation metrics such as sensitivity and

specificity can often be misleading depending on the requirements of the application. Evaluation metrics must ultimately reflect the needs of users yet be sufficiently sensitive to guide algorithm development. Feedback from critical care clinicians who use automated event detection software in clinical applications has been overwhelmingly emphatic that a low false alarm rate, typically measured in units of the number of errors per 24 hours, is the single most important criterion for user acceptance. Though using a single metric is not often as insightful as examining performance over a range of operating conditions, there is a need for a single scalar figure of merit. Shah et al. (2019) discuss the deficiencies of existing metrics for a seizure detection task and proposes several new metrics that offer a more balanced view of performance. In this section, we compare the architectures previously described using one of these measures, the Any-Overlap Method (OVL). We also provide detection error tradeoff (DET) curves (Fiscus et al., 2007).

5.1. Evaluation Metrics

Researchers in biomedical fields typically report performance in terms of sensitivity and specificity (Japkowicz et al., 2014). In a two-class classification problem such as seizure detection, we can define four types of errors:

- Positives (TP): the number of 'positives' detected correctly
- True Negatives (TN): the number of 'negatives' detected correctly
- False Positives (FP): the number of 'negatives' detected as 'positives'
- False Negatives (FN): the number of 'positives' detected as 'negatives'

Sensitivity ($TP/(TP+FN)$) and specificity ($TN/(TN+FP)$) are derived from these quantities. There are a large number of auxiliary measures that can be calculated from these four basic quantities that are used extensively in the literature. For example, in information retrieval applications, systems are often evaluated using accuracy ($(TP+TN)/(TP+FN+TN+FP)$), precision ($TP/(TP+FP)$), recall (another term for sensitivity) and F1 score ($(2 \cdot \text{Precision} \cdot \text{Recall})/(\text{Precision} + \text{Recall})$). However, none of these measures address the time scale on which the scoring must occur or how you score situations where the mapping of hypothesized events to reference events is ambiguous. These kinds of decisions are critical in the interpretation of scoring metrics such as sensitivity for many sequential decoding tasks such as automatic seizure detection (Shah et al., 2019; Fiscus et al., 2007).

In some applications, it is preferable to score every unit of time. With multichannel signals, such as EEGs, scoring for each channel for each unit of time might be appropriate since significant events such as seizures occur on a subset of the channels present in the signal. However, it is more common in the literature to simply score a summary decision per unit of time, such as every 1 sec, that is based on an aggregation of the per-channel inputs (e.g., a majority vote). We refer to this type of scoring as epoch-based (Liu et al., 1992; Navakatikyan et al., 2006). An alternative, that is more common in speech and image recognition applications, is term-based (Fiscus et al., 1992; Xiong et al., 2006), in which we consider the start and stop time of the event, and each event identified in the reference annotation is counted once. There are fundamental differences between the two conventions. For example, one event containing many epochs will count more heavily in an epoch-based scoring scenario. Epoch-based scoring generally weights the duration of an event more heavily since each unit of time is assessed independently.

Term-based metrics score on an event basis and do not count individual frames. A typical approach for calculating errors in term-based scoring is the Any-Overlap Method (OVL) (Shah et al., 2019). This approach is illustrated in Figure 11. TPs are counted when the hypothesis overlaps with the reference annotation. FPs correspond to situations in which a hypothesis does not overlap with the reference.

OVL is a more permissive metric that tends to produce much higher sensitivities. If an event is detected in close proximity to a reference event, the reference event is considered correctly detected. If a long event in the reference annotation is detected as multiple shorter events in the hypothesis, the reference event is also considered correctly detected. Multiple events in the hypothesis annotation corresponding to the same event in the reference annotation are not typically counted as FAs. Since the FA rate is a very important

measure of performance in critical care applications, this is another cause for concern. However, since OVLP metric is the most popular choice in the neuroengineering community, we present our results in terms of OVLP.

Note that results are still reported in terms of sensitivity, specificity and false alarm rate. But, as previously mentioned, how one measures the errors that contribute to these measures is open for interpretation. Shah et al. (2019) studied this problem extensively and showed that many of these measures correlate and are not significantly different in terms of the rank ordering and statistical significance of scoring differences for the TUSZ task. We provide a software package that allows researchers to replicate our metrics and reports on many of the most popular metrics (Shah & Picone, 2019).

5.2. Postprocessing with Heuristics Improves Performance

Because epoch-based scoring produces a hypothesis every epoch (1 sec in this case), and these are scored against annotations that are essentially asynchronous, there is an opportunity to improve performance by examining sequences of epochs and collapsing multiple events into a single hypothesis. We have experimented with heuristic approaches to this and deep learning-based approaches and have found no significant advantage for the deep learning approaches. As is well known in machine learning research, a good heuristic can be very powerful. We apply a series of heuristics, summarized in Figure 12, to improve performance. These heuristics are very important in reducing the false alarm rate to an acceptable level.

The first heuristic we apply is a popular method that focuses on a model's confidence in its output. Probabilistic filters (Sundermeyer et al., 2015) are implemented to only consider target events which are detected above a specified probability threshold. This method tends to suppress spurious long duration events (e.g., slowing) and extremely short duration events (e.g., muscle artifacts). This decision function is applied on the seizure (target) labels only. We compare each seizure label's posterior with the threshold value. If the posterior is above the threshold, the label is kept as is; otherwise, it is changed to the non-seizure label, which we denote "background."

Our second heuristic was developed after performing extensive error analysis. The most common types of errors we observed were false detections of background events as seizures (FPs) which tend to occur in bursts. Usually these erroneous bursts occur for a very small duration of time (e.g., 3 to 7 seconds). To suppress these, any seizure event whose duration is below a specified threshold is automatically considered as a non-seizure, or background, event.

Finally, we also implement a smoothing method that collapses sequences of two seizure events separated by a background event into one long seizure event. This is typically used to eliminate spurious background events. If seizures are observed in clusters separated by small intervals of time classified as background events, these isolated events are most likely part of one longer seizure event. In this method, we apply a nonlinear function that computes a pad time to extend the duration of an isolated event. If the modified endpoint of that event overlaps with another seizure event, the intervening background event is eliminated. We used a simple regression approach to derive a quadratic function that produces a padding factor: $w(x) = -0.0083d^2 + 0.45d - 0.66$, where d is the duration of the event. This method tends to reduce isolated background events when they are surrounding by seizure events, thereby increasing the specificity.

The combination of these three postprocessing methods tends to decrease sensitivity slightly and reduce false alarms by two orders of magnitude, so their impact is significant. The ordering in which these methods is applied is important. We apply them in the order described above to achieve optimal performance.

5.3. A Comprehensive Evaluation of Hybrid Approaches

A series of experiments was conducted to optimize the feature extraction process. These are described in detail in Harati et al. (2015). Subsequent attempts to replace feature extraction with deep learning-based

approaches have resulted in a slight degradation in performance. A reasonable tradeoff between computational complexity and performance was to split the 10 sec window, popular with neurologists who manually interpret these waveforms, into 1 sec epochs, and to further subdivide these into 0.1 sec frames. Hence, features were computed every 0.1 sec using a 0.2 sec overlapping analysis window. The output of the feature extraction system is 22 channels of data, where in each channel, a feature vector of dimension 26 corresponds to every 0.1 secs. This type of analysis is very compatible with the way HMM systems operate, so it was a reasonable starting point for this work.

We next evaluated several architectures using these features as inputs on TUSZ. These results are presented in Table 2. The related DET curve is illustrated in Figure 13. An expanded version of the DET curve in Figure 13 that compares the performance of these architectures in a region of the DET curve where the false positive rate, also known as the false alarm (FA) rate, is low is presented in Figure 14. Since our focus is achieving a low false alarm rate, behavior in this region of the DET curve is very important. As previously mentioned, these systems were evaluated using the OVLP method, though results are similar for a variety of these metrics.

It is important to note that the accuracy reported here is much lower than what is often published in the literature on other seizure detection tasks. This is due to a combination of factors including (1) the neuroscience community has favored a more permissive method of scoring that tends to produce much higher sensitivities and lower false alarm rates; and (2) TUSZ is a much more difficult task than any corpus previously released as open source. The evaluation set was designed to be representative of common clinical issues and includes many challenging examples of seizures. We have achieved much higher performance on other publicly available tasks, such as the Children's Hospital of Boston MIT (CHB-MIT) Corpus (Obeid et al., 2014), and demonstrated that the performance of these techniques exceeds that of published or commercially-available technology. TUSZ is simply a much more difficult task and one that better represents the clinical challenges this technology faces.

Also, note that the HMM baseline system, which is shown in the first row of Table 2, and channel-based LSTM, which is shown in the last row of Table 2, operate on each channel independently. The other methods consider all channels simultaneously by using a supervector that is a concatenation of the feature vectors for all channels. The baseline HMM system only classifies epochs (1 sec in duration) using data from within that epoch. It does not look across channels or across multiple epochs when performing epoch-level classification.

From Table 2 we can see that adding a deep learning structure for temporal and spatial analysis of EEGs can decrease the false alarm rate dramatically. Further, by comparing the results of HMM/SdA with HMM/LSTM, we find that a simple one-layer LSTM performs better than 3 layers of SdA due to LSTM's ability to explicitly model long-term dependencies. Note that in this case the complexity and training time of these two systems is comparable.

The best overall systems shown in Table 2 are CNN/LSTM and channel-based LSTM. CNN/LSTM is a doubly deep recurrent convolutional structure that models both spatial relationships (e.g., cross-channel dependencies) and temporal dynamics (e.g., spikes). For example, CNN/LSTM does a much better job rejecting artifacts that are easily confused with spikes because these appear on only a few channels, and hence can be filtered based on correlations between channels. The depth of the convolutional network is important since the top convolutional layers tend to learn generic features while the deeper layers learn dataset specific features. Performance degrades if a single convolutional layer is removed. For example, removing any of the middle convolutional layers results in a loss of about 4% in the sensitivity. However, it is important to note that the computational complexity of the channel-based systems is significantly higher than the systems that aggregate channel-based features into a single vector, since the channel-based systems are decoding each channel independently.

As shown in Figure 13 and Figure 13 we find that CNN/LSTM has a significantly lower FA rate than CNN/GRU. We speculate that this is due to the fact that while GRU unit controls the flow of information like the LSTM unit, it does not have a memory unit. LSTMs can remember longer sequences better than GRUs. Since seizure detection requires modeling long distance relationships, we believe this explains why there is a difference in performance between the two systems.

The time required for training for CNN/GRU was 10% less than CNN/LSTM. The training time of these two systems is comparable since most of the cycles are spent training the convolutional layers. We also observe that the ResNet structure improves the performance of CNN/MLP, but the best overall system is still CNN/LSTM.

We have also conducted an open-set evaluation of the best systems, CNN/LSTM and channel-based LSTM, on a completely different corpus – DUSZ. These results are shown in Table 3. A DET curve is shown in Figure 15. This is an important evaluation because none of these systems were exposed to DUSZ data during training or development testing. Parameter optimizations were performed only on TUSZ data. As can be seen, at high FA rates, performance between the two systems is comparable. At low FA rates, however, CNN/LSTM performance on TUSZ is lower than on DUSZ. For channel-based LSTM, in the region of low FA rate, performance on TUSZ and DUSZ is very similar. This is reflected by the two middle curves in Figure 15. The differences in performance for channel-based LSTM when the data changes are small. However, for CNN/LSTM, which gives the best overall performance on TUSZ, performance decreases rapidly on DUSZ. Recall that we did not train these systems on DUSZ – this is true open set testing. Hence, we can conclude in this limited study that channel-based LSTM generalized better than the CNN/LSTM system.

5.4. Optimization of Core Components

Throughout these experiments we observed that the choice of optimization method had a considerable impact on performance. The CNN/LSTM system was evaluated using a variety of optimization methods, including Stochastic gradient descent (SGD) (Kingma et al. 2015), RMSprop (Bottou et al. 2004), Adagrad (Tieleman et al. 2012), Adadelta (Duchi et al. 2011), Adam (Kingma et al. 2015), Adamax (Kingma et al. 2015) and Nadam (Zeiler et al. 2013). These results are shown in Table 4. The best performance is achieved with Adam, a learning rate of $\alpha = 0.0005$, a learning rate decay of 0.0001, exponential decay rates of $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for the moment estimates and a fuzz factor of $\epsilon = 10^{-8}$. The parameters follow the notation described in (Kingma et al. 2015). Table 4 also illustrates that Nadam delivers comparable performance to Adam. Adam combines the advantages of AdaGrad which works well with sparse gradients, and RMSProp which works well in non-stationary settings.

Similarly, we evaluated our CNN/LSTM using different activation functions, as shown in Table 5. ELU delivers a small but measurable increase in sensitivity, and more importantly, a reduction in false alarms. The ELU activation function is defined as:

$$f(x) = \begin{cases} x & x > 0 \\ \alpha \cdot (e^x - 1) & x \leq 0 \end{cases}, \quad (8)$$

where α is slope of negative section. The derivative of ELU is:

$$\hat{f}(x) = \begin{cases} 1 & x > 0 \\ \alpha \cdot e^x & x \leq 0 \end{cases}, \quad (9)$$

and ReLU activation function is defined as:

$$f(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}. \quad (10)$$

The derivative of ReLU is:

$$\hat{f}(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases} \quad (11)$$

ELU is very similar to ReLU except for negative inputs. ReLUs and ELUs accelerate learning [102] decreasing the gap between the normal gradient and the unit natural gradient (Clevert et al., 2017). ELUs push the mean towards zero but with a significantly smaller computational footprint. In the region where the input is negative ($x < 0$), since an ReLU's gradient is zero, the weights will not get adjusted. Those neurons which connect into that state will stop responding to variations in error or input. This is referred to as the dying ReLU problem. But unlike ReLUs, ELUs have a clear saturation plateau in their negative region, allowing them to learn a more robust and stable representation.

