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

Yassin Khalifa[a], Danilo Mandic[b] and Ervin Sejdić[a,c,d,e,*]

[a]Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA, USA
[b]Department of Electrical and Computer Engineering, Imperial College, London, SW7 2BT United Kingdom
[c]Department of Bioengineering, University of Pittsburgh, Pittsburgh, PA, USA
[d]Department of Biomedical Informatics, University of Pittsburgh, Pittsburgh, PA, USA
[e]Intelligent Systems Program, University of Pittsburgh, Pittsburgh, PA, USA

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 is able to 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 which 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 a 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.imedlab.org (E. Sejdić)
ORCID(s): 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 in order to best represent the target processes. In addition, 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 in order 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 sinusoides or exponentials and aim to estimate the values of the parameters contributing to these properties (e.g. amplitude, frequency, and phase of the sinusoides) [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 statistical models category 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: $P(q_t = j|q_{t-1} = i, q_{t-2} = k, \dots) = P(q_t = j|q_{t-1} = i)$. This results in the following properties for the transition probabilities:

$$\begin{aligned} a_{ij} &= P(q_t = j|q_{t-1} = i); \qquad i \geq 1,\ j \leq N \\ a_{ij} &\geq 0 \\ \sum_{j=1}^{N} a_{ij} &= 1 \end{aligned}$$

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

$$\begin{aligned} \pi_i &= P[q_1 = S_i]; \qquad 1 \leq i \leq N \\ \Pi &= [\pi_1, \pi_2, \dots, \pi_N]^T \end{aligned}$$

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

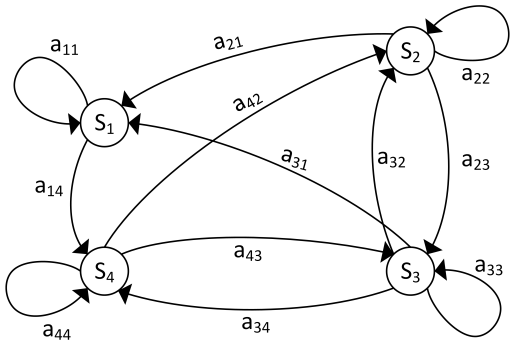

Figure 1: An example of a Markov chain with 4 states, S_1 to S_4, and selected state transitions. A set of probabilities is associated with each state to indicate how the system undergoes state change from one state to itself or to another at regular discrete times.

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.

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

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 $O = \{O_1, O_2, \dots, O_T\}$ 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 $O = \{O_1, O_2, \dots, O_T\}$, given the model ($P(O|\lambda)$).

Algorithm 1: HMM as observations generator

```
Set t = 1;
Choose an initial state q_1 = S_i according to Π;
while t ≤ T do
    Choose O_t = v_k according to the observation distribution in
      the current state (b_i(k));
    Move from the current state S_i to the new state q_{t+1} = S_j
      according to a_ij;
    set t = t + 1;
end
Result: O = {O_1, O_2, ..., O_T}
```

- **Decoding:** Choosing the optimal hidden state sequence $Q = \{q_1, q_2, \dots, q_T\}$ that best represents a given observation sequence ($O = \{O_1, O_2, \dots, O_T\}$).
- **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 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 $\alpha(t, i)$ which represents the probability of the partial observation sequence $P(O_1\ O_2\ \dots\ O_t, q_t = S_i|\lambda)$ [20, 27, 28]. The forward algorithm for the likelihood problem is fully described as follows:

Algorithm 2: The forward algorithm

```
O = {O_1, O_2, ..., O_T};
S ∈ {S_1, S_2, ..., S_N};
Create the forward probability table α[T, N];
foreach state S ∈ {S_1, S_2, ..., S_N} do
    α[1, S] ← π_S × b_S(O_1);                    // Initialization
end
foreach time step t ∈ 2, 3, ..., T do
    foreach state S ∈ {S_1, S_2, ..., S_N} do
        α[t, S] ← Σ_{Ŝ=S_1}^{S_N} α[t − 1, Ŝ] × a_{Ŝ,S} × b_S(O_t);
        // Induction
    end
end
P(O|λ(A, B, Π)) ← Σ_{S=S_1}^{S_N} α[T, S];        // Termination
Result: P(O|λ(A, B, Π))
```

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, $\beta(t,i) = P(O_{t+1},\ O_{t+2},\ \dots,\ O_T|q_t = S_i, \lambda(A,\ B,\ \Pi))$, which represents the probability of the partial observation sequence that starts one time step after the current observation, given the current state S_i 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

Create the backward probability table $\beta[T, N]$;
foreach *state* $S \in \{S_1, S_2, \dots, S_N\}$ **do**
    $\beta[T, S] \leftarrow 1$; // Initialization
end
foreach *time step* $t \in T-1, T-2, \dots, 1$ **do**
    foreach *state* $S \in \{S_1, S_2, \dots, S_N\}$ **do**
        $\beta[t, S] \leftarrow \sum_{\hat{S}=S_1}^{S_N} \beta[t+1, \hat{S}] \times a_{S,\hat{S}} \times b_{\hat{S}}(O_{t+1})$;
        // Induction
    end
end
Result: $\beta[T, N]$

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) $q_1,\ q_2,\ \dots,\ q_{t-1}$ that leads to this state. The Viterbi algorithm is shown in Algorithm 4.

2.5. Model Estimation Problem Solution

The third problem can be formulated as finding HMM's model parameters (A, B, Π) 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 λ,

Algorithm 4: The Viterbi algorithm

$O = \{O_1, O_2, \dots, O_T\}$;
$S \in \{S_1, S_2, \dots, S_N\}$;
Create the best path probability table $\delta[T, N]$;
Create the state index table (the index of state that by adding to the path, maximizes δ) $\psi[T, N]$;
foreach *state* $S \in \{S_1, S_2, \dots, S_N\}$ **do**
    $\delta[1, S] \leftarrow \pi_S \times b_S(O_1)$; // Initialization
    $\psi[1, S] \leftarrow 0$;
end
foreach *time step* $t \in 2, 3, \dots, T$ **do**
    foreach *state* $S \in \{S_1, S_2, \dots, S_N\}$ **do**
        $\delta[t, S] \leftarrow \max_{\hat{S}=S_1}^{S_N} \delta[t-1, \hat{S}] \times a_{\hat{S},S} \times b_S O_t$; // Induction
        $\psi[t, S] \leftarrow \arg\max_{\hat{S}=S_1}^{S_N} \delta[t-1, \hat{S}] \times a_{\hat{S},S} \times b_S O_t$;
    end
end
$P^* \leftarrow \max_{S=S_1}^{S_N} \delta[T, S]$; // Termination
$q_T^* \leftarrow \arg\max_{S=S_1}^{S_N} \delta[T, S]$;
for $t \in \{T,\ T-1,\ T-2,\ \dots,\ 2\}$ **do**
    $q_{t-1}^* \leftarrow \psi[t, q_t]$; // Backtracking
end
Result: The optimal state sequence: $q_1^*,\ q_2^*,\ \dots,\ q_T^*$

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

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 $\bar{\lambda}$ 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.

