A Review of Hidden Markov Models and Recurrent Neural Networks for Event Detection and Localization in Biomedical Signals

Yassin Khalifa, Danilo Mandic and Ervin Sejdić

ARTICLE INFO

Keywords: Event Detection, Hidden Markov Models, Recurrent Neural Networks, Deep Learning, Biomedical Signal Processing, Transfer Learning

ABSTRACT

Biomedical signals carry signature rhythms of complex physiological processes that control our daily bodily activity. The properties of these rhythms indicate the nature of interaction dynamics among physiological processes that maintain a homeostasis. Abnormalities associated with diseases or disorders usually appear as disruptions in the structure of the rhythms which makes isolating these rhythms and the ability to differentiate between them, indispensable. Computer-aided diagnosis systems are ubiquitous nowadays in almost every medical facility and more closely in wearable technology, and rhythm or event detection is the first of many intelligent steps that they perform. How these rhythms are isolated? How to develop a model that can describe the transition between processes in time? Many methods exist in the literature that address these questions and perform the decoding of biomedical signals into separate rhythms. In here, we demystify the most effective methods that are used for detection and isolation of rhythms or events in time series and highlight the way in which they were applied to different biomedical signals and how they contribute to information fusion. The key strengths and limitations of these methods are also discussed as well as the challenges encountered with application in biomedical signals.

1. Introduction

Physiological processes are complex tasks performed by the different systems of the human body in a rarely periodic but rather irregular manner to deliver an action that could be biochemical, electrical, or mechanical [1, 2]. Some of these actions are obvious like heart beating, breathing, and other physical activities and some are not as obvious like hormonal stimulation that regulates multiple body functions. The action produced can be usually manifested as some sort of a signal that holds information about the parent physiological process [2]. Disruptions in these physiological processes associated with diseases, lead to the development of pathological processes that alter the performance of the human body. Both normal and pathological processes in addition to other artifacts from the environment and surrounding processes, are all held in the manifested signals and the associated changes in their waveform. These signals are called biomedical signals and can be of many forms including the electrical form (potential or current changes) or physical (force or temperature) [2].

Artificial intelligence is currently taking over to empower a variety of assistive technologies that help solve the problems of the healthcare sector given the continuously increasing cost and shortage of professional caregivers. These technologies are advancing to perform not only diagnosis but also intervention and curing due to the superior sensitivity, adaptability, and fast response. Of these assistive technologies, computer-aided diagnosis and wearable systems are powered by the virtual side of artificial intelligence (machine learning techniques) and play a vital role in anomaly detection, monitoring, and even emergency response [3]. The rise of such systems has led to the evolution of biomedical signal analysis which has been the focus of researchers for the last couple of decades. This evolution not only included the macro-analysis of gross processes but also the detection and analysis of micro-events within each gross process [3]. As mentioned before, biomedical signals carry the signatures of many processes and artifacts, which makes the extraction/identification of the specific part of interest (called event or epoch), the first step of any systematic signal analysis or monitoring [4]. Further, the need for robust event extraction algorithms for biomedical signals is driven by the exponential growth of the amount and complexity of data generated by biomedical systems [5]. Moreover, reducing the human-dependent steps in the analysis, mitigates the reliability and subjectivity issues associated with human tolerance.

Epoch extraction is not only essential for systematic signal analysis, but also substantial to information fusion for multi-channel systems and/or sensor networks which represent a large portion of biomedical-signal-based decision-making systems nowadays. Multiple fusion models can employ epoch extraction and event detection to overcome different obstacles including but not limited to signal synchronization and feature fusion [6, 7]. In complementary data-level fusion, events can be used to align signals as preparation for feature extraction such as using heart beats to align the signals from multiple electrocardiography (ECG) leads. In feature-level fusion

*Corresponding author

Email address: esejdic@ieee.org (E. Sejdić)

URL: www.imecbio.org (E. Sejdić)

ORCID(i): 0000-0003-4987-8298 (E. Sejdić)
models, event detection can be used to combine features from different signals during only the events of interest that contribute to morphology analysis and the decision-making process [7, 8].

Epoch extraction algorithms have been used repeatedly in segmentation of many biomedical signals, including, but not limited to, heart sound and ECG [9, 10], electroencephalography (EEG) [11–13], and swallowing vibrations [14–17]. Such algorithms immensely depend on modeling time-series, the paradigm that is not explicitly provided by regular machine learning and sequence-agnostic models such as support vector machines, regression, and feed forward neural networks [18]. These models depend on a major assumption that the training and test examples are independent and not related in time or space which in result initiates a reset to the entire state of the model [18]. Particularly speaking, splitting time series into data chunks and using consecutive chunks independently in building models is unacceptable because even in the case of modeling a time series with iid processes, the underlying processes might be longer than a single chunk which induces dependency between consecutive chunks.

Sliding window approach has been introduced to tackle the problem of dependence between consecutive chunks through using an overlap which guarantees that a part of each chunk will be carried over to the next chunk. Although this might be useful in modeling many processes, it fails to model long range dependencies and requires the optimization of both data chunk and overlap lengths to best represent the target processes. Additionally, using windowing in time domain provokes a sort of distortion to the frequency representation due to the leakage effect and can only be used for modeling fixed-length input/output scenarios [18]. All of this raised the need for models capable of selectively transferring states across time, processing sequences of not necessarily independent elements, and yielding a computational paradigm that can handle variable-length inputs and outputs [19]. It was not that long before the researchers started to bring stochastic-based models [20] and design deep recurrent networks [19] to perfectly fit the event extraction problems and overcome the limitations of regular machine learning methodologies.

Multiple models have been offered for time dependency representation including Hidden Markov models (HMMs) and Recurrent Neural Networks (RNNs). HMMs were introduced as an extension to Markov chains to probabilistically model a sequence of observations based on an unobserved sequence of states [20]. On the other hand, RNNs generalize the feed-forward neural networks with the ability to process sequential data one step at a time while selectively transferring information across sequence elements [18]. Hence, RNNs are successful in modeling sequences with unknown length, components that are not independent, and multi-scale sequential dependencies [19, 21, 22]. Further, RNNs overcame a major HMM limitation in modeling the long-range dependencies within the sequences [18, 23].

In this manuscript, we review the fundamental methods developed for event extraction in biomedical signals and unravel the key differences between these methods based on the state-of-the-art practices and results. We show the theoretical and practical aspects for most of the methods and the way in which they were used to handle the time modeling in event detection problems. Further, we discuss the recent major machine learning applications in biomedical signal processing and the anticipated advances for future implementations.

2. Hidden Markov Models

A time series can be characterized using either deterministic or stochastic models. Deterministic models usually describe the series using some specific properties such as being the sum of sinusoids or exponentials and aim to estimate the values of the parameters contributing to these properties (e.g. amplitude, frequency, and phase of the sinusoids) [20]. On the other hand, statistical models assume that the series can be described through a parametric random process whose parameters can be estimated in a well-defined way [20, 24]. HMMs belong to the category of statistical models and usually are referred to as probabilistic functions of Markov chains in the literature [20, 25].

2.1. Markov Chains

Markov chain is a stochastic process modeled by a finite state machine that can be described at any instance of time to be one of $N$ distinct states. These states can be tags or symbols representing the problem of interest. The machine may stay at the same state or switch to another state at regularly spaced discrete times according to a set of transition probabilities associated with each state [20, 24] and the transition probabilities are assumed to be time independent. The initial state is deemed to be known and the transition probabilities are described using the transition matrix: $A = \{a_{ij}\}$; where $a_{ij}$ is the transition probability from state $S_i$ to state $S_j$ and both $i$, and $j$ can take values from 1 to $N$. The actual state at time $t$ is denoted as $q_t$ and for a full description of the probabilistic model, the current state as well as at least the state previous to it (for a first order Markov chain), need to be specified. The first order Markov chain assumes that the current state depends only on the previous state: $P(q_t = j|q_{t-1} = i, q_{t-2} = k, \ldots) = P(q_t = j|q_{t-1} = i)$. This results in the following properties for the transition probabilities:

$$a_{ij} = P(q_t = j|q_{t-1} = i); \quad i \geq 1, \ j \leq N$$

$$a_{ij} \geq 0$$

$$\sum_{j=1}^{N} a_{ij} = 1$$

The probability of being at state $S_i$ at $t = 1$ is denoted as $\pi_i$, and the initial probability distribution as:

$$\pi_i = P(q_1 = S_i); \quad 1 \leq i \leq N$$

$$\Pi = [\pi_1, \pi_2, \ldots, \pi_N]^T$$
Khalifa et al., 2020 Page 3 of 21

An example of a 4-states Markov chain is shown in Fig. 1. This stochastic process is called the observable Markov model since each state corresponds to a visible (observable) event.

Figure 1: An example of a Markov chain with 4 states, $S_1$ to $S_4$, and selected state transitions. A set of probabilities is associated with each state to indicate how the system undergoes state change from one state to itself or another at regular discrete times.

2.2. Hidden Markov Models

So far, we introduced Markov chains in which each state corresponds to an observable event, however this is insufficient for most of the applications where the states cannot always be observable. Therefore, Markov chain models are extended to HMMs which can be widely used in many applications [20]. HMM is considered a doubly stochastic process with one of them hidden or not observable; states, in this case, are hidden from the observer [20]. An HMM is characterized through the following properties [20, 24]:

1. The number of states, $N$, included in the model. As mentioned before, the states are usually hidden in HMMs but sometimes they have a physical significance.

$$q_t \in \{S_1, S_2, \ldots, S_N\}$$

2. The number of distinct observations, a state can take, $M$.

3. The state transition matrix or distribution $A = \{a_{ij}\}$.

4. The observation probability distribution for each state $B = \{b_j(k)\} = P[v_k \text{ at } t | q_t = S_j]$; where $v_k$ represents an element of the distinct observations that a state can take and $1 \leq j \leq N$, $1 \leq k \leq M$.

5. The initial state distribution $\Pi = \{\pi_i\}$.

When known, the previously mentioned parameters can be used to fully describe the HMM ($\lambda(A, B, \Pi)$) and generate an observation sequence $O = \{O_1, O_2, \ldots, O_T\}$ as in the algorithm shown in Algorithm 1.

For the model to be useful for trending applications, it must address three fundamental problems [26]:

- **Likelihood**: Computing the probability of an observation sequence $O = \{O_1, O_2, \ldots, O_T\}$, given the model ($P(O|\lambda)$).

**Algorithm 1: HMM as observations generator**

1. Set $t = 1$;
2. Choose an initial state $q_1 = S_i$ according to $\Pi$;
3. while $t \leq T$ do
   4. Choose $O_t = v_j$ according to the observation distribution in the current state ($b_j(k)$);
   5. Move from the current state $S_i$ to the new state $q_{t+1} = S_j$ according to $a_{ij}$;
   6. set $t = t + 1$;
7. end
8. Result: $O = \{O_1, O_2, \ldots, O_T\}$

- **Decoding**: Choosing the optimal hidden state sequence $Q = \{q_1, q_2, \ldots, q_T\}$ that best represents a given observation sequence ($O = \{O_1, O_2, \ldots, O_T\}$).

- **Estimation**: Adjusting the model parameters $\lambda(A, B, \Pi)$ to maximize the likelihood of a given sequence of observations $O$.