Determining the proper initialization strategy for the parameters in the model is part of the difficulty in training. Hence, we investigated a variety of initialization methods using the CNN/LSTM structure introduced in Figure 8. These results are presented in Table 6. The related DET curve is illustrated in Figure 16. In our experiments, we observed that proper initialization of weights in a convolutional recurrent neural network is critical to convergence. For example, initialization of tensor values to zero or one completely stalled the convergence process. Also, as we can see in Table 6, the FA rate of the system in the range of 30% sensitivity can change from 7 to 40, for different initialization methods. This decrease in performance and deceleration of convergence arises because some initializations can result in the deeper layers receiving inputs with small variances, which in turn slows down back propagation, and retards the overall convergence process.

Best performance is achieved using orthogonal initialization (Saxe et al., 2014). This method [38] simple yet effective way of combatting exploding and vanishing gradients. In orthogonal initialization, the weight matrix is chosen as a random orthogonal matrix, i.e., a square matrix $m \times W$ for which $W^T W = I$. Typically, the orthogonal matrix [3] obtained from the QR decomposition of a matrix of random numbers drawn from a normal distribution. Orthogonal matrices preserve the norm of a vector and their eigenvalues have an absolute value of one. This means that no matter how many times we perform repeated matrix multiplication, the resulting matrix doesn't explode or vanish. Also, in orthogonal matrices, columns and rows are all orthonormal to one another, which helps the weights to learn different input features. For [38] mple, if we apply orthogonal initialization on a CNN architecture, in each layer, each channel has a weight vector that is orthogonal to the weight vectors of the other channels.

Overfitting is a serious problem in deep neural nets with many parameters. We have explored five popular regularization methods to address this problem. [50] techniques collectively known as L1, L2 and L1/L2 (Goodfellow et al., 2016) prevent overfitting by adding a [22] larization term to the loss function. The L1 regularization technique, also known as Lasso regression, is defined as adding the sum of weights to the loss function:

$$\text{Cost Function} = \text{Loss Function} + \lambda \sum_{i=1}^k |w_i|, \quad (12)$$

where w is the weight [47] or and λ is a regularization parameter. The L2 technique, also known as ridge regression, is defined as adding the sum of the square of the weights to the loss function:

$$\text{Cost Function} = \text{Loss Function} + \lambda \sum_{i=1}^k w_i^2. \quad (13)$$

The L1/L2 technique is a combination of both techniques:

$$\text{Cost Function} = \text{Loss Function} + \lambda \sum_{i=1}^k |w_i| + \lambda \sum_{i=1}^k w_i^2. \quad (14)$$

In an alternative approach, we used dropout to prevents units from co-adapting too much by randomly dropping units and their connections from the neural network during training.

Also, we studied the impact of introducing zero-centered Gaussian noise to the network. In this regularization method [124] which is considered a random data augmentation method (Perez & Wang, 2017), we add zero-centered Gaussian noise with a standard deviation of 0.2 to all hidden layers in the network as

well as the visible or input layer. The results of these experiments are presented in Table 7 along with a DET curve in Figure 17.

While L1/L2 regularization has the best overall performance, in the region where FA rates are low, the dropout method delivers a lower FA rate. Additionally, we found that the primary error modalities observed were false alarms generated during brief delta range slowing patterns such as intermittent rhythmic delta activity (von Weltin et al., 2017). Our closed-loop experiments demonstrated that all regularization methods presented in Table 7, unfortunately, tend to increase the false alarm rate for slowing patterns.

Finally, in Figure 18, an example of an EEG that is generated by the DCGAN structure of Figure 6 is shown. Note that to generate these EEGs, we use a generator block in DCGAN in which each EEG signal has a 7 sec duration. We apply a 25 Hz low pass filter on the output of DCGAN, since most of the cerebral signals observed in scalp EEGs fall in the range of 1–20 Hz (in standard clinical recordings, activity below or above this range is likely to be an artifact). Unfortunately, in a simple pilot experiment in which we randomly mixed actual EEGs with synthetic EEGs, expert annotators could easily detect the synthetic EEGs, which was a bit discouraging. Seizures in the synthetic EEGs were sharper and more closely resembled a slowing event. Clearly, more work is needed with this architecture.

However, our expert annotators also noted that the synthetic EEGs did exhibit focality. An example of focality is when activity is observed on the CZ-C4 channel, we would expect to observe the inverse of this pattern on the C4-T4 channel. As can be seen in Figure 18, in last two seconds of the generated EEG, we observe slowing activity on the CZ-C4 channel and the inverse pattern of the same slowing activity on the C4-T4 channel. Hence, it is possible to generate synthetic multi-channel EEG signals with DCGAN that resemble clinical EEGs. However, DCGAN is not yet at the point where it is generating data that is resulting in an improvement in the performance of our best systems.

6. CONCLUSIONS

EEGs remain one of the main clinical tools that physicians use to understand brain function. New applications of EEGs are emerging including diagnosis of head trauma-related injuries which offer the potential to vastly expand the market for EEGs. A board-certified EEG specialist is required by law to interpret an EEG and produce a diagnosis. Since it takes several years of additional training post-medical school for a physician to qualify as a clinical specialist, the ability to generate data far exceeds the available expertise to interpret these data, creating a critical bottleneck. Despite rapid advances in deep learning in recent years, automatic interpretation of EEGs is still a very challenging problem.

We have introduced a variety of deep learning architectures for automatic classification of EEGs including a hybrid architecture that integrates CNN and LSTM technology. Two systems are particularly promising: CNN/LSTM and channel-based LSTM. While these architectures deliver better performance than other deep structures, their performance still does not meet the needs of clinicians. Human performance on similar tasks is in the range of 75% sensitivity with a false alarm rate of 1 per 24 hours (Haider et al., 2016). The false alarm rate is particularly important to critical care applications since it impacts the workload experienced by healthcare providers.

The primary error modalities for our deep learning-based approaches were false alarms generated during brief delta range slowing patterns such as intermittent rhythmic delta activity. A variety of these types of artifacts have been observed during inter-ictal and post-ictal stages. Training models on such events with diverse morphologies is potentially one way to reduce the remaining false alarms. This is one reason we are continuing our efforts to annotate a larger portion of TUSZ.

We are also exploring the potential of supervised GAN frameworks for spatio-temporal modeling of EEGs. Most of the research on GANs is focused on either unsupervised learning or supervised learning using conditional GANs. Given that the annotation process to produce accurate labels is expensive and time-

10

consuming, we are exploring semi-supervised learning in which only a small fraction of the data has labels. GANs can be used to perform semi-supervised classification by using a generator-discriminator pair to learn an unconditional model of data and then tune the discriminator using the small amount of labeled data for prediction.

2

We are also continuing to manually label EEG data. We invite you to register at our project web site, www.isip.piconepress.com/projects/tuh_eeg/, to be kept aware of the latest developments.

REFERENCES

- 2 American Clinical Neurophysiology Society. (2012). ACNS Standardized ICU EEG Nomenclature v. 2012. March 1, 2019. <https://www.acns.org/pdf/guidelines/Guideline-14-pocket-version.pdf>.
- 95 Ahmedt-Aristizabal, D., Fookes, C., Denman, S., Nguyen, K., Sridharan, S., & 101 nisio, S. (2019). Aberrant epileptic seizure identification: A computer vision perspective. *Seizure: European Journal of Epilepsy*, 65, 65–71. <https://doi.org/10.1016/j.seizure.2018.12.017>.
- 12 Alotaiby, T., Alshebeili, S., Alshawi, T., Ahmad, I., & Abd El-Samie, F. (2014). EEG seizure detection and prediction algorithms: a survey. *EURASIP Journal on Advances in Signal Processing*, (16876180), 1–21. <https://doi.org/10.1186/1687-6180-2014-183>.
- 20 Bahl, L., Brown, P., de Souza, P., & Mercer, R. (1986). Maximum mutual information estimation of hidden Markov model parameters for speech recognition. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (Vol. 11, pp. 49–52). Tokyo, Japan. <https://doi.org/10.1109/ICASSP.1986.1169179>.
- 11 Baumgartner, C., & Koren, J. P. (2018). Seizure detection using scalp-EEG. *Epilepsia*, 59(S1), 14–22. <https://doi.org/10.1111/epi.14052>.
- 59 Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy layer-wise training of deep networks. *Advances in Neural Information Processing System* (pp. 153–160). Vancouver, B.C., Canada. <https://doi.org/10.1.1.70.2022>.
- 12 Bishop, C. (2011). *Pattern Recognition and Machine Learning* (2nd ed.). New York, New York, USA: Springer. <https://www.springer.com/us/book/9780387310732>.
- 91 Boashash, B. (2015). *Time-Frequency Signal Analysis and Processing: A Comprehensive Reference* (2nd ed.). London, UK: Academic Press, Inc. https://www.abebooks.com/servlet/BookDetailsPL?bi=20272644641&searchurl=isbn%3D0123984998%26sortby%3D17&cm_sp=snippet-_srp1-_title1.
- 90 Bottou, L., & Lecun, Y. (2004). Large Scale Online Learning. *Advances in Neural Information Processing Systems*, 217–225. <https://papers.nips.cc/paper/2365-large-scale-online-learning>.
- 49 Bridi, A. C., Louro, T. Q., & Da Silva, R. C. L. (2014). Clinical Alarms in intensive care: implications of alarm fatigue for the safety of patients. *Revista Latino-Americana de Enfermagem*, 22(6), 1034. <https://doi.org/10.1590/0104-1169.3488.2513>.
- 14 Cho, K., Merrienboer, B. Van, Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734. <https://doi.org/10.3115/v1/D14-1179>.
- 6 Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *NIPS Deep Learning and Representation Learning Workshop*, 1–9. <https://doi.org/http://arxiv.org/abs/1412.3555>.

- 19 Clevert, D., Unterthiner, T., & Hochreiter, S. (2016). Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *Proceedings of the International Conference on Learning Representations (ICLR)* (pp. 1–14). San Juan, Puerto Rico. <https://arxiv.org/abs/1511.07289>.
- 82 Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297. <https://doi.org/10.1007/BF00994018>.
- 4 Da Rocha Garrit, P. H., Guimaraes Moura, A., Obeid, I., & Picone, J. (2015). Wavelet Analysis for Feature Extraction on EEG Signals. Presented at the NEDC Summer Research Experience for Undergraduates, Department of Electrical and Computer Engineering, Temple University (p. 1). Philadelphia, Pennsylvania, USA. http://www.isip.piconepress.com/publications/unpublished/conferences/2015/summer_of_code/wavelets/.
- 8 Deburghraeve, W., Cherian, P. J., De Vos, M., Swarte, R. M., Blok, J. H., Visser, G. H., ... Van Huffel, S. (2008). Automated neonatal seizure detection mimicking a human observer reading EEG. *Clinical Neurophysiology*, 119(11), 2447–2454. <https://doi.org/10.1016/j.clinph.2008.07.281>.
- 17 Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12, 2121–2159. <https://dl.acm.org/citation.cfm?id=2021068>.
- 4 Ebersole, J. S., & Pedley, T. A. (2014). *Current practice clinical electroencephalography* (4th ed.). Philadelphia, Pennsylvania, USA: Wolters Kluwer. <http://www.amazon.com/Current-Practice-Clinical-Electroencephalography-Ebersole/dp/145113195X>.
- 6 Eichler, M., Dahlhaus, R., & Dueck, J. (2017). Graphical Modeling for Multivariate Hawkes Processes with Nonparametric Link Functions. *Journal of Time Series Analysis*, 38(2), pp.225–242. <https://arxiv.org/abs/1605.06759>.
- Ercegovac, M., & Berisavac, I. (2015). Importance of EEG in intensive care unit. *Clinical Neurophysiology*, 126(9), e178–e179. <https://doi.org/10.1016/j.clinph.2015.04.027>.
- 12 Fiscus, J., Ajot, J., Garofolo, J., & Doddington, G. (2007). Results of the 2006 Spoken Term Detection Evaluation. *Proceedings of the SIGIR 2007 Workshop: Searching Spontaneous Conversational Speech* (pp. 45–50). Amsterdam, Netherlands. <https://doi.org/978-90-365-2542-8>.
- 2 Golmohammadi, M., Harati Nejad Torbati, A., Lopez, S., Obeid, I., & Picone, J. (2019). Automatic Analysis of EEGs Using Big Data and Hybrid Deep Learning Architectures. *Frontiers in Human Neuroscience*, 1–30. <https://doi.org/10.3389/fnhum.2019.00076> (abstract – in print). https://www.isip.piconepress.com/publications/unpublished/journals/2018/frontiers_neuroscience/hybrid/.
- 4 Golmohammadi, M., Ziyabari, S., Shah, V., Obeid, I., & Picone, J. (2018). Deep Architectures for Spatio-Temporal Modeling: Automated Seizure Detection in Scalp EEGs. *Proceedings of the IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 1–6). Orlando, FL, USA. <http://dx.doi.org/10.1109/ICMLA.2018.00118>.
- 4 Golmohammadi, M., Ziyabari, S., Shah, V., Obeid, I., & Picone, J. (2017). Gated Recurrent Networks for Seizure Detection. I. Obeid & J. Picone (Eds.), *Proceedings of the IEEE Signal Processing in Medicine and Biology Symposium* (pp. 1–5). Philadelphia, Pennsylvania, USA: IEEE. <https://doi.org/10.1109/SPMB.2017.8257020>.