2.6. Continuous Density HMM

The previous described adaptations for HMM problems are based on the requirement that the observations are discrete which is considered restrictive due to the fact that in most cases they are continuous. Therefore, a necessary first step will be the transformation of continuous observation sequence into a discrete vector. This can be done through dividing the observations' space into sub-spaces and using codebooks to give discrete symbol/value for each sub-space [24]; however, this introduces quantization errors into the problem. One way to overcome this, is using continuous observation densities in HMM's. The finite mixture representation of the observation density function, is one of the representations that has a formulated re-estimation procedure: $b_j(O) = \sum_{m=1}^{M} c_{jm} \mathfrak{N}[O, \mu_{jm}, U_{jm}]$, where $1 \le j \le N$, O is the observation vector, c_{jm} is the mixture coefficient for the m^{th} mixture in state j, and $\mathfrak{N}$ is an elliptically or long-concave symmetric density with a mean vector of μ_{jm} and a covariance matrix of U_{jm} [31–33]. A Gaussian density function is usually used for $\mathfrak{N}$; however, other non-

Algorithm 5: The estimation algorithm

$O = \{O_1, O_2, \dots, O_T\}$;
$S \in \{S_1, S_2, \dots, S_N\}$;
Initialize $\bar{\lambda} = \lambda(A, B, \Pi)$;
repeat
  Using the forward-backward algorithm and $\bar{\lambda}$ calculate $\alpha[T, N]$ and $\beta[T, N]$;
  Create the probability tables $\xi[T, N, N]$ (the probability of being in a state S_i at time t and a state S_j at time $t+1$) and $\gamma[T, N]$ (the probability of being in a state S_i at time t);
  foreach *time step* $t \in 2, 3, \dots, T$ **do**
    foreach *state* $S \in \{S_1, S_2, \dots, S_N\}$ **do**
      foreach *state* $S^* \in \{S_1, S_2, \dots, S_N\}$ **do**
        $\xi[t, S, S^*] \leftarrow \frac{\alpha[t,S] \times a_{S,S^*} \times b_{S^*}(O_{t+1}) \times \beta[t+1,S^*]}{\sum_{S=S_1}^{S_N} \sum_{S^*=S_1}^{S_N} \alpha[t,S] \times a_{S,S^*} \times b_{S^*}(O_{t+1}) \times \beta[t+1,S^*]}$;
      end
      $\gamma[t, S] \leftarrow \sum_{\bar{S}=S_1}^{S_N} \xi[t, S, \bar{S}]$;
    end
  end
  $\bar{\pi}_S \leftarrow \gamma[1, S]$;
  $\bar{a}_{S,S^*} \leftarrow \frac{\sum_{t=1}^{T-1} \xi[t,S,\bar{S}]}{\sum_{t=1}^{T-1} \gamma[t,S]}$;
  $\bar{b}_S(k) \leftarrow \frac{\sum_{t=1}^{T} \gamma[t,S]}{\sum_{t=1}^{T} \gamma[t,S]}$;
  $\bar{\lambda} = \lambda(\bar{A}, \bar{B}, \bar{\Pi})$;
until *Convergence*;
Result: $\lambda(A, B, \Pi)$

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: $\sum_{m=1}^{M} c_{jm} = 1$ and $c_{jm} \geq 1$, where $1 \leq j \leq N$, $1 \leq m \leq M$. The parameters of the observation density function $(c_{jm}, \mu_{jm}, U_{jm})$ can be estimated through the modified Baum-Welch algorithm [20]. Using continuous density in HMM makes it more accurate; however, it requires a larger dataset and a more complex algorithm to train.

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 coeeficients a_{ii} 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].

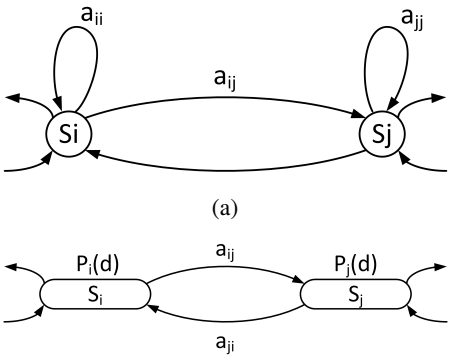

Figure 2: An illustration of interstate connections in HMMs. (a) represents a normal HMM with self transitions from each state back to itself. (b) represents a variable duration HMM with no self state transition and specified state duration densities.

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].

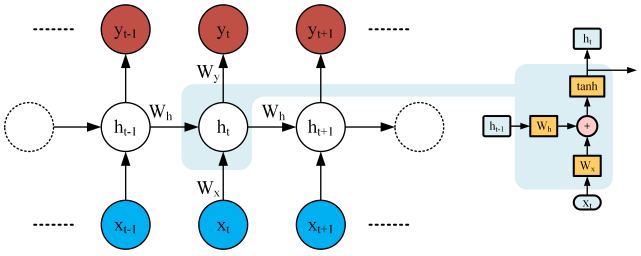

Figure 3: A simple RNN with a single hidden layer. At each time step t, output is produced through passing activations as in a feedforward network. Activations are passed to next node at time $t+1$ as well to achieve recurrence.

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: $h_t = \sigma_h\left(W_x x_t + W_h h_{t-1} + b_h\right)$, $y_t = \sigma_y\left(W_y h_t + b_y\right)$; 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 σ_h and σ_y which can be hyperbolic tangent function (tanh), sigmoid, or rectified linear unit (ReLU). In a simple RNN unit, tanh is usually used.

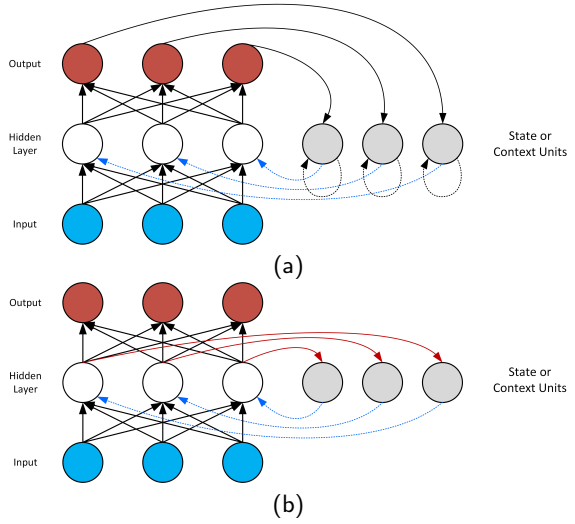

Figure 4: Early designs of RNNs. The dotted arrows represent the edges feeding at the next time step. (a) Jordan network. Output units are connected to context units that provide feedback at next time step to hidden units and themselves. (b) Elman network. Hidden units are connected into the context units that provide feedback to the hidden units only at the next time step.

3.1. Early RNN Architectures

Jordan [41] introduced an early form of recurrence in networks by adding extra "special" units called context or state units that feed values to the hidden units in the following time step. Basically, the network was as simple as a multilayer feed-forward network with the context units taking input from the network output at the current time step and feed them back to themselves and the hidden units at the next time step as shown in Fig. 4 (a). The context units allow the network to remember its outputs at previous time steps and being self connected enables sending information across time steps without intermediate output perturbation [18]. Elman [37] also introduced a simple architecture in which the context units are associated with each each hidden layer unit at the current time step and give feedback to the same hidden unit at the next time step as shown in Fig. 4 (b). This notation of self-connected hidden units became the basis for the work and design of long-short term memory (LSTM) units [19]. This type of recurrence has been demonstrated to learn time dependencies by Elman [37].

3.2. Training of RNNs

The expression of a generic RNN can be represented as $h_t = \mathcal{F}(h_{t-1}, x_t, \theta) = W_h \sigma_h(h_{t-1}) + W_x x_t + b_h$[1], where θ 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 $\varepsilon = \sum_{1 \le t \le T} \varepsilon_t$, where $\varepsilon_t = \mathcal{L}(h_t)$, T is the sequence length (total number of time steps), and $\mathcal{L}$ 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.