2.3. Likelihood Problem Solution

In the case of Markov chains, where the states are not hidden, the computation of the likelihood is much easier as it narrows the computational burden to just multiplying the transition probabilities within the underlying state sequence. In HMMs, states are hidden which necessitates including all possible state sequences in computing the joint probability ($N^T$ possible hidden state sequences). A dynamic programming solution called the forward-backward algorithm was created for the likelihood problem with a simple time complexity [20]. The forward-backward algorithm sums the probabilities of all possible state sequences that could be included in generating the target observation sequence. The algorithm considers an efficient way to calculate the probability through defining and inductively computing the forward variable $a(t, i)$ which represents the probability of the partial observation sequence $P(O_1, O_2 \ldots O_t, q_t = S_i | \lambda)$ [20, 27, 28]. The forward algorithm for the likelihood problem is fully described as follows:

**Algorithm 2: The forward algorithm**

1. $O = \{O_1, O_2, \ldots, O_T\}$
2. $S \in \{S_1, S_2, \ldots, S_N\}$
3. Create the forward probability table $a[T, N]$
4. foreach state $S \in \{S_1, S_2, \ldots, S_N\}$ do
   5. $a[1, S] \leftarrow a_{S\pi} \times b_S(O_1)$; // Initialization
7. foreach time step $t \in 2, 3, \ldots, T$ do
   8. foreach state $S \in \{S_1, S_2, \ldots, S_N\}$ do
      9. $a[t, S] \leftarrow \sum_{S_{-t}} a[t-1, S_{-t}] \times a_{S_{-t}} \times b_S(O_t)$; // Induction
11. end
12. $P(O|\lambda(A, B, \Pi)) \leftarrow \sum_{S_N} a[T, S]$; // Termination
13. Result: $P(O|\lambda(A, B, \Pi))$

As a part of the forward-backward algorithm, another
variable is considered that will be of help in the solution of the estimation problem. The variable is called the backward probability table, \( b(t, i) = P(O_{i+1}, O_{i+2}, \ldots, O_T|q_i = S_j, \lambda(A, B, \Pi)) \), which represents the probability of the partial observation sequence that starts one time step after the current observation, given the current state \( S_j \) and the model. The backward probability can be calculated in a similar way as the forward probability (Algorithm 3).

**Algorithm 3: Computing the backward probability**

1. Create the backward probability table \( b[T, N] \);
2. \(
\text{foreach state } S \in \{S_1, S_2, \ldots, S_N\} \) do
3. \( b[t, S] \leftarrow 1; \) // Initialization
4. end
5. \text{foreach time step } t \in T − 1, T − 2, \ldots, 1 \) do
6. \text{foreach state } S \in \{S_1, S_2, \ldots, S_N\} do
7. \( b[t, S] \leftarrow \sum_{S=S_j} b(t+1, S) \times a_{S,S} \times b_S(O_{t+1}); \) // Induction
8. end
9. end
10. Result: \( b[T, N] \)

2.4. Decoding Problem Solution: The Viterbi Algorithm

Finding the optimal hidden states sequence that best represents a sequence of observations is more challenging compared to the likelihood problem. Unlike the likelihood problem, the decoding problem does not have an exact solution unless the model is degenerate, which makes it hard to choose the optimality criterion that judges the state sequence [20]. For example, one may choose states based on the individual likelihood of occurrence which achieves the maximum number of correct states individually but not for the overall computed sequence [20]. Another way to solve the decoding problem can be achieved through running the forward-backward algorithm for all possible hidden state sequences and choose the sequence with the maximum likelihood probability, however this is computationally unfeasible [26].

In the same way as the forward-backward algorithm, the Viterbi algorithm solves the decoding problem using dynamic programming. The algorithm recursively computes the probability of being in a state \( S_j \) at time \( t \) taking in consideration the most probable state sequence (path) \( q_1, q_2, \ldots, q_{t-1} \) that leads to this state. The Viterbi algorithm is shown in Algorithm 4.

**Algorithm 4: The Viterbi algorithm**

1. \( O = \{O_1, O_2, \ldots, O_T\}; \)
2. \( S \in \{S_1, S_2, \ldots, S_N\}; \)
3. Create the best path probability table \( \delta[T, N]; \)
4. Create the state index table (the index of state that by adding to the path, maximizes \( \delta \)) \( \psi[T, N]; \)
5. \text{foreach state } S \in \{S_1, S_2, \ldots, S_N\} \) do
6. \( \delta[1, S] \leftarrow a_S \times b_S(O_1); \) // Initialization
7. \( \psi[1, S] \leftarrow 0; \)
8. end
9. \text{foreach time step } t \in 2, 3, \ldots, T \) do
10. \text{foreach state } S \in \{S_1, S_2, \ldots, S_N\} do
11. \( \delta[t, S] \leftarrow \max_{S=S_j} \delta[t−1, S] \times a_{S,S} \times b_S(O_t); \) // Induction
12. \( \psi[t, S] \leftarrow \arg \max_{S=S_j} \delta[t−1, S] \times a_{S,S} \times b_S(O_t); \)
13. end
14. end
15. \( P^* \leftarrow \max_{S=S_j} \delta[T, S]; \) // Termination
16. \( q^*_T \leftarrow \arg \max_{S=S_j} \delta[T, S]; \)
17. for \( t \in \{T, T−1, T−2, \ldots, 2\} \) do
18. \( q^*_t \leftarrow \psi[t, q^*_t]; \) // Backtracking
19. end
20. Result: The optimal state sequence: \( q^*_1, q^*_2, \ldots, q^*_T \)

\[
Q(\tilde{\lambda}, \lambda) = \sum_{\forall q} P(O_{1:T}, q_{1:T}|\tilde{\lambda}) \log P(O_{1:T}, q_{1:T}|\lambda),
\]

where \( P(O_{1:T}, q_{1:T}|\lambda) = \pi \prod_{t=1}^{T-1} a_{q_t, q_{t+1}} b_{q_{t+1}}(O_{t+1}) \), and \( \tilde{\lambda} \) is the initial model. The iterations are performed based on the calculations by the forward-backward probabilities described previously in the solution of the first two problems, and they go as described in Algorithm 5.

2.5. Model Estimation Problem Solution

The third problem can be formulated as finding HMM’s model parameters \( (A, B, \Pi) \) to maximize the conditional probability of observation sequence, given that model [20]. Such a problem doesn’t have an analytical solution, however, iterative methods can be used to find a local maxima for \( P(O|\lambda) \). Here, we focus on the Baum-Welch algorithm that is based on the expectation-maximization method [29, 30]. The algorithm is based on maximizing Baum’s auxiliary function over the updated model parameters \( \lambda \),
Algorithm 5: The estimation algorithm

1. \( O = \{O_1, O_2, \ldots, O_T\} \);
2. \( S \in \{S_1, S_2, \ldots, S_N\} \);
3. Initialize \( \lambda (A, B, \Pi) \);
4. repeat
   5. Using the forward-backward algorithm and \( \lambda \) calculate \( a(T, N) \) and \( \beta(T, N) \).
   6. Create the probability tables \( \xi[T, N] \) (the probability of being in a state \( S_t \) at time \( t \) and a state \( S_{t+1} \) at time \( t+1 \)) and \( \gamma[T, N] \) (the probability of being in a state \( S_t \) at time \( t \));
   7. for each time step \( t \) \( 2, 3, \ldots, T \) do
     8. for each state \( S \in \{S_1, S_2, \ldots, S_N\} \) do
         9. for each state \( S' \in \{S_1, S_2, \ldots, S_N\} \) do
             10. \( \xi[t, S, S'] \leftarrow a(t, S) a(s, S') \gamma(t+1, S') \);\n                \( \gamma[t, S] \leftarrow \sum_{S'=S}^{S_N} \xi[t, S, S'] \);\n         11. end
     12. end
     13. end
     14. set \( a_S \leftarrow \gamma[1, S] \);
     15. \( \sum_{i=1}^{M} c_{jm} = 1 \) and \( c_{jm} \geq 1 \), where \( 1 \leq j \leq N \) and \( 1 \leq m \leq M \).
     16. \( \sum_{i=1}^{M} \gamma[i, S] \);\n     17. \( \bar{b}_{S}(k) \leftarrow \frac{x \cdot O \cdot \bar{b}_{S} \gamma[i, S]}{\sum_{i=1}^{M} \gamma[i, S]} \);\n     18. end
     19. until Convergence;
20. Result: \( \lambda (A, B, \Pi) \);

for \( \Phi \); however, other non-Gaussian models have been considered as well in many applications [34, 35]. The pdf is guaranteed to be normalized, given that the mixture coefficients satisfy the following stochastic conditions: \( \sum_{m=1}^{M} \left( x \cdot O \cdot \bar{b}_{S} \gamma[i, S] \right) \);\n
2.7. State Duration in HMM

One of the convenient ways to include state duration in HMMs, especially with physical signals, is through explicitly modeling the duration density and setting the self-transition coefficients into zeros [20]. The transition from a state to another only occurs after a certain number of observations, specified by duration density, is made in the current state as shown in Fig. 2. In normal HMMs, the states have exponential duration densities that depend on the self transition coefficients \( a_{ij} \) and \( a_{jj} \) as in Fig. 2(a). In HMMs where state duration is modeled by explicit duration densities, there is no self transition and the transition happens only after a specific number of observations determined by the duration density as in Fig. 2(b). The re-estimation formulae needed for model estimation can be defined through including the state duration in the calculation of forward and backward variables. The re-estimation formulae can be found in detail in the tutorial of Rabiner [20].

3. Recurrent Neural Networks

Neural networks are biologically-inspired computational models that are composed of a set of artificial neurons (nodes) joined with directed weighted edges which recently became popular as pattern classifiers [18, 36]. The network is usually activated by feeding an input that then spreads throughout the network along the edges. Many types of neural networks have evolved since its first appearance; however, they will fall under two main categories, the networks whose connections form cycles and the ones that are acyclic [36]. RNNs are the type of neural networks that introduces the notion of time by using cyclic edges between adjacent steps. RNNs have been proposed in many forms including Elman networks, Jordan networks, and echo state networks [37–40].

Figure 2: An illustration of interstate connections in HMMs. (a) represents a normal HMM with self transitions from each state back to itself. (b) represents a variable duration HMM with no self state transition and specified state duration densities.

Figure 3: A simple RNN with a single hidden layer. At each time step \( t \), output is produced through passing activations as in a feedforward network. Activations are passed to next node at time \( t+1 \) as well to achieve recurrence.

As shown in Fig. 3, the hidden units at time \( t \) receive input from the current input \( x_t \) and the previous hidden unit value \( h_{t-1} \). The output \( y_t \) is calculated using the current hidden unit value \( h_t \). Time dependency is created between time steps
by means of recurrent connections between hidden units. In a forward pass, all the computations are specified using the following two equations:

\[ h_t = \sigma_h (W_x x_t + W_h h_{t-1} + b_h), \]
\[ y_t = \sigma_y (W_y h_t + b_y); \]

where \( W_x \) and \( W_y \) represent the matrices of weights between the hidden units and both input and output respectively and \( W_h \) is the matrix of weights between adjacent time steps. \( b_h \) and \( b_y \) are bias vectors which allow offset learning at each node. Nonlinearity is introduced through the activation functions \( \sigma_h \) and \( \sigma_y \) which can be hyperbolic tangent function (tanh), sigmoid, or rectified linear unit (ReLU). In a simple RNN unit, tanh is usually used.

### 3.2. Training of RNNs

The expression of a generic RNN can be represented as

\[ h_t = F(h_{t-1}, x_t, \theta) = W_h \sigma_h(h_{t-1}) + W_x x_t + b_h, \]

where \( \theta \) refers to the network parameters \( W_h \): recurrent weight matrix, \( W_x \): input weight matrix, and \( b_h \): the bias. Initial state \( h_0 \), is usually set to zero, provided by user, or learned. Network performance on a certain task is measured through a cost function \( \epsilon = \sum_{1 \leq t \leq T} \epsilon_t \), where \( \epsilon_t = L(h_t) \), \( T \) is the sequence length (total number of time steps), and \( L \) is the cost operator that measures the performance of the network (e.g. squared error and entropy). Necessary gradients for optimization can be computed using backpropagation through time (BPTT), where the network is unrolled in time so that the application of backpropagation is feasible as shown in Fig. 5.

![Figure 4: Early designs of RNNs](image)

(a) Jordan network. Output units are connected to context units that provide feedback at next time step to hidden units and themselves. (b) Elman network. Hidden units are connected to the context units that provide feedback to the hidden units only at the next time step.