- 61 Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning* (1st ed.). Cambridge, Massachusetts, USA: MIT Press. <https://doi.org/https://mitpress.mit.edu/books/deep-learning>.
- 57 Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative Adversarial Nets. *Proceedings of the Conference on Neural Information Processing Systems*, 2672–2680. <https://doi.org/10.1017/CBO9781139058452>.
- 100 Gotman, J. (1999). Automatic detection of seizures and spikes. *Journal of Clinical Neurophysiology: Official Publication of the American Electroencephalographic Society*, 16(2), 130–140. <http://www.ncbi.nlm.nih.gov/pubmed/10359498>.
- 16 Graves, A., Mohamed, A., & Hinton, G. (2013). Speech Recognition with Deep Recurrent Neural Networks. *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (3), 6645–6649. <https://doi.org/10.1109/ICASSP.2013.6638947>.
- 94 Haider, H. A., Steller, R. D., Hahn, C. B., Westover, M. J., Halford, J. W., Lee, J. M., & LaRoche, S. (2016). Sensitivity of quantitative EEG for seizure identification in the intensive care unit. *Neurology*, 87(9), 935–944. <https://doi.org/10.1212/WNL.0000000000003034>.
- 4 Harati Nejad Torbati, A. H., Golmohammadi, M., Jacobson, M., Lopez, S., Obeid, I., Picone, J., & Tobochnik, S. (2016). Automatic Interpretation of EEGs for Clinical Decision Support. In *American Clinical Neurophysiology Society (ACNS) Annual Meeting* (p. 1). Orlando, Florida, USA. https://www.isip.piconepress.com/publications/conference_presentations/2016/acns/autoeeg/.
- 2 Harati, A., Golmohammadi, M., Lopez, S., Obeid, I., & Picone, J. (2015). Improved EEG Event Classification Using Differential Energy. I. Obeid & J. Picone (Eds.), *Proceedings of the IEEE Medical Processing in Medicine and Biology Symposium* (pp. 1–4). Philadelphia, Pennsylvania, USA. <https://doi.org/10.1109/SPMB.2015.7405421>.
- 2 Harati, A., Lopez, S., Obeid, I., Jacobson, M., Tobochnik, S., & Picone, J. (2014). The TUI EEG Corpus: A Big Data Resource for Automated EEG Interpretation. I. Obeid & J. Picone (Eds.), *Proceedings of the IEEE Signal Processing in Medicine and Biology Symposium* (pp. 1–5). Philadelphia, Pennsylvania, USA. <https://doi.org/10.1109/SPMB.2014.7002953>.
- 4 Obeid, I., Harati, A., Jacobson, M., & Picone, J. (2014). A Big-Data Approach to Automated EEG Labeling. In *Proceedings of Neuroinformatics*. Leiden, The Netherlands. https://www.frontiersin.org/10.3389/conf.fninf.2014.18.00094/event_abstract.
- 46 He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778). Las Vegas, Nevada, USA. <https://doi.org/10.1109/CVPR.2016.90>.
- 6 Hermans, M., & Schrauwen, B. (2013). Training and Analysing Deep Recurrent Neural Networks. *Advances in Neural Information Processing Systems*, 190–198. <https://papers.nips.cc/paper/5166-training-and-analysing-deep-recurrent-neural-networks>.
- 36 Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7), 1527–1554. <https://doi.org/10.1162/neco.2006.18.7.1527>.
- 77 Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–80. <https://doi.org/10.1162/neco.1997.9.8.1735>.

- 4 Huang, K., & Picone, J. (2002). Internet-Accessible Speech Recognition Technology. In *Proceedings of IEEE Midwest Symposium on Circuits and Systems* (p. III-73-III-76). Tulsa, Oklahoma, USA. <https://doi.org/10.1109/MWSCAS.2002.1186973>.
- Ilmoniemi, R., & Sarvas, J. (1981). *Brain Signals: Physics and Mathematics of MEG and EEG* (1st ed.). Boston, Massachusetts, USA: The MIT Press. <https://mitpress.mit.edu/books/brain-signals>.
- 6 Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of the International Conference on Machine Learning (ICML)* (pp. 448–456). Lille, France. <https://arxiv.org/abs/1502.03167>.
- 2 Jahankhani, P., Kodogiannis, V., & Revett, K. (2006). EEG Signal Classification Using Wavelet Feature Extraction and Neural Networks. *IEEE Journal of Modern Computing (JVA '06)*, 120–124. <https://doi.org/10.1109/JVA.2006.17>.
- 99 Japkowicz, N., & El-Hajj, M. (2014). *Evaluating Learning Algorithms: a classification perspective*. New York City, New York, USA: Cambridge University Press. <https://doi.org/10.1017/CBO9780511921803>.
- 50 Jelinek, F. (1997). *Statistical Methods for Speech Recognition*. Boston, Massachusetts, USA: The MIT Press. <https://doi.org/https://mitpress.mit.edu/books/statistical-methods-speech-recognition>.
- 13 Kingma, D. P., & Ba, J. L. (2015). Adam: a Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations* (pp. 1–15). San Diego, California, USA. <https://doi.org/https://arxiv.org/abs/1412.6980>.
- 88 Krauss, G. L., & Fisher, R. S. (2011). *The Johns Hopkins Atlas of Digital EEG: An Interactive Training Guide* (2nd ed.). Baltimore, Maryland, USA: Johns Hopkins University Press. <https://www.amazon.com/Johns-Hopkins-Atlas-Digital-EEG/dp/0801897335>.
- 105 Kubota, Y., Nakamoto, H., Egawa, S., & Yamata, T. (2018). Continuous EEG monitoring in ICU. *Journal of Intensive Care*, 6(1), 39. <https://doi.org/10.1186/s40560-018-0310-z>.
- 14 Lang, K. J., Waibel, A., & Hinton, G. E. (1990). A time-delay neural network architecture for isolated word recognition. *Neural Networks*, 3(1), 23–43. [https://doi.org/10.1016/0893-6080\(90\)90044-L](https://doi.org/10.1016/0893-6080(90)90044-L).
- 5 LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <http://dx.doi.org/10.1038/nature14539> <https://doi.org/10.1038/nature14539>.
- 64 Levy, A., & Lindenbaum, M. (2000). Sequential Karhunen-Loeve basis extraction and its application to images. *Proceedings of the IEEE Transactions on Image Processing*, 9(8), 1371–1374. <https://doi.org/10.1109/ICIP.1998.723422>.
- 45 Li, P., Wang, X., Li, F., Zhang, R., Ma, T., Peng, Y., ... Xu, P. (2015). Autoregressive model in the L_p norm space for EEG analysis. *Journal of Neuroscience Methods*, 240, 170–178. <https://doi.org/10.1016/j.jneumeth.2014.11.007>.
- 5 Li, Y., Luo, M.-L., & Li, K. (2016). A multiwavelet-based time-varying model identification approach for frequency analysis of EEG signals. *Neurocomputing*, 193, 106–114. <https://doi.org/10.1016/j.neucom.2016.01.062>.

- 58 Liu, A., Hahn, J. S., Heldt, G. P., & Coen, R. W. (1992). Detection of neonatal seizures through computerized EEG analysis. *Electroencephalography and Clinical Neurophysiology*, 82(2), 32–37. [https://doi.org/10.1016/0013-4694\(92\)90179-L](https://doi.org/10.1016/0013-4694(92)90179-L).
- 80 Lopez, S. (2017). *Automated Identification of Abnormal Adult Electroencephalograms*. Temple University. https://www.isip.piconepress.com/publications/ms_theses/2017/abnormal/thesis/.
- 2 Lopez, S., Golmohammadi, M., Obeid, I., & Picone, J. (2016). An Analysis of Two Common Reference Points for EEGs. In I. Obeid & J. Picone (Eds.), *Proceedings of the IEEE Signal Processing in Medicine and Biology Symposium* (pp. 1–4). Philadelphia, Pennsylvania, USA. <https://doi.org/10.1109/SPMB.2016.7846854>.
- 4 Lu, S., & Picone, J. (2013). Fingerspelling Gesture Recognition Using A Two-Level Hidden Markov Model. In *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (ICPVR)* (pp. 538–543). Las Vegas, Nevada, USA. http://www.isip.piconepress.com/publications/conference_proceedings/2013/ipcv/asl_hmm/.
- 10 Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshop on Deep Learning for Audio, Speech and Language Processing* (p. 6). Atlanta, Georgia, USA. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.693.1422>.
- 63 Mirowski, P., Madhavan, D., Lecun, Y., & Kuzniecky, R. (2009). Classification of patterns of EEG synchronization for seizure prediction. *Clinical Neurophysiology*, 120(11), 1927–1940. <https://doi.org/10.1016/j.clinph.2009.09.002>.
- 62 Nair, V., & Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. *Proceedings of the International Conference on Machine Learning (ICML)*, (3), 807–814. <https://doi.org/10.1.1.165.6419>.
- Natus Medical. (2019). Nicolet® NicVue Connectivity Solution. March 1, 2019. <https://neuro.natus.com/products-services/nicolet-nicvue-connectivity-solution>.
- 8 Navakatikyan, M. A., Colditz, P. B., Burke, C. J., Inder, T. E., Richmond, J., & Williams, C. E. (2006). Seizure detection algorithm for neonates based on wave-sequence analysis. *Clinical Neurophysiology*, 117(6), 1190–1203. <http://dx.doi.org/10.1016/j.clinph.2006.02.016>.
- 44 Ney, J. P., van der Goes, D. N., Nuwer, M. R., & Nelson, L. (2016). Continuous and routine EEG in intensive care: utilization and outcomes, United States 2005–2009. *Neurology*, 81(23), 2002–2008. <https://doi.org/10.1212/01.wnl.0000436948.93399.2a>.
- Nihon Kohden Corporation. (March 1, 2019). <https://us.nihonkohden.com/products/eeg-1200>.
- 4 Obeid, I., & Picone, J. (2016). The Temple University Hospital EEG Data Corpus. *Frontiers in Neuroscience, Section Neural Technology*, 10, 196. <http://dx.doi.org/10.3389/fnins.2016.00196>.
- 4 Obeid, I., & Picone, J. (2018). Machine Learning Approaches to Automatic Interpretation of EEGs. In E. Sejdik & T. Falk (Eds.), *Signal Processing and Machine Learning for Biomedical Big Data* (1st ed., p. 30). Boca Raton, Florida, USA: Taylor & Francis Group. <https://doi.org/10.1201/9781351061223>.

- Pandey, P. (2019). Deep Generative Models. <https://towardsdatascience.com/deep-generative-models-25ab2821afd3>.
- Parker, D., Picone, J., Harati, A., Lu, S., Jenkyns, M., & Polgreen, P. (2013). Detecting paroxysmal coughing from pertussis cases using voice recognition technology. *PLoS ONE*, 8(12), e82971. <https://doi.org/10.1371/journal.pone.0082971>.
- Perez, L., & Wang, J. (2017). The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *ArXiv*. <https://arxiv.org/abs/1712.04621>.
- Picone, J. (1990). Continuous Speech Recognition Using Hidden Markov Models. *IEEE ASSP Magazine*, 7(3), 26–41. <https://doi.org/10.1109/53.54527>.
- Picone, J. (1993). Signal modeling techniques in speech recognition. *Proceedings of the IEEE*, 81(9), 1215–1247. <https://doi.org/10.1109/5.237532>.
- Radford, A., Metz, L., & Chintala, S. (2016). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *Proceedings of the International Conference on Learning Representations (ICLR)* (pp. 1–16). San Juan, Puerto Rico. <https://doi.org/https://arxiv.org/pdf/1511.06434.pdf>.
- Ramgopal, S. (2014). Seizure detection, seizure prediction, and closed-loop warning systems in epilepsy. *Epilepsy & Behavior*, 37, 291–307. <https://doi.org/10.1016/j.yebeh.2014.06.023>.
- Rodriguez-Bermudez, G., & Pedro J. Garcia-Laencina. (2015). Analysis of EEG Signals using Nonlinear Dynamics and Chaos: A review. *Applied Mathematics and Information Science*, 9(5), 2309–2321. <https://doi.org/10.12785/amis/090512>.
- Ross, D. A., Lim, J., Lin, R. S., & Yang, M. H. (2008). Incremental Learning for robust visual tracking. *International Journal of Computer Vision*, 77(1–3), 125–141. <https://doi.org/10.1007/s11263-007-0075-7>.
- Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved Techniques for Training GANs. In *Proceedings of Neural Information Processing Systems (NIPS)* (pp. 1–9). Barcelona, Spain. <https://papers.nips.cc/paper/6125-improved-techniques-for-training-gans.pdf>.
- Saon, G., Sercu, T., Rennie, S., & Kuo, H.-K. J. K. J. (2016). The IBM 2016 English Conversational Telephone Speech Recognition System. In *Proceedings of the Annual Conference of the International Speech Communication Association* (Vol. 08–12–Sept, pp. 7–11). <https://doi.org/10.21437/Interspeech.2016-1460>.
- Saxe, A. M., McClelland, J. L., & Ganguli, S. (2014). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *Proceedings of the International Conference on Learning Representations (ICLR)* (pp. 1–22). Banff, Canada. <https://doi.org/https://arxiv.org/abs/1312.6120>.
- Schad, A., Schindler, K., Schelter, B., Maiwald, T., Brandt, A., Timmer, J., & Schulze-Bonhage, A. (2008). Application of a multivariate seizure detection and prediction method to non-invasive and intracranial long-term EEG recordings. *Clinical Neurophysiology*, 119(1), 197–211. <http://www.sciencedirect.com/science/article/pii/S1388245707005986>.