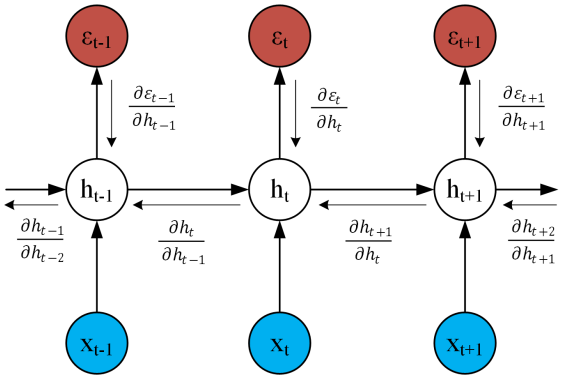

Figure 5: Unfolded recurrent neural network in time [42]. ε_t denotes the error calculated from the output, h_t represents the hidden state, and x_t represents the input at time t.

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

$$\frac{\partial \varepsilon}{\partial \theta} = \sum_{1 \le t \le T} \frac{\partial \varepsilon_t}{\partial \theta}$$

$$\frac{\partial \varepsilon_t}{\partial \theta} = \sum_{1 \le k \le t} \left(\frac{\partial \varepsilon_t}{\partial h_t} \times \frac{\partial h_t}{\partial h_k} \times \frac{\partial h_k}{\partial \theta} \right)$$

$$\frac{\partial h_t}{\partial h_k} = \prod_{t \ge i > k} \frac{\partial h_i}{\partial h_{i-1}} = \prod_{t \ge i > k} W_h^T \, diag\left(\sigma_h'\left(h_{i-1}\right)\right) \quad (1)$$

The effect that the network parameters (θ) at step k have over the cost at subsequent steps ($t > k$), can be measured through the temporal gradient component $\frac{\partial \varepsilon_t}{\partial h_t} \times \frac{\partial h_t}{\partial h_k} \times \frac{\partial h_k}{\partial \theta}$. 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 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

[1] This formulation doesn't contradict with the previously mentioned formulation $\left(h_t = \sigma_h\left(W_x x_t + W_h h_{t-1} + b_h\right)\right)$ and both have the same behavior [42].

of the solutions to such 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 a 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 past time step and the input from 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].

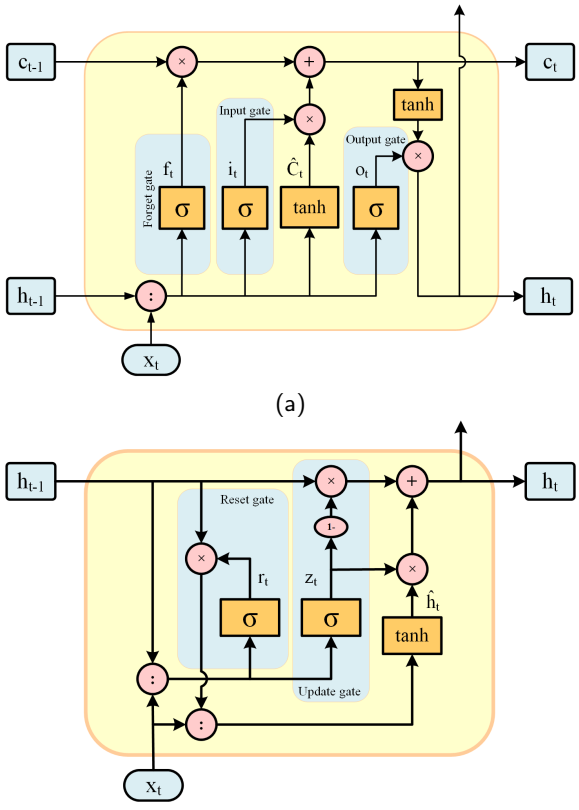

Figure 6: Current designs of RNNs. The symbols used in both diagrams are as follows, : represents concatenation, + represents element-wise summation, × represents element-wise multiplication, σ represents a sigmoid activation, and *tanh* represents a hyperbolic tangent (tanh) activation. (a) Schematic of an LSTM unit which is typically composed of three main parts, input, output, and forget gates. (b) Schematic of a GRU unit which is a simplified version of LSTM with only reset and update gates.

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 Electrocardiography

ECG is the graphical interpretation of skin-recorded electrical activity of the electric field originating in the heart [74]. ECG provides information that is not readily available through other methods about heart activity and is considered the most commonly used procedure in the diagnosis of cardiac diseases due to the fact that it is non-invasive, simple, and cost-effective. This makes ECG subject to intense research related to the automatic analysis in order to reduce the subjectivity and the time spent on interpreting hours of recordings [54, 75, 76]. ECG is a time periodic signal, which allows to mark out an elementary beat that constitutes the basis for ECG signal analysis [54]. For instance, heart rate can be estimated through the detection of QRS-complex from an ECG signal and the time interval between successive QRS-complexes (also known as R-R interval) can be used to detect premature ectopic beats [74]. In that sense, ECG beat detection is considered fundamental for most of the automated analysis algorithms. A detailed description of the recent publications that cover event detection in ECG using different methods, is included in Table 1.

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 a 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 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 in order 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 the neural activity of the brain's sensorimotor cortex in a way that is as observable as if the

Table 1
Summary of event detection work done in ECG event detection

Publication	Event under investigation	Implementation details	Dataset
Gersch 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 rhythms through the probability that the observed R-R symbol sequence is generated by any of a set of models generated from multiple cardiac arrhythmias. Theb manuscript used a maximum likelihood approach to determine the arrhythmia type.	Clinical test data from patients with atrial fibrillation (AF)
Coast et al. [52], 1990	Beat detection for arrhythmia analysis	A parallel combination of HMMs (one for each arrhythmia type), is used to classify arrhythmia. The classification process is inferred through determining the most likely path through the parallel models. All ECG 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 [53]
Andreao et al. [54], 2006	ECG beat detection and segmentation	An HMM was constructed for ECG beat with each waveform part represented in the model including the isoelectric parts (ISO, P, PQ, QRS, ST, T). Model parameters were estimated using Baum-Welch method and the number of states in each model were specified empirically to achieve a good complexity-performance compromise. The proposed segmentation in this study was based on a single channel but the authors provided insights about the possibility of adaptation with multi-channel fusion.	QT database [55]
Sandberg et al. [56], 2008	Atrial fibrillation frequency tracking	An HMM is used for frequency tracking to overcome the corruption of residual ECG by muscular activity or insufficient beat cancellation. States of the HMM were used to represent the underlying frequencies in short-time Fourier transform while observations corresponded to the estimated frequency of specific time intervals from the signal. Experiments were performed on single channel simulated signals with inclusion of mutli-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 segmentation (beat) of ECG and Phonocardiogram (PCG)	An ECG channel along with a phonocardiogram were fused in a single coupled HMM for beat detection. The coupled HMM was constructed to consider the high dynamics and non-stationarity of the signals where the channels were assumed to be co-dependent through past states and observations. Each of ECG and phonocardiogram was modeled using 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.	A self-recorded dataset from healthy male adults.
Übeyli [58], 2009	Arrhythmia detection/classification	An Elman-based RNN is used for beat classification with the Levenberg-Marquardat algorithm for training (a least-squares estimation algorithm based on the maximum neighborhood idea). This model used power spectral density (calculated with three different methods; Pisarenko, MUSIC, and Minimum-Norm) of ECG signals as input. All the models trained in this study, used feature-level fusion.	Four types of ECG beats obtained from Physiobank Database [59].
Zhang et al. [60], 2017	Supraventriular and verntricular ectopic beat detection (SVEB and VEB)	An LSTM-based RNN preceded by a density-based clustering for training data selection from a large data pool. In this implementation, 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. The RNN layers were followed by two fully connected layers in order to combine the temporal features and generate the desired output. This study only used a single channel ECG (limb lead II) with no multi-channel fusion.	MIT-BIH Arrhythmia database (MITDB) [61].
Xiong et al. [62], 2017	Atrial fibrillation automatic detection	A 3 layer RNN was implemented 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 [59].
Schwab et al. [49], 2017	Different cardiac arrhythmia classification/detection	In this work a combination of GRU and bidirectional LSTM (BLSTM) based RNNs and nonparameteric Hidden Semi-Markov Models (HSMM), was used for building the beat classification model and then a blender [63] was used to combine the predictions from the models. No multi-channel fusion was performed in this study and only a single ECG lead was employed.	The 2017 PhysioNet/CinC Challenge dataset [59].
Zihlmann et al. [64], 2017	Atrial fibrillation detection	A single layer LSTM-based convolutional RNN (CRNN) was constructed for atrial fibrillation detection in arbitrary length ECG recordings. This work employed the log spectrogram as an input to the CRNN to increase the accuracy. No multi-channel fusion was performed in this study and only a single ECG lead was used.	The 2017 PhysioNet/CinC Challenge dataset [59].
Limam and Precioso [65], 2017	Atrial fibrillation detection	A two layer LSTM-based CRNN 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 compared.	The 2017 PhysioNet/CinC Challenge dataset [59].
Chang et al. [66], 2018	Atrial fibrillation detection	A single layer LSTM-based RNN 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 spectrograms of multi-lead ECG before feeding into the LSTM units.	Multiple datasets for atrial fibrillation and normal sinus rhythms [59, 61, 67–70].
Lui and Chow [71], 2018	Myocardial infarction classification	A deep single-layer LSTM based CRNN 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-one 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. [73], 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 sizes 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 database (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 due to the fact that 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