![Figure 5: Unfolded recurrent neural network in time](image)

\( \epsilon_t \) denotes the error calculated from the output, \( h_t \) represents the hidden state, and \( x_t \) represents the input at time \( t \).

A gradient component \( \frac{\partial \epsilon_t}{\partial \theta} \) is calculated through the summation of temporal components as follows:

\[
\frac{\partial \epsilon_t}{\partial \theta} = \sum_{1 \leq t \leq T} \frac{\partial \epsilon_t}{\partial \theta} = \sum_{1 \leq k \leq T} \left( \frac{\partial \epsilon_t}{\partial h_t} \times \frac{\partial h_t}{\partial h_k} \times \frac{\partial h_k}{\partial \theta} \right)
\]

\[
\frac{\partial h_t}{\partial h_k} = \prod_{i \geq k} \frac{\partial h_i}{\partial h_{i-1}} = \prod_{i \geq k} W_h^T \text{diag} (\sigma_h'(h_{i-1})) \tag{1}
\]

The effect that the network parameters (\( \theta \)) at step \( k \) have over the cost at subsequent steps (\( t > k \)), can be measured through the temporal gradient component \( \frac{\partial h_t}{\partial h_k} \times \frac{\partial h_k}{\partial \theta} \). In Eq. 1, the matrix factors are in the form of a product of \( t - k \) Jacobian matrices which will either explode or shrink to zero depending on whether the recurrent weights are greater or smaller than one [42]. The vanishing gradient is common when using sigmoid activations, while the exploding gradient

---

[1] Khalifa et al., 2020
is more common when using rectified linear unit activations [18, 42]. Enforcing the weights through regularization to values that help avoid gradient vanishing and exploding, is one of the solutions to such a problem. Truncated backpropagation through time (TBPTT) is also used as another solution for exploding gradient through setting a maximum number of time steps through which the error is propagated [18].

3.3. Current RNN Designs

Although early designs of RNNs helped to map input into output sequences through using contextual information, this contextual mapping had limited range and the influence of input on hidden layers and thus output, either vanishes or blows up due to cycling through the network recurrent connections as described previously [43]. Gradient vanishing/exploding problem has led to the emergence of new network designs that improve convergence [44, 45]. Of these designs, LSTM, gated recurrent units (GRU), and bidirectional RNNs (BRNN) have proved superiority in long-range contextual mappings and employing both future and past contexts to determine the output of the network [18]. Both LSTM and GRU resemble a standard RNN but with each hidden node replaced by a complete cell as shown in Fig. 6. They also employ a unity-weighted recurrent edge to ensure the transfer of gradient across time steps without decaying or exploding. LSTM forms the long-term memory through the weights which change slowly during training. On the other hand, short term memory is formed by transient activations that pass between successive node [18]. GRU is an LSTM alternative that has a simpler structure and is faster to train; however, it still provides comparable performance to LSTM [46].

In an LSTM unit, a forget gate is an adaptive gate whose output is squashed through a sigmoid activation in order to reset the memory blocks once they are out of date and prevent information storage for arbitrary time lags [47]. The input gate is a sigmoid activated gate whose function is to regulate the new information to be written to the cell state. The output gate is also a sigmoid activated gate that regulates the internal state after being dynamically customized through a tanh activation to be forwarded as the unit output. In the same way, the GRU unit has a similar design; however, it doesn’t have an output gate. It has a reset gate that works as a forget gate and an update gate to regulate the write operation into the unit output from both the state of the past time step and the input from the current time step.

On the other hand, BRNNs resemble a standard RNN architecture as well but with two hidden layers instead of one and each hidden layer is connected to both input and output. One hidden layer passes activations in the forward directions (from the past time steps) and the other layer passes the activations in the backward direction (from future time steps). BRNN is in fact a wiring method for RNN hidden layers regardless of the type of the nodes, which makes it compatible with most RNN architectures including LSTM and GRU [18, 44].

4. Critical Differences between HMMs and RNNs

As demonstrated in the previous sections, construction of hidden Markov models relies on a representing state space from which states are drawn. Scaling such system has long been considered to be difficult or infeasible even with the presence of dynamic programming solutions such as the Viterbi algorithm due to the quadratic complexity nature of the inference problem and transition probability matrix which causes the model parameter estimation and inference to scale in time as the size of the state space grows [48]. Modeling long range dependencies also is impractical in HMMs as transitions occur from a state to the following with no memory of the previous state unless a new space is created with all possible cross-transitions at each time window which leads to exponential growth of the state space size [18, 23]. On the other hand, the number of states that can be represented by a hidden layer in RNNs increases exponentially with the number of nodes in the layer leading to nodes that can carry
information from contexts of arbitrary lengths. Moreover, despite of the exponential growth of the expressive power of the network, training and inference complexities only grow quadratically at most [18]. From a theoretical point of view, RNNs can be efficient in the perception of long contexts; however, this comes at the cost of error propagation. Highly sampled inputs as in the case of raw waveforms, can lead to elongation of the range through which the error signal propagates, thus making the network hard to optimize and reducing the efficiency of computational acceleration tools such as GPUs [49, 50].

5. Event Detection in Electroencephalography

EEG is mostly a non-invasive technique to measure the electrical activity of the brain through a set of electrodes placed on the subject’s scalp. EEG exhibits highly non-stationary behavior and significant non-linear dynamics [77]. The excitatory and inhibitory postsynaptic potentials of the cortical nerve cells are considered the main source of EEG signals [78]. EEG can be invasive if acquired using subdural electrode grids or using depth electrodes and is called intracranial EEG (iEEG); however, typical EEG signals are recorded from scalp locations specified by the 10-20 electrode placement criterion designed by the International Federation of Societies for Electroencephalography and have an amplitude of 10-100 μV and a frequency range of 1-100 Hz [77, 78]. EEG signals are used in the diagnosis of multiple neurological disorders including epilepsy, lesions, tumors, and depression and their characteristics depend strongly on the age and state of the subject. There are multiple events that influence EEG and require the tedious job of analyzing hours of recordings to be extracted. These events range from the diagnosis/detection of certain seizures and syndromes to the tasks of brain computer interface (BCI). These events include the different sleep stages and sleep disorders, epileptic seizures, the effect of music or other artifacts, and the motor imagery tasks.

6. Event Detection in Electroencephalography

EEG is mostly a non-invasive technique to measure the electrical activity of the brain through a set of electrodes placed on the subject’s scalp. EEG exhibits highly non-stationary behavior and significant non-linear dynamics [77]. The excitatory and inhibitory postsynaptic potentials of the cortical nerve cells are considered the main source of EEG signals [78]. EEG can be invasive if acquired using subdural electrode grids or using depth electrodes and is called intracranial EEG (iEEG); however, typical EEG signals are recorded from scalp locations specified by the 10-20 electrode placement criterion designed by the International Federation of Societies for Electroencephalography and have an amplitude of 10-100 μV and a frequency range of 1-100 Hz [77, 78]. EEG signals are used in the diagnosis of multiple neurological disorders including epilepsy, lesions, tumors, and depression and their characteristics depend strongly on the age and state of the subject. There are multiple events that influence EEG and require the tedious job of analyzing hours of recordings to be extracted. These events range from the diagnosis/detection of certain seizures and syndromes to the tasks of brain computer interface (BCI). These events include the different sleep stages and sleep disorders, epileptic seizures, the effect of music or other artifacts, and the motor imagery tasks.

6.1. Sleep Staging in EEG

Sleep is an essential part of the human life cycle and plays a vital role in maintaining most of the body functionality [99]. Sleep disorders include problems with initiating sleep, insomnia, and sleep apnea syndrome (SAS) [100]. Diagnosis of sleep disorders can be done through identifying sleep stages in an overnight polysomnogram (PSG) which utilizes EEG as one of its sensing modalities [101]. Visual scoring of the PSG components is the basic way to categorize sleep epochs and as any manual rating, it suffers from subjectivity and inter-rater tolerance. Many attempts have been proposed in the literature to remedy the problems of expert-based visual scoring of the different components of PSG. The attempts employed multiple algorithms to achieve automatic sleep staging including Markov models and neural networks. Here, we list the recent publications (Table 2) for sleep staging and the detailed description of the methods used within the scope of our review.

6.2. Epilepsy detection in EEG

Epilepsy is one of the episodic disorders of the brain that is characterized by recurrent seizures, unjustified by any known immediate cause [102, 103]. Epileptic seizure is the clinical manifestation that results from the abnormal excessive discharge of some set of neurons in the brain [102]. The seizure consists of transient abnormal alterations of sensory, motor, consciousness, or psychic behavior [102, 103]. Around 80% of the epileptic seizures can be effectively treated if early discovered [104]. Although seizure activity can be easily distinguished in EEG as transient spikes and relatively quiescent periods, it is a time-consuming process and needs clinicians to devote a tremendous amount of time going through hours and days of EEG activity [105]. An efficient and reliable seizure prediction/detection method can be of a great help for the diagnosis, treatment, and even early warning for patients to stop activities that might be of a significant danger during an episode like driving. Several methods have been proposed for seizure prediction, at which EEG signal features are temporally analyzed and compared to heuristic thresholds to trigger a warning for seizures; however, these methods lack generalization when investigated on extensive datasets [106–111]. This can be referred to using feature sets that are not highly affected by the transition from seizure-free to peri-ictal or seizure states or simply the effect cannot be tracked using low-order statistics [106]. Therefore, stochastic-based models, multivariate analysis, and long-range analysis methods were investigated to provide better performance and generalization for EEG-based epileptic seizure prediction. In Table 3, we review the recent publications that use HMMs and RNNs for seizure prediction.

6.3. BCI Tasks in EEG

Motor imagery alters the neural activity of the brain’s sensorimotor cortex in a way that is as observable as if the
transient patterns in EEG signals during the different motor movement was really executed [132].

Identification of the arrhythmia type was performed to combine spectrograms of multi-lead ECG before processing. The authors fed the RNN with the current ECG beat and the T wave part from the former beat to automatically learn the underlying features. This work employed the log spectrogram as an input to the LSTM-based CRNN to increase the accuracy. No multi-channel fusion was performed in this study and only a single ECG lead was employed.