- 53 Schindler, K., Wiest, R., Kollar, M., & Donati, F. (2001). Using simulated neuronal cell models for 11 detection of epileptic seizures in foramen ovale and scalp EEG. *Clinical Neurophysiology*, 112(6), 1006–17. [https://doi.org/10.1016/S1388-2457\(01\)00522-3](https://doi.org/10.1016/S1388-2457(01)00522-3).
- Shah, V., & Picone, J. (2019). NEDC Eval EEG: A Comprehensive Scoring Package for Sequential 118 Editing of Multichannel Signals. https://www.isip.piconepress.com/projects/tuh_eeg/downloads/nedc_eval_eeg/.
- Shah, V., Anstotz, R., Obeid, I., & Picone, J. (2018). Adapting an Automatic Speech Recognition System 111 Event Classification of Electroencephalograms. I. Obeid & J. Picone (Eds.), *Proceedings of the IEEE Signal Processing in Medicine and Biology Symposium (SPMB)* (p. 1). Philadelphia, Pennsylvania, USA. <https://doi.org/10.1109/SPMB.2018.8615625>.
- 28 Shah, V., Golmohammadi, M., Obeid, I., & Picone, J. (2019). Objective Evaluation Metrics for 23 Automatic Classification of EEG Events. *Journal of Neural Engineering*, 1–19. https://www.isip.piconepress.com/publications/unpublished/journals/2018/iop_jne/metrics/ (in review).
- 4 Shah, V., Golmohammadi, M., Ziyabari, S., von Weltin, E., Obeid, I., & Picone, J. (2017). Optimizing 4 Channel Selection for Seizure Detection. In I. Obeid & J. Picone (Eds.), *Proceedings of the IEEE Signal Processing in Medicine and Biology Symposium* (pp. 1–5). Philadelphia, Pennsylvania, USA: IEEE. <https://doi.org/10.1109/SPMB.2017.8257019>.
- 2 Shah, V., von Weltin, E., Lopez, S., McHugh, J. R., Veloso, L., Golmohammadi, M., ... Picone, J. (2018). The Temple University Hospital Seizure Detection Corpus. *Frontiers in Neuroinformatics*, 12, 83. <https://doi.org/10.3389/fninf.2018.00083>.
- 13 Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image 21 Recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)* (pp. 1–14). San Diego, California, USA. <https://doi.org/10.1016/j.infsof.2008.09.005>.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15, 1929–1958. <https://doi.org/10.1214/12-AOS1000>.
- 29 Subasi, A. (2007). EEG signal classification using wavelet feature extraction and a mixture of expert model. *Expert Systems with Applications*, 32(4), 1084–1093. <https://doi.org/10.1016/j.eswa.2006.02.005>.
- 5 Sundermeyer, M., Ney, H., & Schluter, R. (2015). From Feedforward to Recurrent LSTM Neural Networks for Language Modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3), 517–529. <https://doi.org/10.1109/TASLP.2015.2400218>.
- 16 Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the Importance of Initialization and Momentum in Deep Learning. In *Proceedings of the International Conference on Machine Learning (ICML)* (pp. 1139–1147). Atlanta, Georgia, USA. <https://doi.org/http://proceedings.mlr.press/v28/sutskever13.html>.

- Swisher, C. B., White, C. R., Mace, B. E., & Dombrowski, K. E. (2015). Diagnostic Accuracy of Electrographic Seizure Detection by Neurophysiologists and Non-Neurophysiologists in the Adult ICU Using a Panel of Quantitative EEG Trends. *Journal of Clinical Neurophysiology*, 32(4), 324–330. <https://doi.org/10.1097/WNP.000000000000144>.
- Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1–9). Boston, Massachusetts, USA: IEEE. <https://doi.org/10.1109/CVPR.2015.7298594>.
- Thodoroff, P., Pineau, J., & Lim, A. (2016). Learning Robust Features using Deep Learning for Automatic Seizure Detection. In *Proceedings of the Machine Learning and Healthcare Conference* (pp. 178–190). Los Angeles, California, USA. <https://doi.org/http://proceedings.mlr.press/v56/Thodoroff16.html>.
- Hinton, G. (2012). RMSPROP: Divide the Gradient by a Running Average of its Recent Magnitude. *Coursera: Neural Networks for Machine Learning*. <https://doi.org/https://www.coursera.org/learn/neural-networks/lecture/YQHki/rmsprop-divide-the-gradient-by-a-running-average-of-its-recent-magnitude>.
- van der Maaten, L., Postma, E., & van den Herik, J. (2009). Dimensionality Reduction: A Comparative Review. *Journal of Machine Learning Research*, 10(February), 1–41. <https://doi.org/10.1080/1350628044000102>.
- Varsavsky, A., & Mareels, I. (2006). Patient un-specific detection of epileptic seizures through changes in variance. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 3747–3750). New York, New York, USA: IEEE. <https://doi.org/10.1109/IEMBS.2006.260452>.
- Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. *Proceedings of the International Conference on Machine Learning (ICML)* (pp. 1096–1103). New York, New York, USA. <https://doi.org/10.1145/1390156.1390294>.
- Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A. (2010). Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion Pierre-Antoine Manzagol. *Journal of Machine Learning Research*, 11, 3371–3408. <https://doi.org/10.1111/1467-8535.00290>.
- von Weltin, E., Ahsan, T., Shah, V., Jamshed, D., Golmohammadi, M., Obeid, I., & Picone, J. (2017). Electroencephalographic Imaging: A Primary Source of Error in Automatic Seizure Detection. I. Obeid & J. Picone (Eds.), *Proceedings of the IEEE Signal Processing in Medicine and Biology Symposium* (pp. 1–5). Philadelphia, Pennsylvania, USA: IEEE. <https://doi.org/10.1109/SPMB.2017.8257018>.
- Wu, Z., & King, S. (2016). Investigating gated recurrent networks for speech synthesis. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1–5). Shanghai, China. <https://doi.org/10.1109/ICASSP.2016.7472657>.
- Xiong, W., Wu, L., Alleva, F., Droppo, J., Huang, X., & Stolcke, A. (2017). The Microsoft 2017 Conversational Speech Recognition System. In *Proceedings of the IEEE International Conference*

- 76** Acoustics, Speech, and Signal Processing (ICASSP) (pp. 1–5). Calgary, Canada. <https://doi.org/10.1109/ICASSP.2018.8461870>.
- 2** Yamada, T., & Meng, E. (2017). **28** *Practical guide for clinical neurophysiologic testing: EEG*. (E. Meng & **2** L. (Online Service), Eds.). Philadelphia, Pennsylvania, USA: Lippincott Williams & Wilkins. <https://shop.lww.com/Practical-Guide-for-Clinical-Neurophysiologic-Testing--EEG/p/9781496383020>.
- 4** Yang, S., López, S., Golmohammadi, M., Obeid, I., & Picone, J. (2016) **4** Semi-automated annotation of signal events in clinical EEG data. In I. Obeid & J. Picone (Eds.), *Proceedings of the IEEE Signal Processing in **48** edicine and Biology Symposium (SPMB)* (pp. 1–5). Philadelphia, Pennsylvania, USA: IEEE. <https://doi.org/10.1109/SPMB.2016.7846855>.
- 98** Zeiler, M. D. (2012). ADADELTA: An Adaptive Learning Rate Method. *ArXiv, abs/1212.5*, 1–6. <https://arxiv.org/abs/1212.5701>.
- 75** Zinkevich, M., Weimer, M., Smola, A., & Li, L. (2010). Parallelized Stochastic Gradient Descent. In *Proceedings of Neural Information Processing Systems* (Vol. 2, pp. 2595–2603). Vancouver, British Columbia, Canada. <https://doi.org/https://papers.nips.cc/paper/4006-parallelized-stochastic-gradient-descent>.

81

LIST OF TABLES

Table 1. An overview of the corpora used to develop the technology described in this chapter: The Temple University Seizure Detection Corpus (TUSZ) and the Duke University Seizure Detection Corpus (DUSZ).

Table 2. Performance of the proposed architectures on TUSZ

Table 3. A comparison of several CNN and LSTM architectures on DUSZ

Table 4. Comparison of optimization algorithms for a CNN/LSTM system on TUSZ

Table 5. A comparison of activation functions for CNN/LSTM on TUSZ

Table 6. A comparison of FA rates on a ³CNN/LSTM system for sensitivity in the range of 30% for different initialization methods

Table 7. A comparison of performance for a CNN/LSTM system for different regularizations

Table 1. An overview of the corpora used to develop the technology described in this chapter: The Temple University Seizure Detection Corpus (TUSZ) and the Duke University Seizure Detection Corpus (DUSZ).

Description	TUSZ		DUSZ
	Train	Eval	Eval
Patients	64	50	45
Sessions	281	229	45
Files	1,028	985	45
Seizure (secs)	17,686	45,649	48,567
Non-Seizure (secs)	596,696	556,033	599,381
Total (secs)	614,382	601,682	647,948

Table 2. Performance of the proposed architectures on TUSZ

System	Sensitivity	Specificity	FA/24 Hrs.
HMM	30.32%	80.07%	244
HMM/SdA	35.35%	73.35%	77
HMM/LSTM	30.05%	80.53%	60
IPCA/LSTM	32.97%	77.57%	73
CNN/MLP	39.09%	76.84%	77
CNN/GRU	30.83%	91.49%	21
ResNet	30.50%	94.24%	13
CNN/LSTM	30.83%	97.10%	6
Channel-Based LSTM	39.46%	95.20%	11

Table 3. A comparison of several CNN and LSTM architectures on DUSZ

System	Data	³ Sensitivity	Specificity	FA/24 Hrs.
CNN/LSTM	TUSZ	30.83%	97.10%	6
CNN/LSTM	DUSZ	33.71%	70.72%	40
Channel-Based LSTM	TUSZ	39.46%	95.20%	11
Channel-Based LSTM	DUSZ	42.32%	86.93%	14

Table 4. Comparison of optimization algorithms for a CNN/LSTM system on TUSZ

System	Sensitivity	Specificity	FA/24 Hrs.
SGD	23.12%	72.24%	44
RMSprop	25.17%	83.39%	23
Adagrad	26.42%	80.42%	31
Adadelta	26.11%	79.14%	33
Adam	30.83%	97.10%	6
Adamax	29.25%	89.64%	18
Nadam	30.27%	92.17%	14
SGD	23.12%	72.24%	44
RMSprop	25.17%	83.39%	23

Table 5. A comparison of activation functions for CNN/LSTM on TUSZ

System	Sensitivity	Specificity	FA/24 Hrs.
Linear	26.46%	88.48%	25
Tanh	26.53%	89.17%	21
Sigmoid	28.63%	90.08%	19
Softsign	30.05%	90.51%	18
ReLU	30.51%	94.74%	11
ELU	30.83%	97.10%	6
Linear	26.46%	88.48%	25
Tanh	26.53%	89.17%	21
Sigmoid	28.63%	90.08%	19

Table 6. A comparison of FA rates on a CNN/LSTM system for sensitivity in the range of 30% for different initialization methods

System	Sensitivity	Specificity	FA/24 Hrs.
Orthogonal	30.8%	96.9%	7
Lecun Uniform	30.3%	96.5%	8
Glorot Uniform	31.0%	94.2%	13
Glorot Normal	29.5%	92.4%	18
Variance Scaling	31.8%	92.1%	19
Lecun Normal	31.8%	92.1%	19
He Normal	31.3%	91.1%	22
Random Uniform	30.2%	90.0%	25
Truncated Normal	31.6%	87.8%	31
He Uniform	29.2%	85.1%	40

Table 7. A comparison of performance for a CNN/LSTM system for different regularizations

³ System	Sensitivity	Specificity	FA/24 Hrs.
L1/L2	30.8%	97.1%	6
Dropout	30.8%	96.9%	7
Gaussian	30.8%	95.8%	9
L2	30.2%	95.6%	10
L1	30.0%	43.7%	276

LIST OF FIGURES

Figure 1. The design cycle for machine learning typically involves an iterative process where everything from data to the evaluation paradigm are tweaked to improve the overall performance and generalization of the system.

Figure 2. Base features are calculated using linear frequency cepstral coefficients based on frame and window durations of 0.1 and 0.2 seconds respectively. A novel differential energy term is added to the feature vector, and then first and second derivatives are computed.

Figure 3. A hybrid architecture based on HMMs that integrates temporal and spatial context for sequential decoding of EEG events is shown. Two levels of postprocessing are used.

Figure 4. A two-dimensional decoding of EEG signals using a CNN/MLP hybrid architecture that consists of six convolutional layers, three max pooling layers and two fully-connected layers is shown.

Figure 5. A deep residual learning framework, ResNet, is shown that consists of 14 layers of convolution followed by a fully connected layer and a sigmoid as the last layer. The deep residual learning architecture mitigates two important problems: vanishing/exploding gradients and saturation of accuracy when the number of layers is increased.

Figure 6. An unsupervised learning architecture is shown that uses deep convolutional generative adversarial networks (DCGANs).