Table 2
Summary of EEG-based sleep staging.

Publication	Event under investigation	Implementation details	Dataset
Flexerand et al. [79], 2002	Sleep staging in combined EEG and EMG	A three state (wakefulness, deep sleep, and rapid eye movement sleep) Gaussian observation HMM (GOHMM) was used and sleep stages were represented as mixtures of the basic three states. The probability of being in any of the three states was computed for 1 sec windows so that a continuous probability monitoring can be achieved. Expectation-maximization algorithm was used for parameter estimation and the Viterbi algorithm was used to calculate the posteriori estimate for being in each state. Feature-level fusion was performed on features from EEG channels (C3 and C4) and EMG.	Nine whole-night sleep recordings from a group of nine healthy adults.
Flexer et al. [80], 2005	Sleep staging in single channel EEG (C3)	A three state (wakefulness, deep sleep, and rapid eye movement sleep) Gaussian observation HMM (GOHMM) was used and sleep stages were represented as mixtures of the basic three states. The probability of being in any of the three states was computed for 1 sec windows so that a continuous probability monitoring can be achieved. Expectation-maximization algorithm was used for parameter estimation and the Viterbi algorithm was used to calculate the posteriori estimate for being in each state. No multi-channel fusion was performed in this study and only a single EEG channel was used.	Two datasets were used, the first consists of 40 whole night sleep recordings from healthy adults and the second consists of 28 whole night sleep recordings of healthy adults.
Doroshenkov et al. [81], 2007	Sleep staging using two channel EEG (Fpz-Cz and Pz-Oz)	A six state HMM was constructed for the purpose of sleep staging. Baum-welch algorithm was used for model's parameter estimation and the Viterby algorithm for state sequence decoding. Feature-level fusion was performed for features calculated from the two EEG channels.	Sleep-EDF database [82].
Bianchi et al. [83], 2012	Sleep cycle (quantifying probabilistic transitions between stages and multi-exponential dynamics) and fragmentation in case of apnea in PSG	An eight state HMM was constructed for sleep-wake activity. The connectivity between states was inferred through exponential fitting of subsets of the pooled bouts and adjacent-stage analysis.	Sleep Heart Health Study database [84].
Pan et al. [85], 2012	Sleep staging using central EEG (C3-A2), chin electromyography (EMG), and electrooculogram (EOG)	A six state transition-constrained discrete HMM was constructed for sleep staging. Thirteen features were utilized including temporal and spectrum analyses of the EEG, EOG and EMG signals with feature-level fusion employed.	PSG including six channel EEG, EOG, EMG, and ECG signals, was obtained from 20 healthy subjects.
Yaghouby and Sunderam [86], 2015	Sleep staging and scoring (quasi-supervised) in PSG	A five state Gaussian HMM was constructed for sleep staging with Baum-Welch algorithm for parameter estimation. In this implementation, feature-level fusion was achieved through feeding augmented vector of PSG features and human rated scores into the estimation algorithm in order to obtain the parameters to maximize the likelihood that a model with larger number of states explains the data.	Sleep-EDF database [82].
Onton et al. [87], 2016	Sleep staging in 2-channel home EEG (FP1-A2 and FP2-A2) and electrodermal activity (EDA)	A five state Gaussian HMM was constructed for sleep staging with expectation-maximization algorithm for parameter estimation and the Viterbi algorithm to find the maximum a posteriori estimate of state sequence. In this implementation, the relative power across the entire night was averaged in five frequency bands and fed into the model (feature-level fusion).	A self recorded data from 51 participants who were medication-free and self-reported asymptomatic sleepers and wit no history of neurologic or psychiatric disorders.
Davidson et al. [88], 2005	Behavioral microsleep detection in EEG (P3-01 and P4-02)	This study utilized an LSTM-based RNN to detect the lapses in visuomotor performance associated with behavioral microsleep events. The network used the power spectral density of 1 sec windows of the used two channels (calculated using the covariance method) with feature-level fusion in place to combine data. The network included 6 LSTM blocks of 3 memory cells each.	A self-recorded dataset from 15 subjects performing visuomotor tracking task.
Hsu et al. [89], 2013	Automatic deep sleep staging in single channel EEG (Fpz-Cz)	This study utilized an Elman recurrent neural network that works on the energy features extracted from a single channel EEG to perform 5-level sleep staging. No multi-channel fusion was employed in this study.	Sleep-EDF database [82].
Supratak et al. [90], 2017	Automatic sleep staging in single channel EEG (Fpz-Cz or Pz-Oz)	A convolutional RNN (CRNN) was constructed to work directly of the raw signal data. Two branches of CNN, each of 4 layers, were used for representation learning and their outputs were combined and fed into a two layer LSTM-based BRNN with skip branch to generate the sleep stage. No multi-channel fusion was employed in this study.	Montreal Archive of Sleep Studies (MASS) [91] and Sleep-EDF database [82].
Biswal et al. [92], 2017	Automatic sleep staging	Raw EEG signals were split into 30-seconds windows, then the spectrogram and expert defined features were extracted and fused at the feature-level. The best accuracy reported among different RNN architectures, was reported for a 5-layer LSTM-based RNN. This study presented also an LSTM-based CRNN architecture to extract spatial features automatically and then pass them to the RNN part for temporal context extraction.	10,000 PSG studies with multi-channel EEG data (F3, F4, C3, C4, O1 and O2 referenced to the contralateral mastoid, M1 or M2).
Phan et al. [93], 2018	Automatic deep sleep staging in single channel EEG (Fpz-Cz)	A two-layer GRU-based BRNN was constructed to learn temporal features from the single channel EEG. This implementation included an attention mechanism that was applied on the BRNN output features. The weighted output was then used to feed a linear SVM classifier. No multi-channel fusion has been employed in this study.	Sleep-EDF database [82].
Bresch et al. [94], 2018	Sleep staging in single-channel EEG	An LSTM-based CRNN with 3 CNN layers and 3 LSTM layers, was built to process 30-seconds windows of raw EEG data (FPz, left EOG, and right EOG referenced to M2). No multi-channel fusion has been employed in this study.	The SIESTA database [95] and a self-recorded dataset with 147 recordings from 29 healthy subjects.
Phan et al. [96], 2019	Automatic sleep staging	This study featured multi-modality fusion on the feature level between EEG, EOG, and EMG. All were split into windows and converted into time-frequency representation using filter banks. The fused data were fed into a BRNN that is used to encode the features, then the output is passed through an attention layer followed by another BRNN that performs the cclassification of the sleep stage.	Montreal Archive of Sleep Studies (MASS) Dataset [91].
Michielli et al. [97], 2019	Automatic sleep staging in single channel EEG	A dual branch LSTM-based RNN was constructed for the classification of 5 different sleep stages. the network starts with a preprocessing and feature extraction stages and then the data is distributed over two branches. The first branch uses mRMR for feature selection followed by a one layer LSTM and fully connected layer to classify between 4 classes only (W, N1-REM, N2 and N3). The second branch uses PCA for feature selection followed by a 2 layer LSTM and a fully connected layer for binary classification. The LSTM in the second branch takes the classification output from the first branch to consider only the combined stage N1-REM for separation. No multi-channel fusion has been employed in this study.	Sleep-EDF database [59].
Sun et al. [98], 2019	Sleep staging in single channel EEG	A two stage network was built to perform the classification. The first stage is time distributed stage that included two parallel branches, the first included a window deep belief network for feature extraction followed by a dense layer and a second branch with hand-crafted features extraction then a dense layer. The two branches were then fused through another dense layer and fed as an input to an LSTM-based BRNN (the second stage) to generate the classes.	Sleep-EDF database [59].