| Publication            | Event under investigation                                                                 | Implementation details                                                                                                                                                                                                 | Dataset                |
|------------------------|--------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| Gersh et al. [51], 1975 | Premature Ventricular Contraction (PVC) through R-R intervals                               | A three states Markov chain was used to model R-R interval (quantized as short, regular, or long) sequences and then the model is used to characterize rhythm through the probability that the observed R-R symbol sequence is generated by a set of models generated from multiple cardiac arrhythmias. The manuscript used a maximum likelihood approach to determine the arrhythmia type. | Clinical test data from patients with atrial fibrillation (AF)       |
| Coe et al. [52], 1990   | Beat detection for arrhythmia analysis                                                      | A parallel combination of HMMs exists for each arrhythmia type that is used to classify arrhythmia. The classification process is inferred through determining the most likely path through the parallel models. All EEG waveform parts were included in the states of each model. The results reported in this study relied on single ECG channel and didn’t include multi-channel ECG fusion. | The American Heart Association (AHA) - ventricular arrhythmia database [25] |
| Andreao et al. [54], 2006 | ECG beat detection and segmentation                                                         | An Elman-based RNN was used for beat classification with the Loglinear-Markov algorithm for training (a k-Nearest Neighbor estimation algorithm based on the maximum neighborhood idea). This model used power spectral density (calculated with three different methods: Parvano, MISHC, and Minimum-Norm) of ECG signals as input. All the models trained in this study used feature-level fusion. | QY database [56]        |
| Sandberg et al. [55], 2005 | Atrial fibrillation frequency tracing                                                      | An LSTM-based RNN followed by a density-based clustering was used for training data selection from a large data pool. In this implementation, the authors fed the RNN with the current ECG gesture and the T wave part from the former beat to automatically learn the underlying features. The RNN layers were followed by two fully connected layers in order to combine the temporal feature and generate the desired output. This study only used a single channel ECG (limb lead II) with no multi-channel fusion. | Simulated atrial fibrillation signals with four different frequency trends: constant frequency, varying frequency, gradually decreasing frequency, and stepwise decreasing frequency. |
| Oliveira et al. [57], 2017 | Automatic augmentation (beat) of ECG and Phonocardiogram (PCG)                           | An LSTM-based RNN was constructed to consider the high dynamics and non-stationary of the signals where the channels were assumed to be co-dependent through past states and observations. Each of ECG and phonocardiogram was modeled with 4 states. This study introduced a decision-level fusion through combining two channels in a single HMM. The study also experimented two different coupled HMMs, a fully connected where transition can happen between any two states from both channels and a partially connected model where certain limitations were added over transitions through considering the prior knowledge of the relationship between heart sounds and ECG components. | Four types of ECG beats obtained from Physiobank Database [59]        |
| Obyl [58], 2009        | Atrial fibrillation detection/identification                                               | An Elman-based RNN was used for beat classification with the Loglinear-Markov algorithm for training (a k-Nearest Neighbor estimation algorithm based on the maximum neighborhood idea). This model used power spectral density (calculated with three different methods: Parvano, MISHC, and Minimum-Norm) of ECG signals as input. All the models trained in this study used feature-level fusion. | MIT-BIH Atrial fibrillation database (MITDB) [61]                    |
| Zhang et al. [59], 2017 | Atrial fibrillation automatic detection                                                     | A layer RNN was used to extract the temporal features from the raw ECG signals. No multi-channel fusion was performed in this study and only a single ECG channel was employed. | The 2017 PhysioNet/CinC Challenge dataset [10]                         |
| Xiang et al. [60], 2017 | Atrial fibrillation automatic detection                                                     | A different cardiac arrhythmia classification was performed in this study and only a single ECG channel was employed. | The 2017 PhysioNet/CinC Challenge dataset [10]                         |
| Schibbi et al. [61], 2017 | Automatic augmentation (beat) of ECG and Phonocardiogram (PCG)                           | A single layer LSTM-based convolutional RNN (CLNN) was constructed for atrial fibrillation detection in arbitrary length ECG recordings. This work employed the log spectrogram as an input to the CLNN to increase the accuracy. No multi-channel fusion was performed in this study and only a single ECG lead was employed. | The 2017 PhysioNet/CinC Challenge dataset [10]                         |
| Eshelman et al. [62], 2017 | Atrial fibrillation detection                                                             | A two layer LSTM-based CLNN was used for atrial fibrillation detection from single-lead ECG and heart rate. Feature-level fusion was performed after the convolutional neural network (CNN) layers to combine features from both inputs. The output from the RNN was used to either feed a dense layer to perform classification directly or train an SVM for classification and the results from both models were combined. | The 2017 PhysioNet/CinC Challenge dataset [10]                         |
| Chang et al. [63], 2018 | Atrial fibrillation detection                                                              | A single layer LSTM-based CLNN was constructed for atrial fibrillation detection in multi-lead ECG. This model also used spectrograms of the input ECG signals to feed the network. Feature-level fusion was performed to combine features of multi-lead ECG before feeding into the LSTM units. | Multiple databases for atrial fibrillation and normal sinus rhythm [64, 65, 67-70]. |
| Liu and Chow [73], 2018 | Myocardial infarction classification                                                       | A deep single-layer LSTM-based CNN was used for classifying ECG beats from single-lead ECG. Multiple models were performed including a direct 4-class beat classifier from the LSTM CRNN via dense layers and 4-class beat classifier via the fusion of multiple one-versus-ones binary classification networks using stacking. | The Physikalisch-Technische Bundesanstalt (PTB) diagnostic ECG database [70] and the AF classification from a short single lead ECG recording: Physionet/computing in cardiology challenge 2017 database (AF-Challenge) [72]. |
| Singh et al. [74], 2018 | Arrhythmia detection                                                                      | 3 models were built for arrhythmia detection, each of them is based on a different type of RNN. Regular RNNs, GRU, and LSTM were used for each of the three models. Each model included 3 layers of different unit size with a dense layer to generate a classification output (normal/abnormal). No multi-channel fusion was performed in this study and only a single ECG lead (ML2) was employed. | MIT-BIH Arrhythmia databases (MITDB) [61]                              |

movement was really executed [132]. Identification of the transient patterns in EEG signals during the different motor imagery tasks like imagining the movement of one of the limbs, is recognized among the most promising and widely used techniques of BCI [133–136]. This is referred to the relatively low cost of the systems used and the high temporal resolution [135]. This type of BCI is called asynchronous BCI because the subject is free to invoke specific thought [132]. On the other hand, synchronous BCI includes the generation of specific mental states in response to external
stimuli [132]. EEG analysis for BCI applications includes the processing of EEG oscillatory activity and the different shifts in its sub-bands in addition to the event-related potentials like VEP and P300 [132, 137]. Many modeling approaches have been introduced to solve the problem of multi-class BCI problem; however, most of them process EEG signals in short windows where stationarity is assumed, which limits the modeling process and excludes the dynamic EEG patterns such as desynchronization [132]. To overcome such a limitation, probabilistic models like HMMs and models capable of representing long range dependencies have been proposed into the implementation of BCI systems. As follows in Table 4, we list the recent work the relies on HMMs and RNNs in BCI systems and uses EEG as the source signal.
Table 3 Summary of EEG-based seizure prediction.

| Publication                      | Event under investigation                                                                 | Implementation details                                                                 | Dataset                                                                 |
|----------------------------------|-------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|------------------------------------------------------------------------|
| Wong et al. [112], 2007          | Evaluation framework for seizure prediction in iEEG                                        | A three states HMM (baseline, detected, and seizure) was constructed to evaluate the prediction algorithms of epileptic seizures. The prediction algorithm is used to generate a binary sequence which is combined with the ground truth (binary detector outputs plus gold-standard human seizure markings) and converted into a binary observation sequence. The trinary vector is used to train the HMM using Baum-Welch, which is then used to Viterbi decode the observation sequence into the hidden state sequence. A hypothesis test that a statistical association exists between the detected and seizure states is performed by counting the transitions from detected state into seizure states in the HMM output. | iEEG data collected from patients diagnosed with mesial temporal lobe epilepsy using 20-39 surgically implanted electrodes on the brain or brain substance [113]. |
| Santaniello et al. [109], 2011    | Early detection of seizures in iEEG from a ret model                                      | Multichannel iEEG were used and Welch’s cross power spectral density was calculated over windows of 3 sec for each pair of channels which were used as input for the detection model. A two state HMM was constructed to map the iEEG signals into either normal or preictal state. Baum-Welch algorithm was used for parameter estimation and a Bayesian evolution model was used to estimate the time of state transition. | Data collected from male Sprague-Dawley rats with four implanted skull screw EEG electrodes placed bifrontally and posteriorly behind bromas and a fifth depth electrode placed in the hippocampus, were collected and used for this study. |
| Direitos et al. [114], 2012       | Identification of different states of epileptic brain                                    | The relative power in EEG sub-bands (delta, theta, alpha, beta, and gamma) was calculated and used for computing the topographic maps of each sub-band. The maps were then segmented and used to trigger the iEEG to 4 states (preictal, ictal, interictal, and seizure) HMM. The relative power was used to train the model and the Viterbi algorithm to decode the state sequence. | EpilepsiaIRE database [115]. |
| Abdullah et al. [105], 2012       | Seizure detection in iEEG                                                                | A three state discrete HMM was built to classify EEG segments into one of three states (ictal, preictal, and interictal). Seven level decomposition stationary wavelet transform (SWT) was applied on the signals (as input features for the model) and a code book was created to perform vector quantization. Baum-Welch algorithm was used for model parameter estimation and the Viterbi algorithm for recognition. This study employed a feature-level fusion model to feed the data into the prediction model. | Fteching Seizure Prediction EEG (FSPEEG) database [110]. |
| Smart and Chen [109], 2015        | Seizure detection in scalp EEG                                                           | The study used a 5 sec sliding window with 1 sec increments to process the EEG signal. A set of 45 measurements were calculated for each sliding window then principal component analysis (PCA) was used to reduce dimensionality. One of the used models was HMM, particularly a two states (seizure and non-seizure) HMM was constructed to perform the detection. Baum-Welch was used here as well to estimate the model parameters. This study used a feature-level fusion model for multi-channel EEG data to feed the data into the prediction model. | CSH-MIT Scalp EEG Databases [119]. |
| Petrazan et al. [117], 2000       | Direct detection of epileptic seizures in both scalp and intracranial EEG                 | A three state HMM was built to classify EEG segments into one of three states (ictal, preictal, and interictal). Seven level decomposition stationary wavelet transform (SWT) was applied on the signals (as input features for the model) and a code book was created to perform vector quantization. Baum-Welch algorithm was used for model parameter estimation and the Viterbi algorithm for recognition. This study employed a feature-level fusion model to feed the data into the prediction model. | CSH-MIT Scalp EEG Databases [119]. |
| Gilzer [118], 2005                | Identification of subject condition in terms of epilepsy (healthy, epilepsy patient during seizure-free interval, and epilepsy patient during seizure episode) using surface and intracranial EEG | The EEG signals were used to train an Elman RNN. This study used a feature-level fusion model for multi-channel EEG data to train the RNN. | Publicly available epilepsy database by University of Bonn [119]. |
| Kumar et al. [120], 2008          | Automatic detection of epileptic seizures in surface and intracranial EEG                 | Wavelet and spectral entropy were extracted from the EEG signals and used to train an Elman RNN. This study used a feature-level fusion model for multi-channel EEG data to train the RNN. | Publicly available epilepsy database by University of Bonn [119]. |
| Minayen et al. [121], 2010        | Automatic detection of epileptic seizures prior to or immediately after clinical onset in scalp EEG | EEG data was collected from five centers including Thomas Jefferson University, Dartmouth-Mountsin University, University of Virginia, UCLA and University of Michigan medical centers. The data was used to train a 2 state HMM to feed the data into the prediction model. | Publicly available epilepsy database by University of Bonn [119]. |
| Naderi and Mahdavi-Nasab [122], 2010 | Automatic detection of epileptic seizures in surface and intracranial EEG                  | Power spectral density was calculated for EEG signals using Welch method then a dimensionality reduction algorithm was applied and the output was used to train an Elman RNN. This study used a feature-level fusion model for multi-channel EEG data to train the RNN. | Publicly available epilepsy database by University of Bonn [119]. |
| Vadapavut et al. [123], 2018      | Automated patient specific seizure detection using scalp EEG                              | The preprocessed EEG signals were segmented into 1 sec non-overlapping epochs and used to train a BRNN. Data from all channels were used simultaneously (feature-level fusion model). | Publicly available epilepsy database by University of Bonn [119]. |
| Takabhi [124], 2017               | Epileptic seizures detection                                                             | Single-channel EEG data (no multi-channel fusion) were used to train a GRU-based RNN that classifies each EEG segment into one of three states: healthy, interictal, or ictal. Two layers of GRU were used, the first was followed by a fully connected layer and the second was followed by a logistic regression classification layer. | Publicly available epilepsy database by University of Bonn [119]. |
| Gohmohammadi et al. [125], 2017   | Epileptic seizures prediction                                                            | Linear frequency cepstral coefficient feature extraction was performed for the EEG data and used to feed a CRNN that is based on a bidirectional LSTM. Features from multi-channel EEG were fused prior to feeding into the CRNN. The network used in this study employed both 2D and 1D CNN at different stages. Another network where LSTM was replaced with GRU was developed as well for comparison. | A subset of the YUH EEG Corpus (YUEEG) [128] that has been manually annotated for seizure events [127]. |
| Raghav et al. [126], 2017         | Epileptic seizures classification                                                        | The study developed two techniques that are based on Elman RNN that works on features extracted from EEG signals. The first technique used wavelet decomposition with the estimation of log energy and norm entropy to feed the RNN classifier (normal vs preictal). The second way extracted the log energy entropy to feed the RNN classifier. | Publicly available epilepsy database by University of Bonn [119]. |
| Abdulhamed et al. [120], 2018     | Epileptic seizures detection                                                             | This study used EEG signals to feed a 1D CRNN that is based on bidirectional LSTM to classify EEG segments into one of two states (normal and seizure). | Publicly available epilepsy database by University of Bonn [119]. |
| Dasou et Bayount [130], 2018      | Epileptic seizures prediction                                                             | This study used raw EEG signals to feed a 2D CRNN that is based on a bidirectional LSTM to classify EEG segments into one of two classes (preictal and interictal). | A dataset recorded at Children’s Hospital Boston which is publicly available [59, 110]. |
| Hussein et al. [131], 2019        | Epileptic seizures prediction                                                             | This study developed an LSTM-RNN that takes raw EEG signals as input in order to create predictions. The network was composed of a one layer LSTM followed by a fully connected layer and an average pooling layer to combine the temporal features and then an output softmax layer. | Publicly available epilepsy database by University of Bonn [119]. |