Figure 7. An architecture that integrates IPCA for spatial context analysis and LSTM for learning long-term temporal dependencies

Figure 8. A deep recurrent convolutional architecture for two-dimensional decoding of EEG signals that integrates 2D CNNs, 1-D CNNs and LSTM networks

Figure 9. A hybrid architecture that integrates HMM and LSTM

Figure 10. A channel-based long short-term memory (LSTM)

Figure 11. OVLP scoring is very permissive about the degree of overlap between the reference and hypothesis. For example, in example 1, the TP score is 1 with no false alarms. In example 2, the system detects 2 out of 3 seizure events, so the TP and FN scores are 2 and 1 respectively.

Figure 12. An illustration of the postprocessing algorithms used to reduce the FA rate

Figure 13. A DET curve comparison of the proposed architectures on TUSZ

Figure 14. An expanded comparison of performance in a region where the FP rate is low

Figure 15. A comparison of the performance of CNN/LSTM and channel-based LSTM on TUSZ and DUSZ

Figure 16. A comparison of different initialization methods for CNN/LSTM

Figure 17. A comparison of different regularization methods for CNN/LSTM

Figure 18. Synthetic EEG waveforms generated using DCGAN

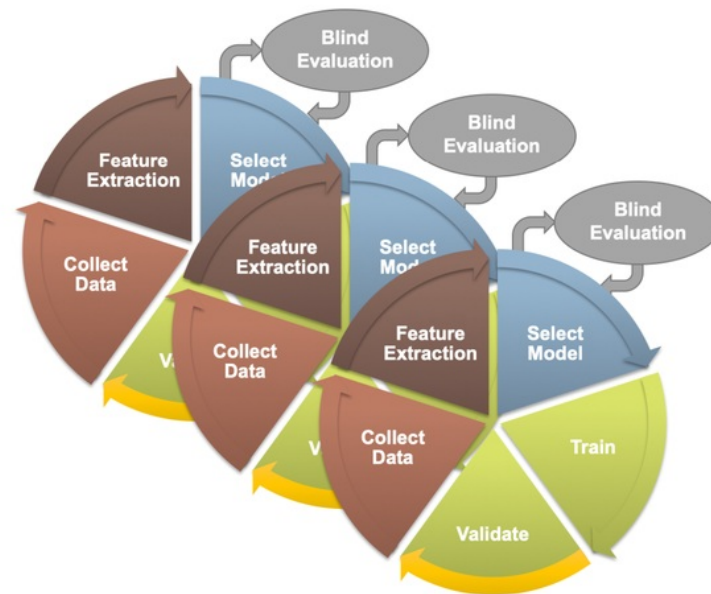
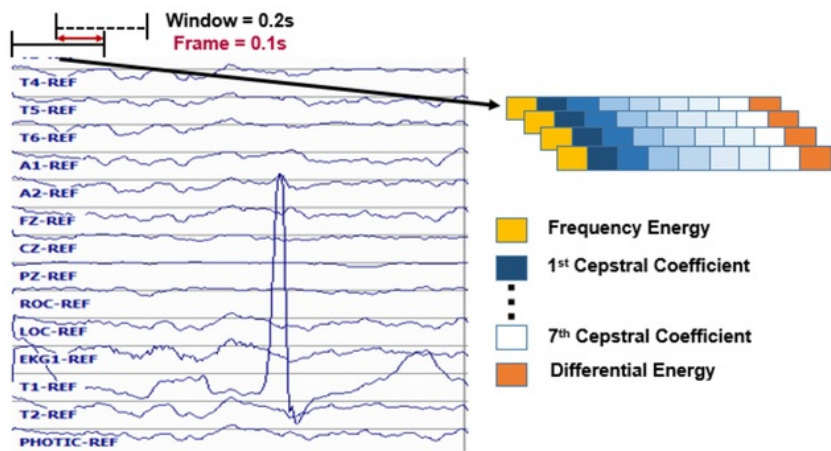


Figure 1. The design cycle for machine learning typically involves an iterative process where everything from data to the evaluation paradigm are tweaked to improve the overall performance and generalization of the system.



93
Figure 2. Base features are calculated using linear frequency cepstral coefficients based 173 frame and window durations of 0.1 and 0.2 seconds respectively. A novel differential energy term is added to the feature vector, and then first and second derivatives are computed. 42

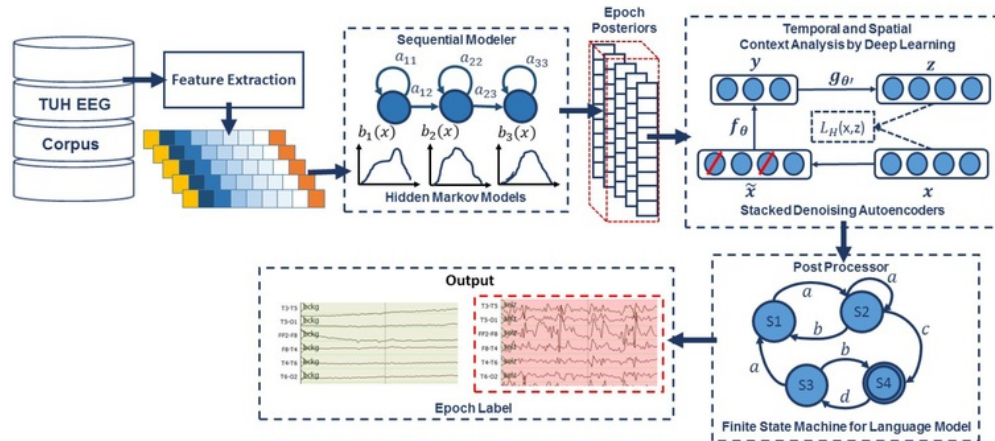


Figure 3. A hybrid architecture based on HMMs ¹that integrates temporal and spatial context for sequential decoding of EEG events is shown. Two levels of postprocessing are used.

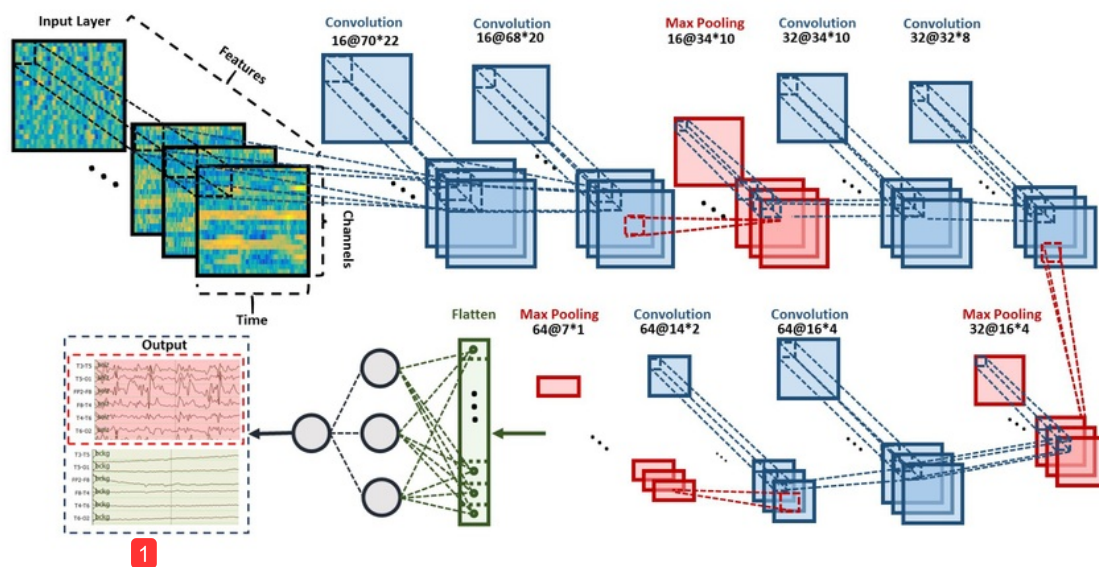


Figure 4. A two-dimensional decoding of EEG signals using a CNN/MLP hybrid architecture that consists of six convolutional layers, three max pooling layers and two fully-connected layers is shown.

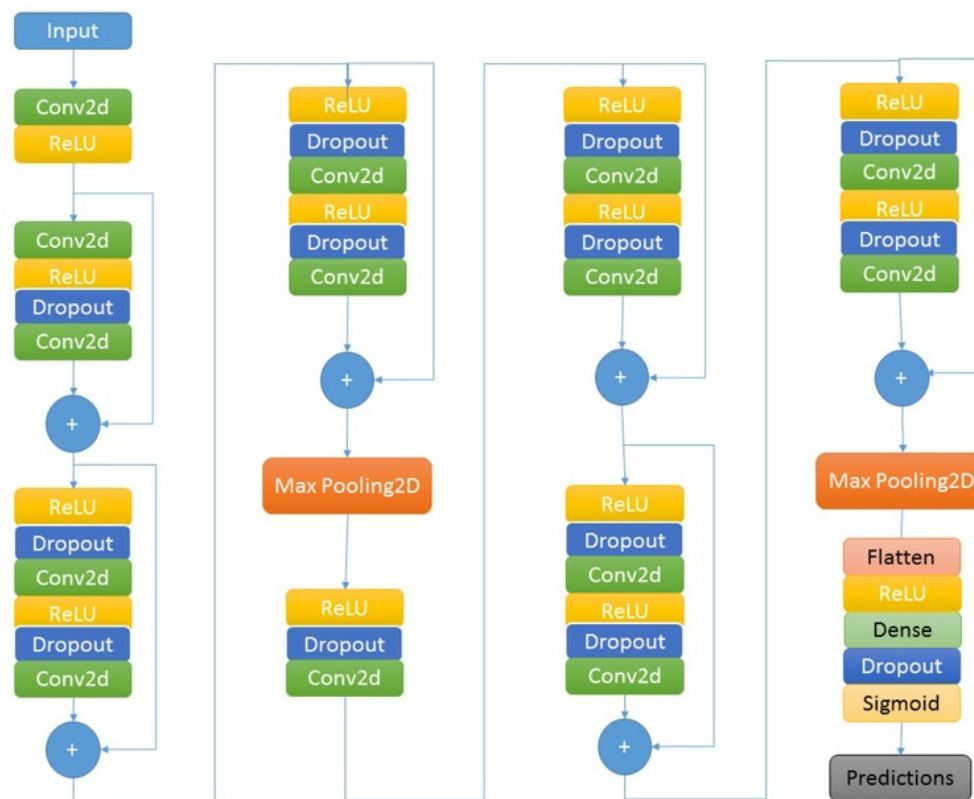


Figure 5. A deep residual learning framework, ResNet, is shown that consists of 14 layers of convolution followed by a fully connected layer and a sigmoid as the last layer. The deep residual learning structure mitigates two important problems: vanishing/exploding gradients and saturation of accuracy when the number of layers is increased.

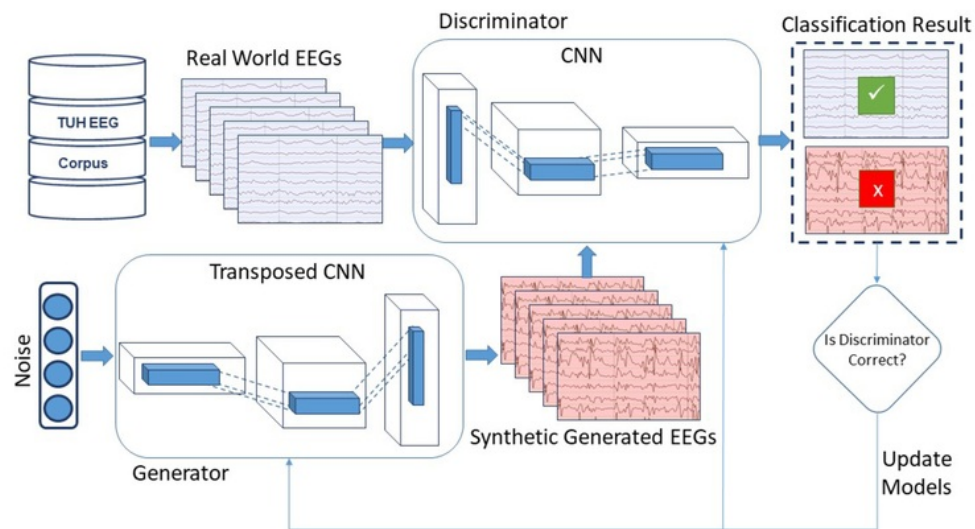


Figure 6. An unsupervised learning architecture is shown that uses deep convolutional generative adversarial networks (DCGANs). GANs have emerged as a powerful learning paradigm technique for learning generative models for high-dimensional unstructured data.