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 schemes have been introduced to solve the 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 state 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 trinary observation sequence. The trinary vector is used to train the HMM using Baum-Welch which is then used to Viterbi decode the observation sequences into the hidden states sequence. A hypothesis test that a statistical association exists between the detected and seizure states, is performed through 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 20-36 surgically implanted electrodes on the brain or brain substance [113].
Santaniello et al. [106], 2011	Early detection of seizures in iEEG from a rat 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 peri-ictal states. Baum-Wlech algorithm was used for parameter estimation and a Bayesian evolution model was used determine the time of state transition.	Data collected from male Sprague-Dawley rats with four implanted skull screw EEG electrodes placed bifrontally and posteriorly behind bregma and a fifth depth electrode placed in hippocampus, were collected and used for this study.
Direito et al. [114], 2012	Identification of the 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 overtime to train a 4 state (preictal, ictal, postictal and interictal) HMM. The Baum–Welch algorithm was used to train the model and the Viterbi algorithm to decode the state-sequence.	EPILEPSIAE database [115].
Abdullah et al. [104], 2012	Seizure detection in iEEG	A three state discrete HMM was built to classify iEEG 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.	Freiburg Seizure Prediction EEG (FSPEEG) database [108].
Smart and Chen [105], 2015	Seizure detection in scalp EEG	This study used a 5 sec sliding window with 1 sec increments to process the EEG signals. A set of 45 measurements was 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 state (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.	CHB-MIT Scalp EEG Database [116].
Petrosian et al. [117], 2000	Onset detection of epileptic seizures in both scalp and intracranial EEG	Both raw EEG data and their wavelet transform "daub4" were used in training an Elman RNN. This study used a feature-level fusion model for multi-channel EEG data to provide an input for the RNN.	Scalp and iEEG data were collected from two patients who were undergoing long-term electrophysiological monitoring for epilepsy.
Güler et al. [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	Lyapunov exponents of the EEG signals were used to train an Elman RNN for the identification task. This study used a feature-level fusion model for multi-channel EEG data to train the RNN.	Publicly available epilepsy dataset by University of Bonn [119].
Kumar et al. [120], 2008	Automatic detection of epileptic seizure 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 dataset by University of Bonn [119].
Minasyan et al. [121], 2010	Automatic detection of epileptic seizures prior to or immediately after clinical onset in scalp EEG	A set of time domain, spectral domain, wavelet domain, and information theoretic features were used to train an ELman RNN per each channel of the EEG and the output is combined in time and space through a decision making module that performs a decision-level fusion in order to declare a seizure event if N out of M channels declared it.	EEG dataset from 25 patients hospitalized for long-term EEG monitoring in five centers including Thomas Jefferson University, Dartmouth University, University of Virginia, UCLA and University of Michigan medical centers.
Naderi and Mahdavi-Nasab [122], 2010	Automatic detection of epileptic seizure 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 dataset by University of Bonn [119].
Vidyaratne et al. [123], 2016	Automated patient specific seizure detection using scalp EEG	The preprocessed (denoised) 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).	CHB-MIT Scalp EEG Database [116].
Talathi [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, inter-ictal, 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 dataset by University of Bonn [119].
Golmohammadi et al. [125], 2017	Epileptic seizure detection	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 devloped as well for comparison.	A subset of the TUH EEG Corpus (TUEEG) [126] that has been manually annotated for seizure events [127].
Raghu et al. [128], 2017	Epileptic seizures classification	This 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 dataset by University of Bonn [119].
Abdelhameed et al. [129], 2018	Epileptic seizure detection	This study used raw EEG signals to feed a 1D CRNN that is based on bidirectional LSTM to classify EEG segments into one of two states (normal-ictal and normal-ictal-interictal).	Publicly available epilepsy dataset by University of Bonn [119].
Daoud and Bayoumi [130], 2018	Epileptic seizure 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, 116].
Hussein et al. [131], 2019	Epileptic seizures detection	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 dataset 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

Table 4
Summary of EEG-based BCI systems.

Publication	Event under investigation	Implementation details	Dataset
Obermaier 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 combined 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.
Obermaier 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 two channels (C3 and C4) were fused and fed into the HMM models to calculate the single best path probabilities 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.
Pfurtscheller et al. [140], 2003	Two class motor imagery BCI for virtual keyboard control	Two HMMs, one 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.
Solhjoo et al. [141], 2005	EEG-based mental task classification (left or right hand movement)	Discrete 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].
Suk and Lee [143], 2010	Multi-class motor imagery classification	In this study, dynamic patterns in EEG signals were modeled using two layers HMM. First time-domain patterns were extracted from the signals and have dimension reduced using PCA. Second, the likelihood for each channel is computed in the first layer of HMM and assembled in vector whose dimension is reduced with PCA as well. finally, the class label is calculated through the largest likelihood in the upper layer of HMM. Baum-Welch algorithm was used to estimate the parameters of the initial state distribution, the state transition probability distribution, and the observation probability distribution and Viterbi algorithm was used for decoding the state sequence.	Dataset IIa of BCI Competition IV (2008) provided by the BCI research group at Graz University [144].
Speier et al. [145], 2014	P300 speller	An HMM was used to model typing as a sequential process where each character selection is influenced by previous selections. The Viterbi algorithm was used to decode the optimal sequence of target characters.	Data were collected from 15 healthy graduate students and faculty with normal or corrected to normal vision between the ages of 20 and 35.
Erfanian and Mahmoudi [146], 2005	Real-time adaptive noise canceler for ocular artifact suppression in EEG	A recurrent multi-layer perceptron 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 [147], 2011	EEG signal forecasting and mental tasks classification	An Elman RNN was trained for forecasting EEG a single time step ahead then an Elman RNN-based classifier was trained to classify the mental task associated with the EEG signals.	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 backward from 100 by 3's, and singing a favorite song.
Balderas et al. [148], 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 [149]
Maddula et al. [150], 2017	P300 BCI classification	A 3D CNN in conjunction with a 2D CNN were combined with an LSTM-based RNN to capture spatio-temporal patterns in EEG.	Data from P300 segment speller were collected, where the subjects mentally noted whenever the flashed letter is part of their target [151].
Thomas et al. [152], 2017	Steady-state visual evoked potential (SSVEP)-based BCI classification	A single layer BRNN was used to perform classification and compared to different architecture and traditional classifying techniques.	5-class SSVEP dataset [153].
Spampinato et al. [154], 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 tat generated the stimuli. The authors here tested different architectures for the encoder including a common LSTM for all channels, channel LSTMs + common LSTM, and Common LSTM + fully connected layer. The authors also trained a CNN-based regressor for generating the EEG features to replace the whole EEG module and work only using source images of visual stimuli.	A subset of ImageNet dataset (40 classes) [155] was used to generate visual stimuli for six subjects while EEG data is recorded.
Hosman et al. [156], 2019	Intercortical BCI for cursor control	An single layer LSTM-based decoder was built with three outputs to generate the cursor speed in x and y directions in addition to the distanc to target.	Intercortical neural signals recorded from three participants, each with 2 96-channel micro-electrode arrays [157].
Zhang et al. [158], 2020	EEG-Based Human Intention Recognition	In this study, multi-channel raw EEG sequences into mesh-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 CRNN that starts with a 2D CNN that processes the meshes 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 investigated was composed of two parallel branches the first was 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. This study used fusion on both data-level and feature-level.	EEG Motor Movement/Imagery Dataset [59, 159].
Tortora et al. [160], 2020	BCI for gait decoding from EEG	EEG data were preprocessed to remove motion artifacts through high pass filtration and independent component analysis. Different frequency bands were then extracted and a separate classifier is trained based on each frequency band. The classifiers were based on a two-layer LSTM-based RNN followed by a fully connected layer, a softmax layer, and an output layer that manifests the prediction output.	EEG data were recorded from 11 subjects walking on a treadmill using a 64-channel amplifier and 10/20 montage.