7. Event detection in EMG

Electromyography (EMG) is the method of sensing the electric potential evoked by the activity of muscle fibers as driven by the spikes from spinal motor neurons. EMGs are recorded either using surface electrodes or via needle electrodes; however, surface EMG (sEMG) is rarely used clinically in the evaluation of neuromuscular function and its use is limited to the measurement of voluntary muscle activity [161]. Routine evaluation of the neuromuscular function...
is typically performed using needle (invasive) EMG that, despite of its effectiveness and the availability of several electrode types that suite many clinical questions, is often painful and traumatic and may lead to the destruction of several muscle fibers [161, 162]. sEMG has been widely used as control signals for multiple applications especially in rehabilitation including but not limited to body-powered prostheses, grasping control, and gesture based interfaces [163]. A myoelectric signal usually has its manifested events as two states, the first is the transient state which emanates as the muscle goes from the resting state to voluntary contraction. The second is the steady state which represents maintaining the contraction level in the muscle [163]. It has been shown that the steady state segments are more robust as control signals.

### Table 4

| Publication | Event under investigation | Implementation details | Dataset |
|-------------|---------------------------|------------------------|---------|
| Obremski et al. [138], 2001 | 5 tasks BCI system (imagining left-hand, right-hand, foot, tongue movements, or simple calculation) | A 5 state HMM with 8 (max) Gaussian mixtures per state, was used to model the spatiotemporal patterns in each signal segment. Features were extracted from all electrodes and fused into a common feature vector and it had its dimensionality reduced before use in building the model. The expectation-maximization algorithm was used for the estimation of the transition matrix and the mixtures. | Data from 3 male subjects were collected for motor imagery tasks with the participants free of any medical or central nervous system conditions. |
| Obremski et al. [139], 2001 | Two class motor imagery (left and right hands) BCI | Two 5 state HMMs (one for each class) with 8 (max) Gaussian mixtures per state, was used to model the spatiotemporal patterns in each signal segment. The Hjorth parameters of a set of channels (C3 and C4) were fused and fed into the HMM model to calculate the single best path probability for both models. The expectation-maximization algorithm was used for the estimation of the transition matrix and the mixtures. | Data from 4 male subjects were collected for motor imagery tasks with the participants free of any medical or central nervous system conditions. |
| Platt et al. [140], 2003 | Two class motor imagery BCI for virtual keyboard control | Two HMMs for each class were trained and the maximal probability achieved by the respective HMM-model represents the chosen class. | Signals from two bipolar channels were acquired from three able-bodied subjects. |
| Szyman [141], 2006 | EEG-based mental task classification (left or right hand movement) | Unimodal HMM and multi-Gaussian HMM-based classifiers have been used for raw EEG signals. | Dataset III of BCI Competition II (2003) provided by the BCI research group at Graz University [142]. |
| Söder et al. [142], 2014 | P300 speller | Asingle layer BRNN was used to perform classification and compared with the regular neural network implementations. | Data was collected from 15 healthy graduate students and faculty with normal or corrected to normal vision between the ages of 20 and 36. |
| Erfanian and Mahmoudi [143], 2008 | Real-time adaptive noise canceler for ocular artifact suppression in EEG | A recurrent multi-layer perception with a single hidden layer was trained for the noise canceling with the inputs as the contaminated EEG signal and the reference EOG. | A simulated EEG dataset was used for this study, generated through Gaussian white noise-based autoregressive process. |
| Forney and Anderson [144], 2013 | EEG signal forecasting and mental tasks classification | For each subject a single classifier was trained to classify the mental task associated with the EEG signals. | A 4 class dataset was collected from 3 subjects including combinations of the following mental tasks: clenching of right hand, shaking of left leg, visualization of a tumbling cube, counting backwards from 100 by 3's, and singing a favorite song. |
| Bader et al. [145], 2015 | EEG classification for 2 class motor imagery (left hand and right hand) | An LSTM-based classifier was trained and evaluated for EEG oscillatory components classification and compared with the regular neural network implementations. | BCI competition IV (2007) dataset 2b [146]. |
| Maddu et al. [150], 2016 | P300 BCI classification | A 3D CNN in conjunction with a 2D CNN were combined with an LSTM-based RNN to capture spatiotemporal patterns in EEG. | Data from P300 segment speller were collected, where the subjects mentally counted backward from 100 by 3's and singing a favorite song. |
| Thomas et al. [151], 2017 | Steady-state visual evoked potential (SSVEP)-based BCI classification | An EEG signal and the reference EOG. | A subset of ImageNet dataset (42 classes) [152] was used to generate visual stimuli for six subjects while EEG data is recorded. |
| Spampart et al. [152], 2017 | Visual object classifier using EEG signals evoked by visual stimuli | An LSTM-based encoder to learn high order and temporal feature representations from EEG signals and then a classifier is used for identifying the visual object named the stimuli. | Dataset of BCI Competition IV (2007) dataset 2b [153]. |
| Hojman et al. [153], 2019 | Intercortical BCI for cursive control | An interclass layer LSTM-based decoder was built with three outputs to generate the cursor speed in x and y directions in addition to the distance to target. | Intercortical neural signals recorded from three participants, each with 20 6-channel micro-electrode arrays [157]. |
| Zhang et al. [154], 2020 | EEG-Based Human Intention | In this study, multi-channel raw EEG sequences into in-shape-like representations that can capture spatiotemporal characteristics of EEG and its acquisition. These meshes are then fed into deep neural networks that perform the recognition process. Multiple network architectures were investigated including a CNN that extracts the features, followed by a two-layer LSTM-based RNN to extract the temporal features, then a fully connected layer and an output layer. The second network was investigated for extraction of a speech signal, followed by a two-layer LSTM-based RNN to extract the temporal features and the second was a multi-layer 2D/3D CNN to extract the spatial features and the output from the two branches is fused and used for recognition. | EEG Motor Movement/Imagery Dataset [59, 155]. |
| Tortora et al. [155], 2020 | BCI for gait decoding from EEG | EEG data was recorded from 11 subjects walking on a treadmill using a 64-channel amplifier and 10/20 montage. | EEG data was recorded from 11 subjects walking on a treadmill using a 64-channel amplifier and 10/20 montage. |
compared to the transient state due to longer duration and better classification rates [164]. As follows in Table 5, we give a review about the recent advances in the detection of myoelectric events in EMG signals.

### Table 5: Summary of event detection in EMG signals.

| Publication | Event under investigation | Implementation details | Dataset |
|-------------|---------------------------|------------------------|---------|
| Chan and Englehart [188], 2005 | Continuous identification of six classes hand movement in eEMG | An HMM with uniformly distributed initial states and Gaussian observation probability density function whose parameters can be completely estimated from the training data, was constructed for the detection process. The expectation-maximization algorithm wasn’t used here due to the assumption of uniform initial state probabilities and directly estimating the Gaussian parameters from the training data. Overlapping 256 ms observation windows were used and in each observation window the root mean square value and the first 6 autocorrelative coefficients were computed as features. | 4-channel eEMG collected from the forearm of 15 subjects for six distinct motions (wrist flexion, wrist extension, supination, pronation, hand open, and hand close) [165] |
| Zhang et al. [197], 2011 | Hand gesture recognition in acceleration and eEMG | In this work, the authors actually identified the active segments via processing and thresholding of the average signal of the multichannel eEMG. The onset is when the energy is higher than threshold and the offset when the energy is lower than another threshold. Features from time, frequency, and time-frequency domains were extracted from both acceleration signals and eEMG, and fed to five separate HMMs for classification. Baum-Welch algorithm was used for training with Gaussian multivariate distribution for observations. Decision making here is done in a tree-structure (decision-based fusion) through four layers of classifiers with the last layer as the HMM. | eEMGs and 36 acceleration were collected from two right-handed subjects who repeated 43 known activities on a computer keyboard. The output of the network was enough to achieve high accuracy. |
| Wheeler et al. [189], 2006 | Hand gesture recognition in eEMG | Moving average was used on the eEMG signals to provide the input for an RNN and the output belief state. A Viterbi algorithm was used for the decoding process. The method introduced was tested on both simulated and experimental EMG signals. The simulated signals were generated via Markov model under 10 kHz sampling frequency and with filter shapes obtained from experimental EMG for more realistic simulations. The experimental EMG signals were acquired from the anterior digitorum of a healthy subject with teflon coated stainless steel wires. | Data from one participant repeating 4 gestures on a joy stick (up, down, left, and right) 5 times per gesture, were collected using four pairs of dry electrodes. Another portion of data was collected using 8 pairs of wet electrodes on gestures of typing on a number pad keyboard (0-9) for 40 strokes on each key. |
| Menafro et al. [198], 2014 | Extraction of the activity of individual motor neurons in single channel intramuscular EMG (iEMG) | The iEMG signal was modeled as a sum of independent filtered spikes trains embedded in noise. A Markov model of the spikes trains was introduced where the sparsity of the trains was exploited through modeling the time between spikes as discrete weibull distribution. An online estimation method for the weibull distribution parameters was introduced as well as an implementation of the impulses responses of the model. | EMG signals were collected from articulatory facial muscles from 8 Korean male subjects. The subjects were asked to pronounce each word from a word vocabulary in a consistent manner in addition to generating a random set of words based on this vocabulary in a consistent manner. The iEMG signals were acquired from the anterior digitorum of a healthy subject with teflon coated stainless steel wires. |
| Lee et al. [170], 2005 | eEMG-based speech recognition | A continuous HMM was constructed with Gaussian mixture model adopted for eEMG-based word recognition based on log mul-filter bank spectrum of the windowed EMG signals. The segmental K-means algorithm was used for optimal HMM parameters estimation where HMM parameters for the i-th state and word are estimated from the observations of the corresponding states of the same word. A dynamic programming algorithm was used for the decoding process. | 22 subjects, 41 different words based on a 60-word vocabulary in a consistent manner in addition to generating a random set of words based on this vocabulary. |
| Chan et al. [171], 2002 | eEMG-based automatic speech recognition | A six state left-right HMM with simple mixture observation densities, was constructed for identifying the words based on three features extracted from eEMG that included the first two autocorrelation coefficients and the integrated absolute value. HMM was trained in this work using the expectation-maximization algorithm. | eEMGS were from five articulatory facial muscle samples. The dataset used was the same as the one used for the experiments described in [172] with ten-English word vocabulary. |
| Li et al. [172], 2015 | Identification/prediction of functional electrical stimulation (FES)-induced muscular dynamics with evolved EMG (eEMG) | A CNN with 3 CNN layers and 2 LSTM layers was used for the prediction and the model used the power spectral density as input. A six state left-right HMM with simple mixture observation densities, was constructed for identifying the words based on three features extracted from eEMG that included the first two autocorrelation coefficients and the integrated absolute value. HMM was trained in this work using the expectation-maximization algorithm. | The experiments were conducted on 5 subjects with spinal cord injury. |
| Xie et al. [174], 2016 | Hand motion estimation from eEMG | A CNN with 3 CNN layers and 2 LSTM layers was used for the prediction and the model used the power spectral density as input. | eEMGs were from 8 healthy subjects using 5 pairs of bipolar electrodes placed on shoulder to record EMG from biceps brachii, triceps brachii, anterior deltoid, posterior deltoid, and middle deltoid. The hand position in 3D space was tracked as the objectives for this system. |
| Quevira et al. [175], 2018 | Simple hand gesture movement identification in eEMG | An LSTM-based RNN was used to predict the stimulated muscular torque and track muscle fatigue. The model takes the eEMG measured at the extensor digitorum as input and produces the predicted torque. | Experiments were performed over the first and second sub-databases of NinaPro (Non Invasive Adaptive Prostheses) database [176] and a similar subset of the dataset described in [177] with ten-English word vocabulary. |
| Hu et al. [177], 2016 | Hand gesture recognition in eEMG | eEMG signals from all channels were segmented into windows of fixed sizes and transformed into an image representation that was then fed into a CNN with two convolutional layers, two locally connected layers, and three fully connected layers followed by an LSTM-based RNN and an attention layer to enhance the output of the network. | The experiments were performed over the first and second sub-databases of NinaPro (Non Invasive Adaptive Prostheses) database [176] and a similar subset of the dataset described in [177] with ten-English word vocabulary. |
| Samadani [178], 2016 | EMG-Based Hand Gesture Classification | Different RNN architectures were tested in this study to choose the best performing architecture. The evaluated models included uni and bidirectional LSTM- and CNN-based RNNs with attention mechanisms. The models worked on the preprocessed (denoised) raw EMG signals. | Publicly-available NinaPro hand gesture dataset (NinaPro2) was used [160]. |
| Simko et al. [179], 2016 | EMG-based online gestures classification | Features were extracted from multi-channel EMG (standard deviation along each time frame) and fed into a dynamic RNN model that is composed of a dense layer followed by an LSTM-based RNN and another dense layer followed by the output layer. This model was compared to a similar GPU-based model and another static feed forward neural network model. This study used combined feature vector as an input for the model. | The synthetic sequences of the UCSD DualMyo dataset [180] and a similar subset of the NinaPro DB5 dataset [181] was used. |