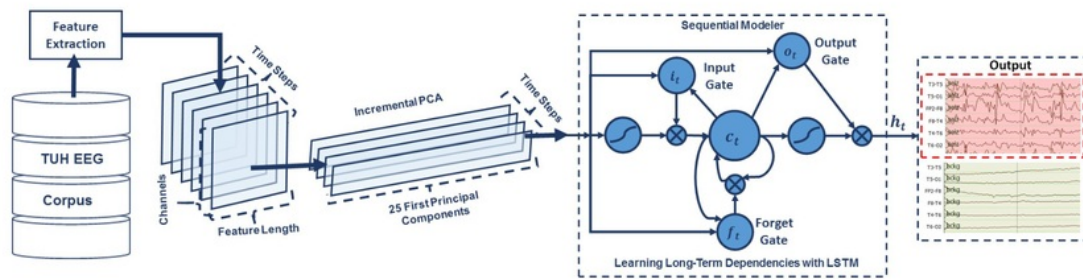


Figure 7. An architecture that integrates IPCA for spatial context analysis and LSTM for learning long-term temporal dependencies

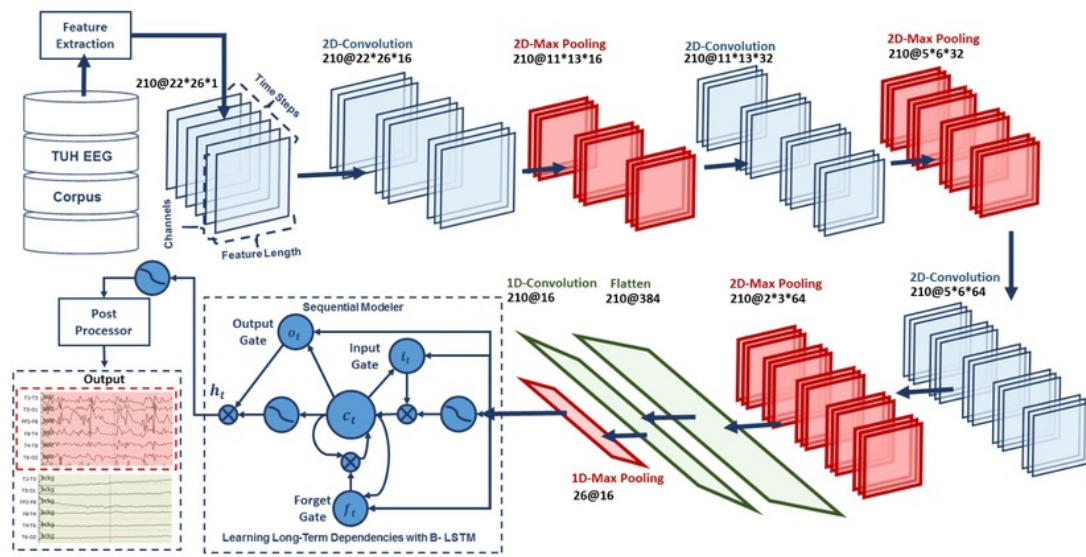


Figure 8. A deep recurrent convolutional architecture for two-dimensional decoding of EEG signals that integrates 2D CNNs, 1-D CNNs and LSTM networks

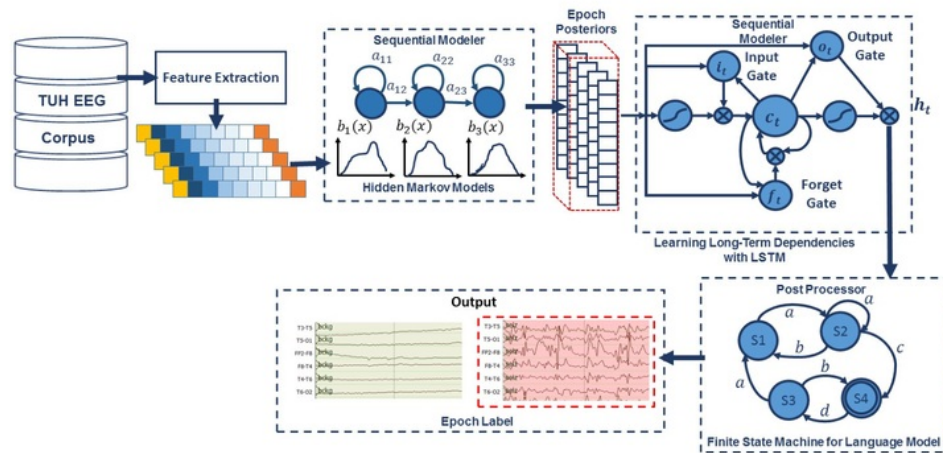


Figure 9. A **hybrid architecture that integrates HMM and LSTM**

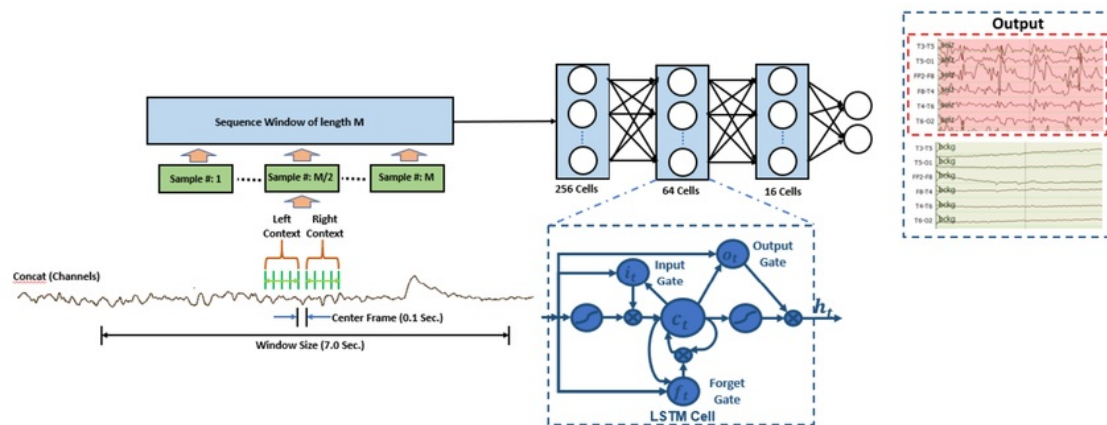


Figure 10. A channel-based long short-term memory (LSTM) architecture

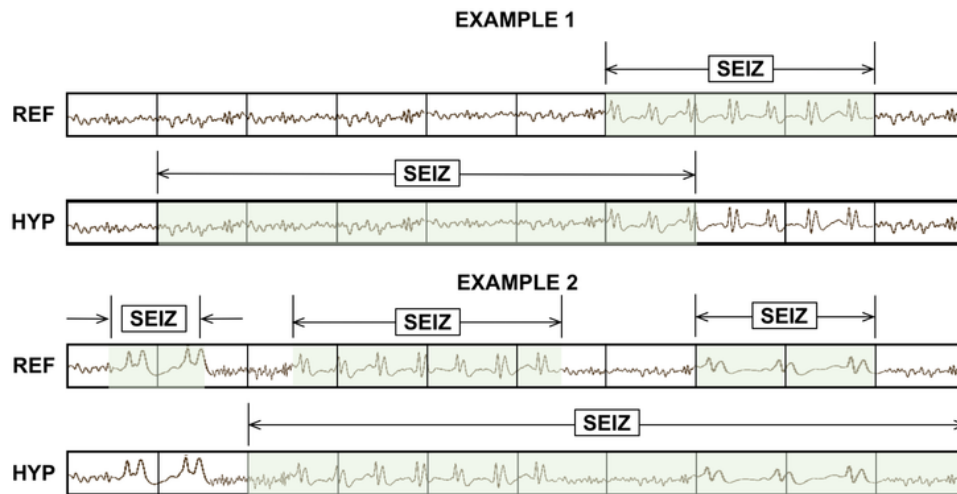


Figure 11. OVLP scoring is very permissive about the degree of overlap between the reference and hypothesis. For example, in example 1, the TP score is 1 with no false alarms. In example 2, the system detects 2 out of 3 seizure events, so the TP and FN scores are 2 and 1 respectively.

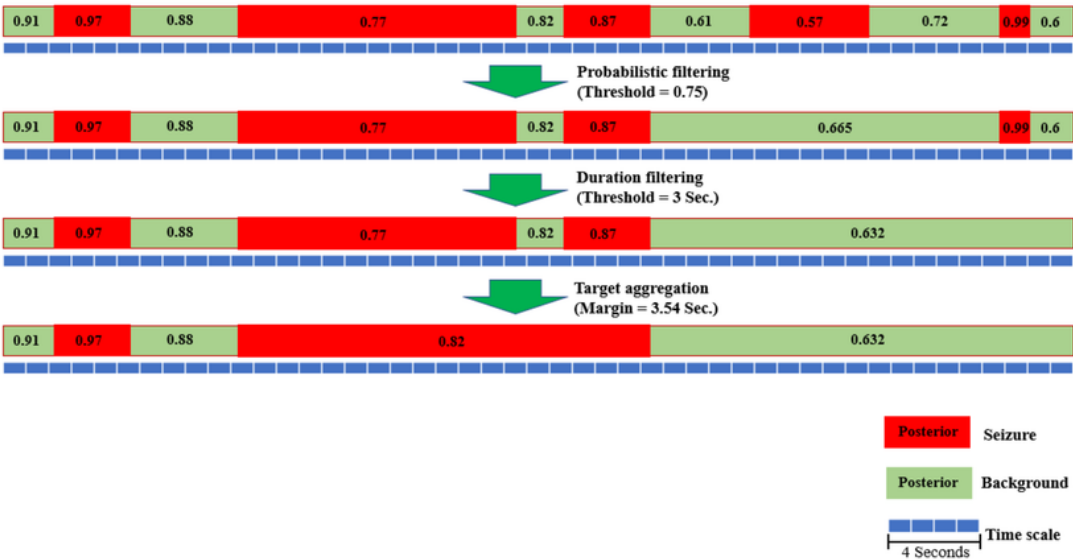


Figure 12. An illustration of the postprocessing algorithms used to reduce the FA rate

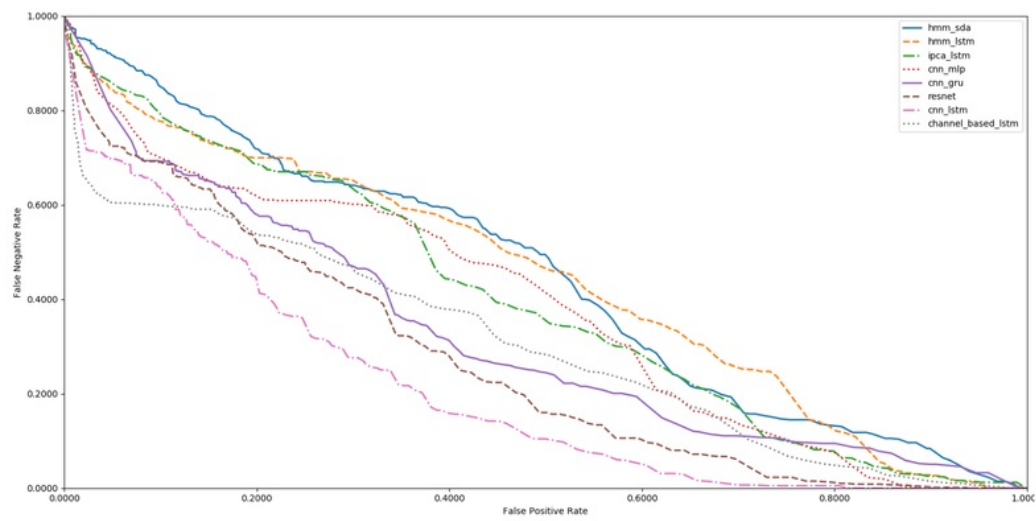


Figure 13. A DET curve comparison of the proposed architectures on TUSZ

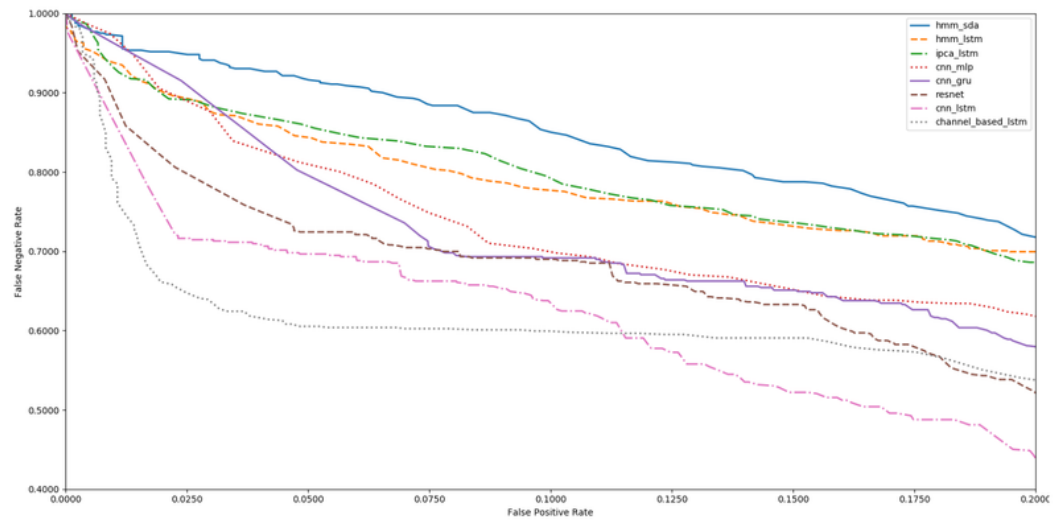


Figure 14. An expanded comparison of performance in a region where the FP rate is low