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 5
Summary of event detection in EMG signals.

Publication	Event under investigation	Implementation details	Dataset
Chan and Englehart [165], 2005	Continuous identification of six classes hand movement in sEMG	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 autoregressive coefficients were computed as features.	4-channel sEMG collected from the forearm of 11 subjects for six distinct motions (wrist flexion, wrist extension, supination, pronation, hand open, and hand close) [166].
Zhang et al. [167], 2011	Hand gesture recognition in acceleration and sEMG	In this work, the authors actually identified the active segments via processing and thresholding of the average signal of the multichannel sEMG. The onset is when the energy is higher than a certain 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 sEMG, and fed to five-state 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-level fusion) through four layers of classifiers with the last layer as the HMM.	sEMG and 3d acceleration were collected from two right-handed subjects who performed 72 Chinese sign language words in a sequence with 12 repetitions per motion, and a predefined 40 sentences with 2 repetitions per sentence.
Wheeler et al. [168], 2006	Hand gesture recognition in sEMG	Moving average was used on the sEMG signals to provide the input for continuous left-to-right HMMs with tied Gaussian mixtures. The training was performed using the Baum-Welch algorithm and the real-time recall was performed with The Viterbi algorithm. The models were also initialized using K-means clustering so that the states were partitioned to equalize the amount of variance within each state. This study employed feature-level fusion to combine multi-channel data.	Data from one participant repeating 4 gestures on a joystick (left, right, up, and down) for 50 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.
Monsifrot et al. [169], 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 spike trains embedded in noise. A Markov model of sparse signals 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 impulse responses of the model.	The method introduced was tested over both simulated and experimental iEMG signals. the simulated signals were generated via Markov model under 10 kHz sampling frequency and with filter shapes obtained from experimental iEMG for more realistic simulation. The experimental iEMG signals were acquired from the extensor digitorum of a healthy subject with teflon coated stainless steel wire electrodes.
Lee [170], 2008	sEMG-based speech recognition	A continuous HMM was constructed with Gaussian mixtures model adopted for sEMG-based word recognition based on log mel-filter bank spectrogram 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 k^{th} word are estimated from the observations of the corresponding state of the same word. Viterbi algorithm was used for the decoding process.	EMG signals were collected from articulatory facial muscles from 8 Korean male subjects. The subjects were asked to pronounce each word from 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	sEMG-based automatic speech recognition	A six state left-right HMM with single mixture observation densities, was constructed for identifying the words based on three features extracted from sEMG that included the first two autoregressive coefficients and the integrated absolute value. HMM was trained in this work using the expectation-maximization algorithm.	sEMG from five articulatory facial muscles were collected. The dataset used here was a subset of the dataset described in [172] with ten-English word vocabulary.
Li et al. [173], 2014	Identification/prediction of functional electrical stimulation (FES)-induced muscular dynamics with evoked EMG (eEMG)	A nonlinear ARX-type RNN was used to predict the stimulated muscular torque and track muscle fatigue. The model takes the eEMG as an input and produces the predicted torque.	The experiments were conducted on 5 subjects with spinal cord injuries.
Xia et al. [174], 2018	Hand motion estimation from sEMG	A CRNN with 3 CNN layers and 2 LSTM layers was used for the prediction and the model used the power spectral density as input.	sEMG signals were collected 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 objective for this system.
Quivira et al. [175], 2018	Simple hand finger movement identification in sEMG	An LSTM-based RNN was used to implement a recurrent mixture density network (RMDN) [176] that probabilistically model the output of the Network in order to capture the complex features present the hand movement.	8 channel EMG signals were collected from the proximal forearm region, targeting most muscles used in hand manipulation. The hand pose tracking was performed with a Leap Motion sensor and the subjects were asked to perform 7 hand gestures with repetitions per gesture.
Hu et al. [177], 2018	Hand gesture recognition in sEMG	sEMG signals from all channels were segmented into windows of fixed size 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.	Experiments were performed over the first and second sub-databases of NinaPro (Non Invasive Adaptive Prosthetics) database [178].
Samadani [179], 2018	EMG-Based Hand Gesture Classification	Different RNN architectures were tested in this study to chose the best performing architecture. The evaluated models included uni and bidirectional LSTM- and GRU-based RNNs with attention mechanisms. The models worked on the preprocessed (denoised) raw EMG signals.	Publicly-available NinaPro hand gesture dataset (NinaPro2) was used [180].
Simão et al. [181], 2019	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 layer and another dense layer followed by the output layer. This model was compared to a similar GRU-based model and another static feed forward neural network model. This study used combined feature vector as an input for the models.	the synthetic sequences of the UC2018 DualMyo dataset [182] and a similar subset of the NinaPro DB5 dataset [183]

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.

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 addresses 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 opening of upper esophageal sphincter 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 due to the fact that 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 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 neron 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 hand writing 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 wide spread 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) 983–983.

[3] P. Rashidi, A. Mihailidis, 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.2009.5333874.

[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. Alinia, 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. Kacprzyk, E. Oja, S. Zadrożny, 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. Eswaran, 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. Schlogl, 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) L14–L22.

[14] Y. Khalifa, J. L. Coyle, E. Sejdić, 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. Sejdić, 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. Juang, 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 sequential machine, in: J. Diederich (Ed.), Artificial Neural Networks, IEEE Press, Piscataway, NJ, USA, 1990, pp. 112–127.

[39] H. Jaeger, The "echo state" approach to analysing and training recurrent neural networks-with an erratum note, Technical Report, German National Research Center for Information Technology, 2001.

[40] Y. Khalifa, Z. Zhang, E. Sejdić, Sparse recovery of time-frequency representations via recurrent neural networks, in: Proceedings of the 22nd International Conference on Digital Signal Processing, ACM, 2017, pp. 1–5.

[41] M. I. Jordan, Serial order: A parallel distributed processing approach, in: Neural Network Models of Cognition, volume 121 of *Advances in Psychology*, North-Holland, 1997, pp. 471–495.

[42] R. Pascanu, T. Mikolov, Y. Bengio, On the difficulty of training recurrent neural networks, in: Proceedings of the 30th International Conference on Machine Learning, volume 28, 2013, pp. III–1310–III–1318.