8. Event detection in other biomedical signals

Physiological monitoring is an essential part of all care units nowadays and it is not limited to the aforementioned biomedical signals only. Tens of variables are collected in the form of time series containing hundreds of events that are of importance to the diagnosis and treatment/rehabilitation.
Event detection methods have had a strong presence in the analysis of such series. For instance, cardiovascular disorders are not only assessed through ECG but also phonocardiogram is used as an easier way for general practitioner to identify the changes in heart sounds. Extracting the cardiac cycle has been one of the major problems in phonocardiogram as well and was addressed using HMMs in multiple pieces of work \([184–187]\). On the other hand, most of RNN based methods in phonocardiogram, have been used for pure classification purposes and anomaly recognition \([152]\).

3D acceleration is an emerging technology as well, that has been extensively used in the assessment and detection of many medical conditions in swallowing \([188]\) and human gait analysis \([189]\). In swallowing, acceleration signals have been used for the detection of pharyngeal swallowing activity via maximum likelihood methods with minimum description length in \([16]\) and using short time Fourier transform and neural networks in \([14]\). RNNs were also employed for event detection in swallowing acceleration signals including the upper esophageal sphincter opening in \([15, 190]\), laryngeal vestibule closure \([191]\), and hyoid bone motion during swallowing \([192]\). In gait analysis, HMMs were used for recognition and extraction in multiple occasions \([193–196]\) as well as RNNs \([197–199]\).

9. Challenges and Future Directions

Event detection in biomedical signals is a critical step for diagnosis and intervention procedures that are extensively used on a daily basis in nearly every standard clinical setting. It also represents the core of various eHealth technologies that employ wearable devices and regular monitoring of physiological signs. Being such a fundamental operation that controls the clinical decision making process, it necessitates precise detection in a fairly complex environment that contains multiple events occurring concurrently. Particularly, false positive rate in clinical testing is an important indicator for how well the detection model generalizes and differentiates between the event of interest and the background noise. Building such highly accurate models depends on many factors that include the diversity in the used dataset and labels in addition to model capacity.

9.1. Classical Models Scaling: Challenges

As mentioned before, biomedical signals are the manifestation of well-coordinated, yet complex physiological processes which involve various anatomical structures that are close in position and share several functions. Hence, the collected signals pick not only the target physiological process but also other unavoidable neighbor processes. An example of that is the detection of the combined activation for multiple muscles in sEMG, eye blinking along with neural activity in EEG, and head movement along with swallowing vibrations in swallowing accelerometry. Extraction of the event of interest in this case requires the exhausting labeling of the underlying set of processes in order to be able to build the predefined state space for classical stochastic methods such as HMM, from which the state sequence is drawn. Manual labeling or interpretation of the biomedical signals is not only an exhausting task, but also requires extensive domain knowledge and expertise to perform.

One way that can be used to enhance the expressive power of stochastic models such as HMM, is the inclusion of non-Gaussian mixtures which can boost the performance in many cases because Gaussianity is not always a reasonable assumption in many applications. One of the mixtures that was proposed as an extension for non-Gaussian mixtures, is independent component analyzers mixture model (ICAMM) and it has been applied in multiple biomedical signal applications such as sleep disorders detection and classification of neuropsychological tasks in EEG \([34, 35]\).

An additional way to increase the model capacity and its ability to model the underlying sequence of events, is through using strongly representing domain features. One of the most popular domains representations, is wavelet decomposition which has proven its superiority to provide high level representation of events in a wide variety of biomedical signals such as phonocardiograms \([9, 187]\), EEG \([104, 117, 120, 121]\), and EMG \([164]\). Handcrafting features, however, is not an easy task and requires an extensive domain knowledge and significant efforts to come up with cues that trigger the identification of specific signal components. Furthermore, mapping the feature space into a more comprehensive space of less dimensionality is often a paramount operation prior to building the model. Given the previous factors, models that are able to learn high level representations simultaneously from raw signals and have the massive expressive power to model tasks involving long time lags, can be of a great benefit \([200]\).

9.2. High Capacity Models Embedding Feature Extraction

The evolution of deep learning has revolutionized the way in which problems are addressed and instead of classification and detection systems that solely relied on handcrafted features, end-to-end systems are being trained to take care of all steps from the raw input till the final output. End-to-end systems are complex, although rich, processing pipelines that make the most of the available information through using a unified scheme that trains the system as a whole from the input till the output is produced \([201]\). It has been shown that deep architectures can replace handcrafted feature extraction stages and work directly on raw data to produce high levels of abstraction. RNNs have been introduced in 1996 for the identification of arm kinematics during hand drawing from raw EMG signals \([202]\) and then the same architecture was adopted for lower limb kinematics in \([203]\). In both studies, the authors verified that an RNN was able to map the relationship between raw EMG signals and limbs’ kinematics during drawing for the arm and human locomotion for the lower limb. Chauhan and Vig \([204]\) and Sujadevi et al. \([205]\) have also used more sophisticated multi-layer LSTM-based RNN architectures on raw ECG signals for arrhythmia detection. Spampinato et al. \([154]\) have employed RNNs as well to extract discriminative brain manifold for visual categories from...
EEG signals. Further, Vidyaratne et al. [123] used RNNs for seizure detection in EEG; however, they used a denoised and segmented version of the signals. As mentioned earlier, although RNNs are efficient in modeling long contexts, they tend to have the error signals propagate through a tremendous number of steps when being fed highly sampled inputs such as raw signals which affects the network optimizability and training speed [49, 50].

In this regard, convolutional neural networks (CNNs) have been utilized to perceive small local contexts which then are propagated to an RNN for the perception of temporal contexts or a feed-forward network for a classification or prediction target. CNNs were introduced as a solution to enable recognition systems to learn hierarchical internal representations that form the scenes in vision applications (pixels form edglets, edglets form motifs, motifs form parts, parts form objects and objects form scenes) [206, 207]. Thus, CNNs are basically multi-stage trainable architectures that are stacked on top of each other to learn each level of the feature hierarchy [200, 206]. Each stage is usually composed of three layers, a filter bank layer, a non-linear activation layer, and a pooling layer. A filter bank layer extracts particular features at all locations on the input. The non-linear activation works as a regulator that determines whether a neuron should fire or not through checking the its value and deciding if the following connections should consider this neuron activated [200]. A pooling layer represents a dimensionality reduction procedure that processes the feature maps in order to produce lower resolution maps that are robust to the small variations in the location of features [206]. The coefficients of the filters are the trainable parameters in the CNNs and they are updated simultaneously by the training algorithm to minimize the discrepancy between the actual output and the desired output [206].

The design concept of CNNs first evolved for vision applications; but since then, the same concept is being adopted for pattern analysis and recognition in biomedical signals [174, 208–212]. For instance, Shashikumar et al. [210] used a 5-layer 2D CNN followed by a BRNN in association with soft attention mechanism to process the wavelet transform of ECG signals for the detection of atrial fibrillation. Tan et al. [211] also used a 1D 2-layer CNN with a 3-layer LSTM-based RNN for the detection of coronary artery disease in ECG. Further, Xiong et al. [212] used a residual convolutional recurrent neural network for the detection of cardiac arrhythmia in ECG. All these experiments using RNNs on top of CNNs for biomedical signal analysis were successful to produce extremely high levels of abstraction and rich temporal representation that can perceive long range contexts without human intervention in addition to being easier to optimize computationally. CNNs have been also utilized in association with fully connected networks to increase the capacity of HMMs in connectionist hybrid DNN-HMM models due to the ability of CNNs to process high-dimensional multi-step inputs [213]. Such hybrid systems provided state of the art performance especially in the field of handwriting recognition [214, 215].

9.3. Transfer Learning

Despite the fact that most of the previously mentioned methods are achieving great results on certain datasets, it is popular that they can easily overfit the data, resulting in poor generalization. Thus, it requires not only very large but also diverse datasets to train and validate models that well generalize. In biomedical signal processing field, the collection of such datasets may pose a challenge towards developing reliable models. Strictly speaking, it may not be feasible to find a large population of subjects when studying a rare disease and yet if it is feasible, it is extremely difficult to acquire the expert reference annotations for the underlying dataset [216]. Many factors contribute to this, as mentioned before, the noisy nature of biomedical signals increases the difficulty of manual interpretation and necessitates the presence of reference modalities to acquire accurate information about the processes such as collection of x-ray videofluoroscopy simultaneously with swallowing accelerometry [217]. Another factor is that the experts annotating the data need to maintain high record of reliability across time and to be compared to peer experts which might be difficult to achieve or require continuous training and checking of the experts’ reliability.

One way to overcome limited-size and/or diversity datasets, is to utilize the the pretrained models from relatively different domains and apply them to solve the particular targeted problem or so-called transfer learning [218]. In transfer learning, the pretrained model’s weights are used as initialization and then fine-tuned accordingly to fit the new dataset. In most cases, retraining happens in a much lower (10 times smaller) learning rate than the original. Transfer learning has been used for event detection and classification tasks in multiple biomedical signals including ECG for cardiac arrhythmia detection [219], EEG for drowsiness detection [220] and driving fatigue detection [221], and EMG for hand gesture classification [222]. However, one thing worth mentioning is that transfer learning sometimes may not help perform better than the originally trained model if there exist huge differences between the datasets or deterioration in inter-subject variability [218].