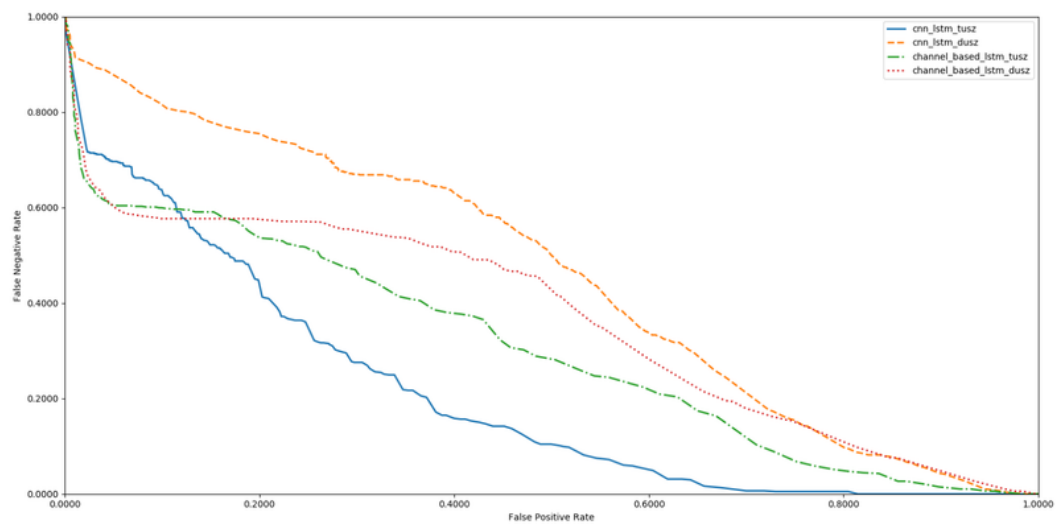


Figure 15. A comparison of the performance of CNN/LSTM and channel-based LSTM on TUSZ and DUSZ

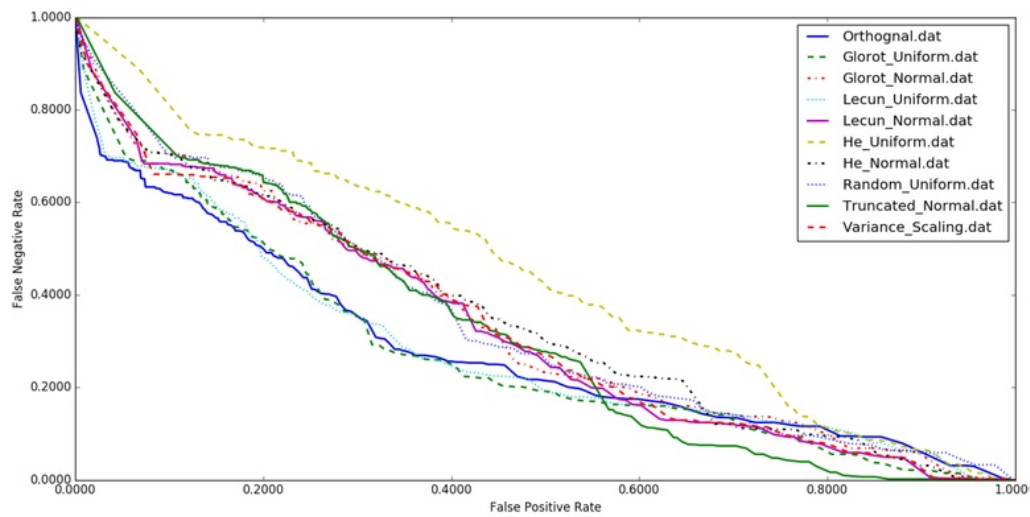


Figure 16. A comparison of different initialization methods for CNN/LSTM

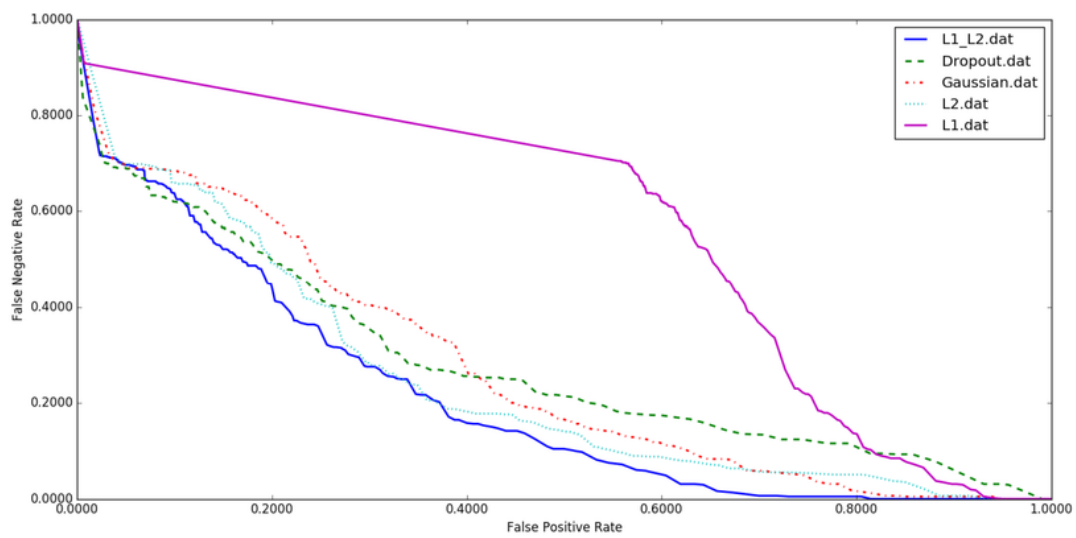


Figure 17. A comparison of different regularization methods for CNN/LSTM

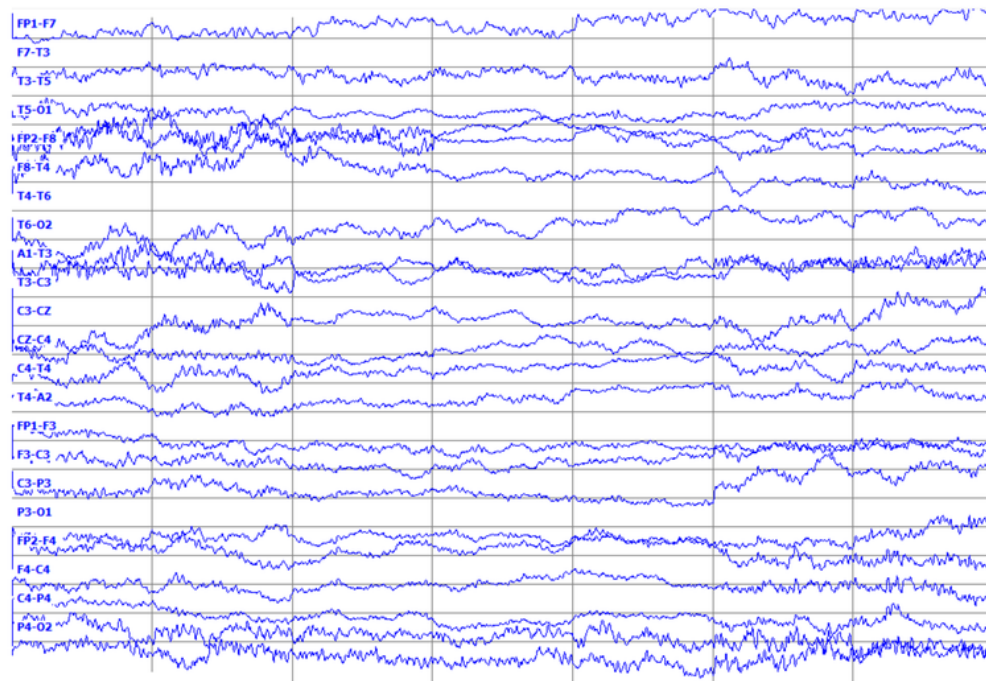


Figure 18. Synthetic EEG waveforms generated using DCGAN

47%

SIMILARITY INDEX

PRIMARY SOURCES

- | | | |
|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| 1 | Meysam Golmohammadi, Saeedeh Ziyabari, Vinit Shah, Iyad Obeid, Joseph Picone. "Deep Architectures for Spatio-Temporal Modeling: Automated Seizure Detection in Scalp EEGs", 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018
<small>Crossref</small> | 2428 words — 12% |
| 2 | www.frontiersin.org
<small>Internet</small> | 1630 words — 8% |
| 3 | M. Golmohammadi, S. Ziyabari, V. Shah, E. Von Weltin, C. Campbell, I. Obeid, J. Picone. "Gated recurrent networks for seizure detection", 2017 IEEE Signal Processing in Medicine and Biology Symposium (SPMB), 2017
<small>Crossref</small> | 828 words — 4% |
| 4 | www.isip.piconepress.com
<small>Internet</small> | 512 words — 3% |
| 5 | link.springer.com
<small>Internet</small> | 174 words — 1% |
| 6 | export.arxiv.org
<small>Internet</small> | 162 words — 1% |
| 7 | docplayer.net
<small>Internet</small> | 100 words — < 1% |
| 8 | file.scirp.org
<small>Internet</small> | 96 words — < 1% |
| 9 | www.aclweb.org
<small>Internet</small> | 81 words — < 1% |

10	tel.archives-ouvertes.fr Internet	80 words — < 1%
11	Matthias Duempelmann. "Early seizure detection for closed loop direct neurostimulation devices in epilepsy", Journal of Neural Engineering, 2019 Crossref	78 words — < 1%
12	www.springerprofessional.de Internet	70 words — < 1%
13	www.mdpi.com Internet	69 words — < 1%
14	ai.vub.ac.be Internet	68 words — < 1%
15	"Computer Vision – ECCV 2016", Springer Nature, 2016 Crossref	59 words — < 1%
16	msb.embopress.org Internet	55 words — < 1%
17	andysarroff.com Internet	52 words — < 1%
18	Meysam Golmohammadi, Amir Hossein Harati Nejad Torbati, Silvia Lopez de Diego, Iyad Obeid, Joseph Picone. "Automatic Analysis of EEGs Using Big Data and Hybrid Deep Learning Architectures", Frontiers in Human Neuroscience, 2019 Crossref	52 words — < 1%
19	www.mitpressjournals.org Internet	48 words — < 1%
20	pdfs.semanticscholar.org Internet	47 words — < 1%
21	webapps.itc.utwente.nl	

	Internet	46 words — < 1%
22	Lecture Notes in Computer Science, 2015. Crossref	46 words — < 1%
23	S. Ferrell, E. von Weltin, I. Obeid, J. Picone. "Open Source Resources to Advance EEG Research", 2018 IEEE Signal Processing in Medicine and Biology Symposium (SPMB), 2018 Crossref	44 words — < 1%
24	journals.plos.org Internet	43 words — < 1%
25	www.deeplearning.net Internet	43 words — < 1%
26	Chaopeng Shen. "A trans-disciplinary review of deep learning research and its relevance for water resources scientists", Water Resources Research, 2018 Crossref	42 words — < 1%
27	www.waset.org Internet	42 words — < 1%
28	V. Shah, M. Golmohammadi, S. Ziyabari, E. Von Weltin, I. Obeid, J. Picone. "Optimizing channel selection for seizure detection", 2017 IEEE Signal Processing in Medicine and Biology Symposium (SPMB), 2017 Crossref	40 words — < 1%
29	m.scirp.org Internet	39 words — < 1%
30	core.ac.uk Internet	38 words — < 1%
31	www.intechopen.com Internet	37 words — < 1%

32	bulletin.temple.edu Internet	37 words — < 1%
33	scikit-learn.org Internet	36 words — < 1%
34	www.tandfonline.com Internet	35 words — < 1%
35	"Computer Vision – ECCV 2018", Springer Nature America, Inc, 2018 Crossref	35 words — < 1%
36	www.univagora.ro Internet	34 words — < 1%
37	isl.anthropomatik.kit.edu Internet	33 words — < 1%
38	hjweide.github.io Internet	33 words — < 1%
39	"Artificial Neural Networks and Machine Learning – ICANN 2018", Springer Nature America, Inc, 2018 Crossref	33 words — < 1%
40	"Computer Vision – ECCV 2018 Workshops", Springer Nature, 2019 Crossref	33 words — < 1%
41	digitalassets.lib.berkeley.edu Internet	33 words — < 1%
42	www.ieeespmb.org Internet	33 words — < 1%
43	arxiv.org Internet	32 words — < 1%
44	N. Capp, E. Krome, I. Obeid, J. Picone. "Facilitating the annotation	

of seizure events through an extensible
visualization tool", 2017 IEEE Signal Processing in
Medicine and Biology Symposium (SPMB), 2017

Crossref

31 words — < 1 %

45 academic-tree.org

Internet

31 words — < 1 %

46 [www.int-arch-photogramm-remote-sens-spatial-
inf-sci.net](http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net)

Internet

31 words — < 1 %

47 doras.dcu.ie

Internet

31 words — < 1 %

48 "Bridging the Semantic Gap in Image and Video
Analysis", Springer Nature, 2018

Crossref

30 words — < 1 %

49 contentdm.exchange.viterbo.edu

Internet

30 words — < 1 %

50 pastel.archives-ouvertes.fr

Internet

30 words — < 1 %

51 towardsdatascience.com

Internet

30 words — < 1 %

52 www.scribd.com

Internet

29 words — < 1 %

53 doc.rero.ch

Internet

28 words — < 1 %

54 www.sastechjournal.com

Internet

28 words — < 1 %

55 Yinda Zhang, Shuhan Yang, Yang Liu, Yexian
Zhang, Bingfeng Han, Fengfeng Zhou. "Integration
of 24 Feature Types to Accurately Detect and Predict Seizures