[43] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE Transactions on Neural Networks 5 (1994) 157–166.

[44] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, J. Schmidhuber, A novel connectionist system for unconstrained handwriting recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2009) 855–868.

[45] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Y. W. Teh, M. Titterington (Eds.), Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, volume 9 of *Proceedings of Machine Learning Research*, PMLR, 2010, pp. 249–256.

[46] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder–decoder for statistical machine translation, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1724–1734.

[47] F. A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: Continual prediction with LSTM, in: Proceedings of the 9th International Conference on Artificial Neural Networks, volume 2, IEEE, 1999, pp. 850–855. doi:10.1049/cp:19991218.

[48] A. Viterbi, Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, IEEE Transactions on Information Theory 13 (1967) 260–269.

[49] P. Schwab, G. C. Scebba, J. Zhang, M. Delai, W. Karlen, Beat by beat: Classifying cardiac arrhythmias with recurrent neural networks, in: Computing in Cardiology, volume 44, 2017, pp. 1–4.

[50] M. F. Stollenga, W. Byeon, M. Liwicki, J. Schmidhuber, Parallel multi-dimensional LSTM, with application to fast biomedical volumetric image segmentation, arXiv preprint arXiv:1506.07452 (2015).

[51] W. Gersch, P. Lilly, E. Dong, PVC detection by the heart-beat interval data—Markov chain approach, Computers and Biomedical Research 8 (1975) 370 – 378.

[52] D. A. Coast, R. M. Stern, G. G. Cano, S. A. Briller, An approach to cardiac arrhythmia analysis using hidden Markov models, IEEE Transactions on Biomedical Engineering 37 (1990) 826–836.

[53] R. E. Hermes, D. B. Geselowitz, G. Oliver, Development, distribution, and use of the American Heart Association database for ventricular arrhythmia detector evaluation, Computers in Cardiology (1980) 263–266.

[54] R. V. Andreao, B. Dorizzi, J. Boudy, ECG signal analysis through hidden Markov models, IEEE Transactions on Biomedical Engineering 53 (2006) 1541–1549.

[55] P. Laguna, R. G. Mark, A. Goldberg, G. B. Moody, A database for evaluation of algorithms for measurement of QT and other waveform intervals in the ECG, in: Computers in Cardiology, 1997, pp. 673–676.

[56] F. Sandberg, M. Stridh, L. Sornmo, Frequency tracking of atrial fibrillation using hidden Markov models, IEEE Transactions on Biomedical Engineering 55 (2008) 502–511.

[57] J. Oliveira, C. Sousa, M. T. Coimbra, Coupled hidden Markov model for automatic ECG and PCG segmentation, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2017, pp. 1023–1027.

[58] E. D. Übeyli, Combining recurrent neural networks with eigenvector methods for classification of ECG beats, Digital Signal Processing 19 (2009) 320–329.

[59] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, H. E. Stanley, PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals, Circulation 101 (2000) E215–E220.

[60] C. Zhang, G. Wang, J. Zhao, P. Gao, J. Lin, H. Yang, Patient-specific ECG classification based on recurrent neural networks and clustering technique, in: Proceedings of the 13th International Conference on Biomedical Engineering, 2017, pp. 63–67.

[61] G. B. Moody, R. G. Mark, The impact of the MIT-BIH arrhythmia database, IEEE Engineering in Medicine and Biology Magazine 20 (2001) 45–50.

[62] Z. Xiong, M. K. Stiles, J. Zhao, Robust ECG signal classification for detection of atrial fibrillation using a novel neural network, in: Computing in Cardiology, volume 44, 2017, pp. 1–4.

[63] D. H. Wolpert, Stacked generalization, Neural Networks 5 (1992) 241–259.

[64] M. Zihlmann, D. Perekrestenko, M. Tschannen, Convolutional recurrent neural networks for electrocardiogram classification, in: Computing in Cardiology, 2017, pp. 1–4.

[65] M. Limam, F. Precioso, Atrial fibrillation detection and ECG classification based on convolutional recurrent neural network, in: Computing in Cardiology, 2017, pp. 1–4. doi:10.22489/CinC.2017.171-325.

[66] Y. Chang, S. Wu, L. Tseng, H. Chao, C. Ko, AF detection by exploiting the spectral and temporal characteristics of ECG signals with the LSTM model, in: Computing in Cardiology, volume 45, 2018, pp. 1–4.

[67] S. Petrutiu, A. V. Sahakian, S. Swiryn, Abrupt changes in fibrillatory wave characteristics at the termination of paroxysmal atrial fibrillation in humans, EP Europace 9 (2007) 466–470.

[68] A. Taddei, G. Distante, M. Emdin, P. Pisani, G. B. Moody, C. Zeelenberg, C. Marchesi, The European ST-T database: standard for evaluating systems for the analysis of ST-T changes in ambulatory electrocardiography, European Heart Journal 13 (1992) 1164–1172.

[69] F. M. Nolle, F. K. Badura, J. M. Catlett, R. W. Bowser, M. H. Sketch, CREI-GARD, a new concept in computerized arrhythmia monitoring systems, Computers in Cardiology 13 (1987) 515–518.

[70] R. Bousseljot, D. Kreiseler, A. Schnabel, Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet, Biomedizinische Technik/Biomedical Engineering 40 (1995) 317–318.

[71] H. W. Lui, K. L. Chow, Multiclass classification of myocardial infarction with convolutional and recurrent neural networks for portable ECG devices, Informatics in Medicine Unlocked 13 (2018) 26–33.

[72] G. D. Clifford, C. Liu, B. Moody, L. H. Lehman, I. Silva, Q. Li, A. E. Johnson, R. G. Mark, AF classification from a short single lead ECG recording: The PhysioNet/computing in cardiology challenge 2017, 2017, pp. 1–4. doi:10.22489/CinC.2017.065-469.

[73] S. Singh, S. K. Pandey, U. Pawar, R. R. Janghel, Classification of ECG Arrhythmia using Recurrent Neural Networks, Procedia Computer Science 132 (2018) 1290–1297.

[74] A. Kadish, A. E. Buxton, H. Kennedy, B. P. Knight, J. W. Mason, C. Schuger, C. Tracy, W. L. Winters, A. W. Boone, M. Elnicki, J. W. Hirshfeld, B. H. Lorell, G. Rodgers, H. H. Weitz, ACC/AHA clinical competence statement on electrocardiography and ambulatory electrocardiography, Journal of the American College of Cardiology 38 (2001) 3169–3178.

[75] M. H Crawford, S. Bernstein, P. Deedwania, J. Dimarco, K. J Ferrick, A. Garson, L. Green, H. Leon Greene, M. Silka, P. H Stone, C. Tracy, R. Gibbons, ACC/AHA guidelines for ambulatory electrocardiography, Journal of the American College of Cardiology 34 (1999) 912–948.