10. Conclusion

In this paper, we provided a comprehensive review of event extraction methods in biomedical signals, in particular hidden Markov models and recurrent neural networks. HMM is a probabilistic model that represents a sequence of observations in terms of a hidden sequence of states and sets the concepts and methods on how to find the optimal state sequence that best describes the observations. RNN is a type of neural networks that was introduced to model the time dependency and perform contextual mapping in sequences. This review showed that the presence of dynamic programming algorithms like the EM and Viterbi, led to the widespread use of HMMs which were used to dynamically transcribe the context of many biomedical signals. It wasn’t too long until HMMs became insufficient for time series modeling needs, specifically modeling long range dependencies.
and larger state spaces, and RNNs started to gradually replace HMMs in time-dependent contextual mappings. So far, RNNs have proven superiority in time series modeling especially in biomedical signals and continue to expand their domination in building automatic detection and diagnosis systems through the emerging designs and practices experimented in nearly every field.

Acknowledgments

The work reported in this manuscript was supported by the National Science Foundation under the CAREER Award Number 1652203. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Science Foundation.

References

[1] L. Glass, Synchronization and rhythmic processes in physiology, Nature 410 (2001) 277–284.
[2] R. M. Rangayyan, N. P. Reddy, Biomedical signal analysis: A case-study approach, Annals of Biomedical Engineering 30 (2002) 983–988.
[3] P. Rashidi, A. Miahilidis, A survey on ambient-assisted living tools for older adults, IEEE Journal of Biomedical and Health Informatics 17 (2013) 579–590.
[4] J. Kim, M. Kim, I. Won, S. Yang, K. Lee, W. Huh, A biomedical signal segmentation algorithm for event detection based on slope tracing, in: Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE, 2009, pp. 1889–1892. doi:10.1109/IEMBS.2009.533874.
[5] J. Andreu-Perez, C. C. Poon, R. D. Merrifield, S. T. Wong, G. Z. Yang, Big data for health, IEEE Journal of Biomedical and Health Informatics 19 (2015) 1193–1208.
[6] R. Gravina, P. Alinea, H. Ghasemzadeh, G. Fortino, Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges, Information Fusion 35 (2017) 68–80.
[7] D. P. Mandic, D. Obradovic, A. Kuh, T. Adali, U. Trutschell, M. Golz, P. De Wilde, J. Barria, A. Constantinides, J. Chambers, W. Duch, J. Kapczynz, E. Oja, S. Zadrozny, Data Fusion for Modern Engineering Applications: An Overview, 2005.
[8] D. Mandic, M. Golz, A. Kuh, D. Obradovic, T. Tanaka, Signal Processing Techniques for Knowledge Extraction and Information Fusion, Springer US, 2008.
[9] L. Huiying, L. Sakari, H. Iiro, A heart sound segmentation algorithm using wavelet decomposition and reconstruction, in: Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, volume 4, IEEE, 1997, pp. 1630–1633.
[10] J. Pan, W. J. Tompkins, A real-time QRS detection algorithm, IEEE Transactions on Biomedical Engineering 32 (1985) 230–236.
[11] V. Srinivasan, C. Esowaran, N. Sriraam, Approximate entropy-based epileptic EEG detection using artificial neural networks, IEEE Transactions on Information Technology in Biomedicine 11 (2007) 288–295.
[12] N. Kannathal, M. L. Choo, U. R. Acharya, P. K. Sadasivan, Entropies for detection of epilepsy in EEG, Computer Methods and Programs in Biomedicine 80 (2005) 187–94.
[13] A. Schllogl, F. Lee, H. Bischof, G. Pfurtscheller, Characterization of four-class motor imagery EEG data for the BCI-competition 2005, Journal of Neural Engineering 2 (2005) L1–L2.
[14] Y. Khalifa, J. L. Coyle, E. Sejdic, Non-invasive identification of swallows via deep learning in high resolution cervical auscultation recordings, Scientific Reports 10 (2020) 8704.
[15] Y. Khalifa, C. Donohue, J. Coyle, E. Sejdic, Upper esophageal sphincter opening segmentation with convolutional recurrent neural networks in high resolution cervical auscultation, IEEE Journal of Biomedical and Health Informatics (2020).
[16] E. Sejdić, C. M. Steele, T. Chau, Segmentation of dual-axis swallowing accelerometry signals in healthy subjects with analysis of anthropometric effects on duration of swallowing activities, IEEE Transactions on Biomedical Engineering 56 (2009) 1090–1097.
[17] S. Damouras, E. Sejdić, C. M. Steele, T. Chau, An online swallow detection algorithm based on the quadratic variation of dual-axis accelerometry, IEEE Transactions on Signal Processing 58 (2010) 3352–3359.
[18] Z. C. Lipton, J. Berkowitz, C. Elkan, A critical review of recurrent neural networks for sequence learning, arXiv preprint arXiv:1506.00019 (2015).
[19] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (1997) 1735–1780.
[20] L. R. Rabiner, A Tutorial on hidden Markov-models and selected applications in speech recognition, Proceedings of the IEEE 77 (1989) 257–286.
[21] P. J. Werbos, Backpropagation through time - what it does and how to do it, Proceedings of the IEEE 78 (1990) 1550–1560.
[22] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, Nature 323 (1986) 533–536.
[23] A. Graves, G. Wayne, I. Danihelka, Neural Turing machines, CoRR abs/1410.5401 (2014).
[24] A. Cohen, Hidden Markov models in biomedical signal processing, in: Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, volume 3, IEEE, 1998, pp. 1145–1150.
[25] L. E. Baum, T. Petrie, Statistical inference for probabilistic functions of finite state Markov chains, Annals of Mathematical Statistics 37 (1966) 1554–1563.
[26] D. Jurafsky, J. H. Martin, Speech and language processing, 2nd ed., Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2009.
[27] L. E. Baum, J. A. Eagon, An Inequality with Applications to Statistical Estimation for Probabilistic Functions of Markov Processes and to a Model for Ecology, Bulletin of the American Mathematical Society 73 (1967) 360–363.
[28] L. E. Baum, G. R. Sell, Growth transformations for functions on manifolds, Pacific Journal of Mathematics 27 (1968) 211–227.
[29] L. Baum, An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process, in: Proceedings of the 3rd Symposium on Inequalities, volume 3, 1972, pp. 1–8.
[30] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 39 (1977) 1–38.
[31] L. A. Liporace, Maximum-likelihood estimation for multivariate observations of Markov sources, IEEE Transactions on Information Theory 28 (1982) 729–734.
[32] B. H. Jiang, Maximum-likelihood estimation for mixture multivariate stochastic observations of Markov-chains, AT&T Technical Journal 64 (1985) 1235–1249.
[33] Levinson, S. M. Sondhi, Maximum likelihood estimation for multivariate mixture observations of markov chains, IEEE Transactions on Information Theory 32 (1986) 307–309.
[34] G. Safont, A. Salazar, L. Vergara, E. Gómez, V. Villanueva, Multichannel dynamic modeling of non-Gaussian mixtures, Pattern Recognition 93 (2019) 312–323.
[35] A. Salazar, L. Vergara, R. Miralles, On including sequential dependence in ICA mixture models, Signal Processing 90 (2010) 2314–2318.
[36] A. Graves, Supervised sequence labelling, in: Supervised sequence labelling with recurrent neural networks, Springer, 2012, pp. 5–13.
[37] J. L. Elman, Finding structure in time, Cognitive Science 14 (1990) 179–211.
[38] M. I. Jordan, Attractor dynamics and parallelism in a connectionist model of a simple motor control system, in: J. Diederich (Ed.), Artificial Neural Networks, IEEE Press, Piscataway, NJ, USA, 1990, pp. 112–127.
[115] M. Ible, H. Feldwisch-Drentrup, C. A. Teixeira, A. Witon, B. Schelter, J. Timmer, A. Schulze-Bonhage. EPIEPILEPSIAE - a European epilepsy database, Computer Methods and Programs in Biomedicine 106 (2012) 127–138.

[116] A. H. Shoeb, Application of machine learning to epileptic seizure onset detection and treatment, [PhD] Thesis, Massachusetts Institute of Technology, 2009.

[117] A. Petroelian, D. Prokhorov, R. Homan, R. Dasheiff, D. Wunsch, Recurrent neural network based prediction of epileptic seizures in intra- and extracranial EEG, Neurocomputing 30 (2000) 201–218.

[118] N. F. Güler, E. D. Übeyli, n. Güler, Recurrent neural networks employing Lyapunov exponents for EEG signals classification, Expert Systems with Applications 29 (2005) 506–514.

[119] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, C. E. Elger. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: dependence on recording region and brain state, Physical Review. E: Statistical, Nonlinear, and Soft Matter Physics 64 (2001) 061907.

[120] S. P. Kumar, N. Srirama, P. G. Benakop. Automated detection of epileptic seizures using wavelet entropy feature with recurrent neural network classifier, in: Proceedings of the IEEE Region Region Conference, 2008, pp. 1–5.

[121] G. R. Minasyan, J. B. Chatten, M. J. Chatten, R. N. Harner. Patient-specific early seizure detection from scalp EEG, Journal of Clinical Neurophysiology 27 (2010) 163–178.

[122] M. A. Naderi, H. Mahdavi-Nasab. Analysis and classification of EEG signals using spectral analysis and recurrent neural networks, in: Proceedings of the 17th Iranian Conference of Biomedical Engineering, 2010, pp. 1–4.

[123] L. Vidyaratne, A. Glandon, M. Alam, K. M. Iftekharuddin. Deep recurrent neural network for seizure detection, in: Proceedings of the IEEE International Joint Conference on Neural Networks, IEEE, 2016, pp. 1202–1207.

[124] S. S. Talathi, Deep Recurrent Networks for seizure detection and early seizure detection systems, arXiv preprint arXiv:1706.02383 (2017).

[125] M. Golmohammadi, S. Ziyabari, V. Shah, E. Von Wethin, C. Campbell, I. Obeid, J. Picone. Gated recurrent networks for seizure detection, in: Proceedings of the 2017 IEEE Signal Processing in Medicine and Biology Symposium (SPMB), 2017, pp. 1–5. doi: 10.1109/SPMB.2017.8257829.

[126] I. Obeid, J. Picone, The Temple University Hospital EEG Data Corpus, Frontiers in Neuroscience 10 (2016).

[127] M. Golmohammadi, V. Shah, S. Lopez, S. Ziyabari, S. Yang, J. Camaratta, I. Obeid, J. Picone. The TUH EEG seizure corpus, in: Proceedings of the American Clinical Neurophysiology Society Annual Meeting, 2017, p. 1.

[128] S. Raghu, N. Sriraam, P. G. Kumar. Classification of epileptic seizures using wavelet packet log energy and norm entropies with recurrent Elman neural network classifier, Cognitive Neurodynamics 11 (2017) 51–66.

[129] A. M. Abdelhameed, H. G. Daoud, M. Bayoumi. Deep Convolutional Bidirectional LSTM Recurrent Neural Network for Epileptic Seizure Detection, in: 2018 16th IEEE International New Circuits and Systems Conference (NEWCAS), 2018, pp. 139–143. doi: 10.1109/NEWCAS.2018.8584542.

[130] H. Daoud, M. Bayoumi. Deep Learning based Reliable Early Epileptic Seizure Predictor, in: 2018 IEEE Biomedical Circuits and Systems Conference (BioCAS), 2018, pp. 1–4. doi: 10.1109/BIOCAS.2018.8584678.

[131] R. Hussein, H. Palangi, R. K. Ward, Z. J. Wang. Optimized deep neural network architecture for robust detection of epileptic seizures using EEG signals, Clinical Neurophysiology 130 (2019) 25–37.

[132] G. Pfurtscheller, C. Neuper. Motor imagery and direct brain-computer communication, Proceedings of the IEEE 89 (2001) 1123–1134.

[133] E. C. Leuthardt, G. Schalk, J. R. Wolpaw, J. G. Ojemann, D. W. Moran. A brain-computer interface using electrocorticographic signals in humans, Journal of Neural Engineering 1 (2004) 63–71.

[134] G. Schalk, E. C. Leuthardt. Brain-computer interfaces using electrocorticographic signals, IEEE Reviews in Biomedical Engineering 4 (2011) 140–154.