27 words — < 1 %

56	gershmanlab.webfactional.com Internet	27 words — < 1%
57	www.hydrol-earth-syst-sci.net Internet	27 words — < 1%
58	stanfordhealthcare.org Internet	26 words — < 1%
59	eprints.fri.uni-lj.si Internet	26 words — < 1%
60	aclweb.org Internet	26 words — < 1%
61	www.biorxiv.org Internet	25 words — < 1%
62	journal.frontiersin.org Internet	24 words — < 1%
63	data.snu.ac.kr Internet	24 words — < 1%
64	bfrl.nist.gov Internet	24 words — < 1%
65	eprints.whiterose.ac.uk Internet	23 words — < 1%
66	Klaus Nordhausen, Hannu Oja. "Independent component analysis: A statistical perspective", Wiley Interdisciplinary Reviews: Computational Statistics, 2018 Crossref	23 words — < 1%
67	Tim Albrecht, Gregory Slabaugh, Eduardo Alonso, SM Masudur R Al-Arif. "Deep learning for single-molecule science", Nanotechnology, 2017 Crossref	22 words — < 1%

68	d-nb.info Internet	22 words — < 1%
69	Jiansheng Wu, Qiuming Zhang, Weijian Wu, Tao Pang, Haifeng Hu, Wallace KB Chan, Xiaoyan Ke, Yang Zhang. "WDL-RF: Predicting Bioactivities of Ligand Molecules Acting with G Protein-coupled Receptors by Combining Weighted Deep Learning and Random Forest", Bioinformatics, 2018 Crossref	22 words — < 1%
70	"Neural Information Processing", Springer Nature, 2017 Crossref	22 words — < 1%
71	Lecture Notes in Computer Science, 2016. Crossref	21 words — < 1%
72	mdpi.com Internet	20 words — < 1%
73	Akira Tanaka, Nozomi Hata, Nariaki Tateiwa, Katsuki Fujisawa. "Practical approach to evacuation planning via network flow and deep learning", 2017 IEEE International Conference on Big Data (Big Data), 2017 Crossref	19 words — < 1%
74	S. Yang, S. Lopez, M. Golmohammadi, I. Obeid, J. Picone. "Semi-automated annotation of signal events in clinical EEG data", 2016 IEEE Signal Processing in Medicine and Biology Symposium (SPMB), 2016 Crossref	19 words — < 1%
75	repository.upi.edu Internet	18 words — < 1%
76	"Advances in Multimedia Information Processing – PCM 2018", Springer Nature America, Inc, 2018 Crossref	18 words — < 1%
77	jcheminf.springeropen.com Internet	18 words — < 1%

78	Signals and Communication Technology, 2015. Crossref	17 words — < 1%
79	lib.tkk.fi Internet	17 words — < 1%
80	Özal Yıldırım, Ulas Baran Baloglu, U. Rajendra Acharya. "A deep convolutional neural network model for automated identification of abnormal EEG signals", Neural Computing and Applications, 2018 Crossref	17 words — < 1%
81	jyx.jyu.fi Internet	17 words — < 1%
82	usir.salford.ac.uk Internet	17 words — < 1%
83	Duo Zhang, Geir Lindholm, Harsha Ratnaweera. "Use long short-term memory to enhance Internet of Things for combined sewer overflow monitoring", Journal of Hydrology, 2018 Crossref	17 words — < 1%
84	publications.polymtl.ca Internet	17 words — < 1%
85	Chen Liang, Xiao Yang, Drew Wham, Bart Pursel, Rebecca Passonneau, C. Lee Giles. "Distractor Generation with Generative Adversarial Nets for Automatically Creating Fill-in-the-blank Questions", Proceedings of the Knowledge Capture Conference on - K-CAP 2017, 2017 Crossref	17 words — < 1%
86	www.analyticsindiamag.com Internet	17 words — < 1%
87	hal.archives-ouvertes.fr Internet	17 words — < 1%
88	d-scholarship.pitt.edu Internet	17 words — < 1%

		15 words — < 1%
89	epilepsycurrents.org Internet	15 words — < 1%
90	ttic.uchicago.edu Internet	15 words — < 1%
91	asa.scitation.org Internet	15 words — < 1%
92	en.wikipedia.org Internet	15 words — < 1%
93	S. Lopez, A. Gross, S. Yang, M. Golmohammadi, I. Obeid, J. Picone. "An analysis of two common reference points for EEGS", 2016 IEEE Signal Processing in Medicine and Biology Symposium (SPMB), 2016 Crossref	14 words — < 1%
94	Aatif M. Husain. "Raw versus Processed EEG: Which One is Better?", Epilepsy Currents, 2018 Crossref	14 words — < 1%
95	www.ncbi.nlm.nih.gov Internet	14 words — < 1%
96	"Speech and Computer", Springer Nature, 2017 Crossref	14 words — < 1%
97	mdp-toolkit.sourceforge.net Internet	14 words — < 1%
98	ruder.io Internet	13 words — < 1%
99	scholar.sun.ac.za Internet	13 words — < 1%
100	www.sourcesignal.com Internet	

13 words — < 1%

101 espace.library.uq.edu.au
Internet

13 words — < 1%

102 image-net.org
Internet

13 words — < 1%

103 Pallavi Kaushik, Anmol Gupta, Partha Pratim Roy, Debi Prosad Dogra. "EEG-Based Age and Gender Prediction Using Deep BLSTM-LSTM Network Model", IEEE Sensors Journal, 2019
Crossref

13 words — < 1%

104 www.mysmu.edu
Internet

12 words — < 1%

105 Anselmo Caricato, Isabella Melchionda, Massimo Antonelli. "Continuous Electroencephalography Monitoring in Adults in the Intensive Care Unit", Critical Care, 2018
Crossref

12 words — < 1%

106 eprints.nottingham.ac.uk
Internet

12 words — < 1%

107 www.geosci-model-dev.net
Internet

12 words — < 1%

108 R. Schirrmeister, L. Gemein, K. Eggenberger, F. Hutter, T. Ball. "Deep learning with convolutional neural networks for decoding and visualization of EEG pathology", 2017 IEEE Signal Processing in Medicine and Biology Symposium (SPMB), 2017
Crossref

12 words — < 1%

109 brage.bibsys.no
Internet

12 words — < 1%

110 Anan Banharnsakun. "Towards improving the convolutional neural networks for deep learning

12 words — < 1%

-
- 111 V. Shah, R. Anstotz, I. Obeid, J. Picone. "Adapting an Automatic Speech Recognition System to Event Classification of Electroencephalograms", 2018 IEEE Signal Processing in Medicine and Biology Symposium (SPMB), 2018
11 words — < 1%
Crossref
-
- 112 openreview.net
Internet 11 words — < 1%
-
- 113 med4you.me
Internet 11 words — < 1%
-
- 114 www.locus.ufv.br
Internet 11 words — < 1%
-
- 115 online.uminho.pt
Internet 11 words — < 1%
-
- 116 oa.upm.es
Internet 11 words — < 1%
-
- 117 Syed Umar Amin, M. Shamim Hossain, Ghulam Muhammad, Musaed Alhussein, Md. Abdur Rahman. "Cognitive Smart Healthcare for Pathology Detection and Monitoring", IEEE Access, 2019
11 words — < 1%
Crossref
-
- 118 L. Veloso, J. McHugh, E. von Weltin, S. Lopez, I. Obeid, J. Picone. "Big data resources for EEGs: Enabling deep learning research", 2017 IEEE Signal Processing in Medicine and Biology Symposium (SPMB), 2017
10 words — < 1%
Crossref
-
- 119 Indar Sugiarto, Felix Pasila. "Understanding a Deep Learning Technique through a Neuromorphic System a Case Study with SpiNNaker Neuromorphic Platform", MATEC Web of Conferences, 2018
10 words — < 1%
Crossref
-

120	n.neurology.org Internet	10 words — < 1%
121	dblp2.uni-trier.de Internet	10 words — < 1%
122	www.m-hikari.com Internet	10 words — < 1%
123	pubman.mpg.de Internet	10 words — < 1%
124	repositorio-aberto.up.pt Internet	10 words — < 1%
125	www1.cmc.edu Internet	10 words — < 1%
126	en.d2l.ai Internet	10 words — < 1%
127	lijiancheng0614.github.io Internet	10 words — < 1%
128	publikationen.sulb.uni-saarland.de Internet	10 words — < 1%
129	"Modern Electroencephalographic Assessment Techniques", Springer Nature America, Inc, 2015 Crossref	10 words — < 1%
130	repository.tudelft.nl Internet	10 words — < 1%
131	hal.inria.fr Internet	10 words — < 1%
132	idus.us.es Internet	10 words — < 1%
133	Cheng-Hao Cai, Yanyan Xu, Dengfeng Ke, Kaile Su, Jing Sun.	

"Trainable back-propagated functional transfer matrices", Applied Intelligence, 2018

Crossref

9 words — < 1%

134 nzdementia.org

Internet

9 words — < 1%

135 www.qucosa.de

Internet

9 words — < 1%

136 www.seas.upenn.edu

Internet

9 words — < 1%

137 www.cbica.upenn.edu

Internet

9 words — < 1%

138 josvanderwesthuizen.github.io

Internet

9 words — < 1%

139 citeseerx.ist.psu.edu

Internet

9 words — < 1%

140 atlantic.cps.msu.edu

Internet

9 words — < 1%

141 aaltodoc.aalto.fi

Internet

9 words — < 1%

142 epdf.tips

Internet

9 words — < 1%

143 shareok.org

Internet

9 words — < 1%

144 baptiste-wicht.com

Internet

9 words — < 1%

145 github.com

Internet

9 words — < 1%

146 Yanfei Zhong, Xiaobing Han, Liangpei Zhang.

9 words — < 1%

"Multi-class geospatial object detection based on a position-sensitive balancing framework for high spatial resolution remote sensing imagery", ISPRS Journal of Photogrammetry and Remote Sensing, 2018

Crossref

147	udspace.udel.edu Internet	9 words — < 1%
148	Communications in Computer and Information Science, 2015. Crossref	9 words — < 1%
149	publications.idiap.ch Internet	9 words — < 1%
150	www.ies.uni-kassel.de Internet	8 words — < 1%
151	www.nap.edu Internet	8 words — < 1%
152	lup.lub.lu.se Internet	8 words — < 1%
153	sispress.org Internet	8 words — < 1%
154	cmuchimps.org Internet	8 words — < 1%
155	www.onepetro.org Internet	8 words — < 1%
156	psycho.unibas.ch Internet	8 words — < 1%
157	ro.ecu.edu.au Internet	8 words — < 1%
158	iab-rubric.org Internet	8 words — < 1%

159	digitalcommons.fiu.edu Internet	8 words — < 1%
160	pericles.pericles-prod.literatumonline.com Internet	8 words — < 1%
161	on-demand-gtc.gputechconf.com Internet	8 words — < 1%
162	cora.ucc.ie Internet	8 words — < 1%
163	dspace.cc.tut.fi Internet	8 words — < 1%
164	es.scribd.com Internet	8 words — < 1%
165	hal.univ-lorraine.fr Internet	8 words — < 1%
166	Patricia L. Suárez, Angel D. Sappa, Boris X. Vintimilla. "Chapter 26 Colorizing Infrared Images Through a Triplet Conditional DCGAN Architecture", Springer Nature, 2017 Crossref	8 words — < 1%
167	Lecture Notes in Computer Science, 2013. Crossref	8 words — < 1%
168	www.slicer.org Internet	8 words — < 1%
169	onlinelibrary.wiley.com Internet	8 words — < 1%
170	Radenen, Mathieu, and Thierry Artières. "Handling signal variability with contextual markovian models", Pattern Recognition Letters, 2013. Crossref	7 words — < 1%

171 Eunbi Seo, Hyun Min Song, Huy Kang Kim. "GIDS: GAN based Intrusion Detection System for In-Vehicle Network", 2018 16th Annual Conference on Privacy, Security and Trust (PST), 2018 7 words — < 1%
Crossref

172 Lecture Notes in Computer Science, 2014. 7 words — < 1%
Crossref

173 Guo, L.. "Automatic feature extraction using genetic programming: An application to epileptic EEG classification", Expert Systems With Applications, 201108 6 words — < 1%
Crossref

174 Helal, Alaa Eldeen Mahmoud, Ahmed Farag Seddik, Mohammed Ali Eldosoky, and Ayat Allah Farouk Hussein. "An Efficient Method for Epileptic Seizure Detection in Long-Term EEG Recordings", Journal of Biomedical Science and Engineering, 2014. 6 words — < 1%
Crossref

175 Vinit Shah, Eva von Weltin, Silvia Lopez, James Riley McHugh, Lillian Veloso, Meysam Golmohammadi, Iyad Obeid, Joseph Picone. "The Temple University Hospital Seizure Detection Corpus", Frontiers in Neuroinformatics, 2018 6 words — < 1%
Crossref

176 Xinyu Li, Randall S. Burd, Yanyi Zhang, Jianyu Zhang et al. "Progress Estimation and Phase Detection for Sequential Processes", Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2017 6 words — < 1%
Crossref

177 "Intelligence Science I", Springer Nature, 2017 6 words — < 1%
Crossref

178 "Image Analysis and Processing - ICIAP 2017", Springer Nature, 2017 6 words — < 1%
Crossref

179

Seyedmahdad Mirsamadi, John H.L. Hansen. "Multi-domain adversarial training of neural network acoustic models for distant speech recognition", Speech Communication, 2018

Crossref

6 words — < 1%

EXCLUDE QUOTES OFF

EXCLUDE BIBLIOGRAPHY OFF

EXCLUDE MATCHES OFF