[135] K. Sayed, M. Kamel, M. Alhaddad, H. M. Malibary, Y. M. Kadah. Characterization of phase space trajectories for Brain-Computer Interface, Biomedical Signal Processing and Control 38 (2017) 55–66.

[136] K. Sayed, M. Kamel, M. Alhaddad, H. M. Malibary, Y. M. Kadah. Extracting phase space morphological features for electroencephalogram-based brain-computer interface, Journal of Medical Imaging and Health Informatics 7 (2017) 771–774.

[137] E. Donchin, K. M. Spencer, R. Wijesinghe. The mental prosthes: Assessing the speed of a P300-based brain-computer interface, IEEE Transactions on Rehabilitation Engineering 8 (2000) 174–179.

[138] B. Obermaier, C. Neuper, C. Guger, G. Pfurtscheller. Information transfer rate in a five-classes brain-computer interface, IEEE Transactions on Neural Systems and Rehabilitation Engineering 9 (2001) 283–288.

[139] B. Obermaier, C. Guger, C. Neuper, G. Pfurtscheller. Hidden Markov models for online classification of single trial EEG data, Pattern Recognition Letters 22 (2001) 1299–1309.

[140] G. Pfurtscheller, C. Neuper, G. R. Muller, B. Obermaier, G. Krausz, A. Schlogl, R. Scherer, B. Graimann, C. Keinrath, D. Skliris, M. Wortz, G. Supp, C. Schrank. Graz-BCI: State of the art and clinical applications, IEEE Transactions on Neural Systems and Rehabilitation Engineering 11 (2003) 177–180.

[141] S. Sohjoo, A. M. Nasrabad, M. R. H. Golpayegani. Classification of chaotic signals using HMM classifiers: EEG-based mental task classification, in: Proceedings of the 13th European Signal Processing Conference, 2005, pp. 1–4.

[142] G. Pfurtscheller, A. Schlögl, Dataset III: Motor imagery, Technical Report, 2003.

[143] H. Suk, S. Lee. Two layer hidden Markov models for multi-class motor imagery classification, in: Proceedings of the 1st Workshop on Brain Decoding: Pattern Recognition Challenges in Neuroimaging, 2010, pp. 5–8.

[144] C. Brunner, R. Leeb, G. Müller-Putz, A. Schlogl, G. Pfurtscheller. Dataset IIa: Graz dataset A, Technical Report, 2008.

[145] W. Speier, C. Arnold, J. Lu, A. Deshpande, N. Pouratian. Integrating language information with a hidden Markov model to improve communication rate in the P300 speller, IEEE Transactions on Neural Systems and Rehabilitation Engineering 22 (2014) 678–684.

[146] A. Erfanian, B. Mahmoudi. Real-time ocular artifact suppression using recurrent neural network for electro-encephalogram based brain-computer interface, Medical & Biological Engineering & Computing 43 (2005) 296–305.

[147] E. M. Forney, C. W. Anderson. Classification of EEG during imagined mental tasks by forecasting with Elman recurrent neural networks, in: Proceedings of the 17th IEEE Conference on Neural Networks, IEEE, 2011, pp. 2749–2755.

[148] D. Balderas, A. Molina, P. Ponce. Alternative classification techniques for brain-computer interfaces for smart sensor manufacturing environments, IFAC-PapersOnLine 48 (2015) 680–685.

[149] R. Leeb, F. Lee, C. Keinrath, R. Scherer, H. Bischof, G. Pfurtscheller. Brain-computer communication: Motivation, aim, and impact of exploring a virtual apartment, IEEE Transactions on Neural Systems and Rehabilitation Engineering 15 (2007) 473–482.

[150] R. Maddula, J. Stivers, M. Mousavi, S. Ravindran, V. de Sa. Deep recurrent convolutional neural networks for classifying P300 BCI signals, in: Proceedings of the 7th Graz Brain-Computer Interface Conference, 2017.

[151] J. Stivers, V. de Sa. Spelling in parallel: Towards a rapid, spatially independent BCI, in: Proceedings of the 7th Graz Brain-Computer Interface Conference, 2017.

[152] J. Thomas, T. Masszyczy, N. Sinha, T. Kluge, J. Dauwels. Deep learning-based classification for brain-computer interfaces, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 2017, pp. 234–239.

[153] V. P. Oikonomou, G. Liaros, K. Georgiadis, E. Chatziari, K. Adam, J. Thomas, T. Maszczyk, N. Sinha, T. Kluge, J. Dauwels. Deep learning-based classification for brain-computer interfaces, in: Proceedings of the 7th Graz Brain-Computer Interface Conference, 2017.
[192] S. Mao, Z. Zhang, Y. Khalifa, C. Donohue, J. L. Coyle, E. Sejdic, Neck sensor-supported hyoid bone movement tracking during swallowing, Royal Society Open Science 6 (2019) 181982.

[193] C. Nickel, C. Busch, S. Rangarajan, M. Mobius, Using hidden Markov models for accelerometer-based biometric gait recognition, in: Proceedings of the IEEE 7th International Colloquium on Signal Processing and its Applications, 2011, pp. 58–63.

[194] A. Mannini, A. M. Sabatini, A hidden Markov model-based technique for gait segmentation using a foot-mounted gyroscope, in: Proceedings of the 33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2011, pp. 4369–4373.

[195] C. Nickel, C. Busch, Classifying Accelerometer Data via Hidden Markov Models to Authenticate People by the Way They Walk, IEEE Aerospace and Electronic Systems Magazine 28 (2013) 29–35.

[196] G. Pananhandeh, N. Mohammadinia, A. Lejon, P. Handel, Continuous hidden Markov model for pedestrian activity classification and gait analysis, IEEE Transactions on Instrumentation and Measurement 62 (2013) 1073–1083.

[197] M. Inoue, S. Inoue, T. Nishida, Deep recurrent neural network for mobile human activity recognition with high throughput, Artificial Life and Robotics 23 (2018) 173–185.

[198] A. Lisowska, G. Wheeler, V. Ceballos Inza, I. Poole, An evaluation of supervised, novelty-based and hybrid approaches to fall detection using silmee accelerometer data, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 402–408.

[199] T. Theodoridis, V. Solachidis, N. Vretos, P. Daras, Human fall detection from acceleration measurements using a recurrent neural network, in: A. Chouraqui, P. de Carvalho (Eds.), Precision Medicine Powered by mHealth and Connected Health, Springer, Singapore, 2017, pp. 145–149.

[200] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (1998) 2278–2324.

[201] T. Glas machers, Limits of End-to-End Learning, arXiv preprint arXiv:1704.08305 (2017).

[202] G. Cheron, J.-P. Draye, M. Bourgeois, G. Libert, A dynamic neural network identification of electromyography and arm trajectory relationship during complex movements, IEEE Transactions on Biomedical Engineering 43 (1996) 552–558.

[203] G. Cheron, F. Leurs, A. Bengoetxea, J. P. Draye, M. Destrée, B. Dan, A dynamic recurrent neural network for multiple muscles electromyo graphic mapping to elevation angles of the lower limb in human locomotion, Journal of Neuroscience Methods 129 (2003) 95–104.

[204] S. Chauhan, L. Vig, Anomaly detection in ECG time signals via deep long short-term memory networks, in: Proceedings of the IEEE International Conference on Data Science and Advanced Analytics, 2015, pp. 1–7, doi: 10.1189/OSSA.2015.7348872.

[205] V. G. Sujadeva, K. P. Soman, R. Vinayakumar, Real-time detection of atrial fibrillation from short time single lead ECG traces using recurrent neural networks, in: S. M. Thampi, S. Mitra, J. Mukhopadhyay, K.-C. Li, A. P. James, S. Berretti (Eds.), Intelligent Systems Technologies and Applications, Advances in Intelligent Systems and Computing, Springer International Publishing, Cham, 2017, pp. 212–221, doi: 10.1007/978-3-319-68385-6_18.

[206] Y. LeCun, K. Kavukcuoglu, C. Farabet, Convolutional networks and applications in vision, in: Proceedings of the 2010 IEEE International Symposium on Circuits and Systems, 2010, pp. 253–256, doi: 10.1109/ISCAS.2010.5537907.

[207] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, L. D. Jackel, Handwritten Digit Recognition with a Back-Propagation Network, in: D. S. Touretzky (Ed.), Proceedings of the 3rd Conference on Neural Information Processing Systems, Morgan-Kaufmann, 1990, pp. 396–404.

[208] H. Cecotti, A. Graser, Convolutional neural networks for P300 detection with application to brain-computer interfaces, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (2011) 433–445.

[209] S. Kiranyaz, T. Ince, M. Gabbouj, Real-time patient-specific ECG classification by 1-D convolutional neural networks, IEEE Transactions on Biomedical Engineering 63 (2016) 664–675.

[210] S. P. Shashikumar, A. J. Shah, G. D. Clifford, S. Nemati, Detection of paroxysmal atrial fibrillation using attention-based bidirectional recurrent neural networks, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD ’18, ACM, London, United Kingdom, 2018, pp. 715–723, doi: 10.1145/3219819.3219912.

[211] J. H. Tan, Y. Hagiwara, W. Pang, I. Lim, S. L. Oh, M. Adam, R. S. Tan, M. Chen, U. R. Acharya, Application of stacked convolutional and long short-term memory network for accurate identification of CAD ECG signals, Computers in Biology and Medicine 94 (2018) 19–26.

[212] Z. Xiong, M. P. Nash, E. Cheng, V. V. Fedorov, M. K. Stiles, J. Zhao, ECG signal classification for the detection of cardiac arrhythmias using a convolutional recurrent neural network, Physiological Measurement 39 (2018) 094006.

[213] Y. M. Saidutta, J. Zou, F. Fekri, Increasing the learning Capacity of BCI Systems via CNN-HMM models, in: Proceedings of the 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, 2018, pp. 1–4.

[214] Z.-R. Wang, J. Du, W.-C. Wang, J.-F. Zhai, J.-S. Hu, A comprehensive study of hybrid neural network hidden Markov model for offline handwritten Chinese text recognition, International Journal on Document Analysis and Recognition (IJDAR) 21 (2018) 241–251.

[215] Z.-R. Wang, J. Du, J.-M. Wang, Writer-aware CNN for parsimonious HMM-based offline handwritten Chinese text recognition, Pattern Recognition 100 (2020) 107102.

[216] N. C. Dvornek, D. Yang, P. Ventola, J. S. Duncan, Learning generalizable recurrent neural networks from small task-MfMRI datasets, in: A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, G. Fichtinger (Eds.), Proceedings of the 21st Conference on Medical Image Computing and Computer Assisted Intervention, Springer International Publishing, 2018, pp. 329–337.

[217] C. Yu, Y. Khalifa, E. Sejdic, Silent aspiration detection in high resolution cervical auscultations, in: Proceedings of the IEEE-EMBS International Conference on Biomedical and Health Informatics, 2019, pp. 1–4.

[218] S. J. Pan, Q. A. Yang, A Survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering 22 (2010) 1345–1359.

[219] A. Isin, S. Ozdalili, Cardiac arrhythmia detection using deep learning, Procedia Computer Science 120 (2017) 268 – 275.

[220] C. Wei, Y. Lin, Y. Wang, T. Jung, N. Bigdely-Shamlo, C. Lin, Selective transfer learning for EEG-based drowsiness detection, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 2015, pp. 3229–3232.

[221] Y.-Q. Zhang, W.-L. Zheng, B.-L. Lu, Transfer components between subjects for EEG-based driving fatigue detection, in: S. Arik, T. Huang, W. K. Lai, Q. Liu (Eds.), Proceedings of the 29th Conference on Neural Information Processing Systems, Springer International Publishing, 2016, pp. 61–68.

[222] U. Côté-Allard, C. L. Fall, A. Drouin, A. Campeau-Lecours, C. Gosselin, K. Glette, F. Laviolette, B. Gosselin, Deep learning for electromyographic hand gesture signal classification using transfer learning, IEEE Transactions on Systems and Rehabilitation Engineering 27 (2019) 760–771.