[76] K. S. Sayed, A. F. Khalaf, Y. M. Kadah, Arrhythmia classification based on novel distance series transform of phase space trajectories, in: Proceedings of the 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2015, pp. 5195–

5198.
[77] D. L. Schomer, F. L. Da Silva, Niedermeyer's electroencephalography: basic principles, clinical applications, and related fields, 6th ed., Lippincott Williams \& Wilkins, 2012.
[78] D. P. Subha, P. K. Joseph, U. R. Acharya, C. M. Lim, EEG signal analysis: A survey, Journal of Medical Systems 34 (2010) 195–212.
[79] A. Flexerand, G. Dorffner, P. Sykacekand, I. Rezek, An automatic, continuous and probabilistic sleep stager based on a hidden markov model, Applied Artificial Intelligence 16 (2002) 199–207.
[80] A. Flexer, G. Gruber, G. Dorffner, A reliable probabilistic sleep stager based on a single EEG signal, Artificial Intelligence in Medicine 33 (2005) 199–207.
[81] L. G. Doroshenkov, V. A. Konyshev, S. V. Selishchev, Classification of human sleep stages based on EEG processing using hidden Markov models, Biomedical Engineering 41 (2007) 25–28.
[82] B. Kemp, A. H. Zwinderman, B. Tuk, H. A. Kamphuisen, J. J. Oberye, Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG, IEEE Transactions on Biomedical Engineering 47 (2000) 1185–1194.
[83] M. T. Bianchi, N. A. Eiseman, S. S. Cash, J. Mietus, C. K. Peng, R. J. Thomas, Probabilistic sleep architecture models in patients with and without sleep apnea, Journal of Sleep Research 21 (2012) 330–341.
[84] S. F. Quan, B. V. Howard, C. Iber, J. P. Kiley, F. J. Nieto, G. T. O'Connor, D. M. Rapoport, S. Redline, J. Robbins, J. M. Samet, P. W. Wahl, The sleep heart health study: Design, rationale, and methods, Sleep 20 (1997) 1077–1085.
[85] S. T. Pan, C. E. Kuo, J. H. Zeng, S. F. Liang, A transition-constrained discrete hidden Markov model for automatic sleep staging, Biomedical Engineering Online 11 (2012) 52.
[86] F. Yaghouby, S. Sunderam, Quasi-supervised scoring of human sleep in polysomnograms using augmented input variables, Computers in Biology and Medicine 59 (2015) 54–63.
[87] J. A. Onton, D. Y. Kang, T. P. Coleman, Visualization of whole-night sleep EEG from 2-channel mobile recording device reveals distinct deep sleep stages with differential electrodermal activity, Frontiers in Human Neuroscience 10 (2016) 605.
[88] P. R. Davidson, R. D. Jones, M. T. R. Peiris, Detecting behavioral microsleeps using EEG and LSTM recurrent neural networks, in: Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE, 2005, pp. 5754–5757.
[89] Y. L. Hsu, Y. T. Yang, J. S. Wang, C. Y. Hsu, Automatic sleep stage recurrent neural classifier using energy features of EEG signals, Neurocomputing 104 (2013) 105–114.
[90] A. Supratak, H. Dong, C. Wu, Y. Guo, DeepSleepNet: A model for automatic sleep stage scoring based on raw single-channel EEG, IEEE Transactions on Neural Systems and Rehabilitation Engineering 25 (2017) 1998–2008.
[91] C. O'Reilly, N. Gosselin, J. Carrier, T. Nielsen, Montreal archive of sleep studies: An open-access resource for instrument benchmarking and exploratory research, Journal of Sleep Research 23 (2014) 628–635.
[92] S. Biswal, J. Kulas, H. Sun, B. Goparaju, M. B. Westover, M. T. Bianchi, J. Sun, SLEEPNET: Automated Sleep Staging System via Deep Learning, arXiv preprint arXiv:1707.08262 (2017).
[93] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, M. D. Vos, Automatic sleep stage classification using single-channel EEG: Learning sequential features with attention-based recurrent neural networks, in: Proceedings of the 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2018, pp. 1452–1455.
[94] E. Bresch, U. Großekathöfer, G. Garcia-Molina, Recurrent Deep Neural Networks for Real-Time Sleep Stage Classification From Single Channel EEG, Frontiers in Computational Neuroscience 12 (2018) 85.
[95] G. Klosh, B. Kemp, T. Penzel, A. Schlogl, P. Rappelsberger, E. Trenker, G. Gruber, J. Zeithofer, B. Saletu, W. M. Herrmann, S. L. Himanen, D. Kunz, M. J. Barbanoj, J. Roschke, A. Varri, G. Dorffner, The SIESTA project polygraphic and clinical database, IEEE Engineering in Medicine and Biology Magazine 20 (2001) 51–57.
[96] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, M. De Vos, SeqSleepNet: End-to-End Hierarchical Recurrent Neural Network for Sequence-to-Sequence Automatic Sleep Staging, IEEE Transactions on Neural Systems and Rehabilitation Engineering 27 (2019) 400–410.
[97] N. Michielli, U. R. Acharya, F. Molinari, Cascaded LSTM recurrent neural network for automated sleep stage classification using single-channel EEG signals, Computers in Biology and Medicine 106 (2019) 71–81.
[98] C. Sun, J. Fan, C. Chen, W. Li, W. Chen, A Two-Stage Neural Network for Sleep Stage Classification Based on Feature Learning, Sequence Learning, and Data Augmentation, IEEE Access 7 (2019) 109386–109397.
[99] S. H. Sheldon, R. Ferber, M. H. Kryger, Principles and practice of pediatric sleep medicine, 1st ed., Elsevier Health Sciences, 2005.
[100] D. Y. Kang, P. N. DeYoung, A. Malhotra, R. L. Owens, T. P. Coleman, A state space and density estimation framework for sleep staging in obstructive sleep apnea, IEEE Transactions on Biomedical Engineering 65 (2018) 1201–1212.
[101] A. Roebuck, V. Monasterio, E. Gederi, M. Osipov, J. Behar, A. Malhotra, T. Penzel, G. D. Clifford, A review of signals used in sleep analysis, Physiological Measurement 35 (2013) R1–R57.
[102] C. on Epidemiology and Prognosis, I. L. A. Epilepsy, Guidelines for epidemiologic studies on epilepsy, Epilepsia 34 (1993) 592–596.
[103] W. W. Lytton, Computer modelling of epilepsy, Nature Reviews: Neuroscience 9 (2008) 626–637.
[104] M. H. Abdullah, J. M. Abdullah, M. Z. Abdullah, Seizure detection by means of hidden Markov model and stationary wavelet transform of electroencephalograph signals, in: Proceedings of the IEEE-EMBS International Conference on Biomedical and Health Informatics, IEEE, 2012, pp. 62–65. doi:10.1109/BHI.2012.6211506.
[105] O. Smart, M. Chen, Semi-automated patient-specific scalp EEG seizure detection with unsupervised machine learning, in: Proceedings of the IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, 2015, pp. 1–7.
[106] S. Santaniello, D. L. Sherman, M. A. Mirski, N. V. Thakor, S. V. Sarma, A Bayesian framework for analyzing iEEG data from a rat model of epilepsy, in: Proceedings of the 33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2011, pp. 1435–1438.
[107] F. Mormann, T. Kreuz, C. Rieke, R. G. Andrzejak, A. Kraskov, P. David, C. E. Elger, K. Lehnertz, On the predictability of epileptic seizures, Clinical Neurophysiology 116 (2005) 569–587.
[108] T. Maiwald, M. Winterhalder, R. Aschenbrenner-Scheibe, H. U. Voss, A. Schulze-Bonhage, J. Timmer, Comparison of three nonlinear seizure prediction methods by means of the seizure prediction characteristic, Physica D-Nonlinear Phenomena 194 (2004) 357–368.
[109] P. E. McSharry, L. A. Smith, L. Tarassenko, Prediction of epileptic seizures: Are nonlinear methods relevant?, Nature Medicine 9 (2003) 241–242.
[110] Y. C. Lai, M. A. Harrison, M. G. Frei, I. Osorio, Controlled test for predictive power of Lyapunov exponents: their inability to predict epileptic seizures, Chaos 14 (2004) 630–642.
[111] M. Winterhalder, T. Maiwald, H. U. Voss, R. Aschenbrenner-Scheibe, J. Timmer, A. Schulze-Bonhage, The seizure prediction characteristic: A general framework to assess and compare seizure prediction methods, Epilepsy & Behavior 4 (2003) 318–325.
[112] S. Wong, A. B. Gardner, A. M. Krieger, B. Litt, A stochastic framework for evaluating seizure prediction algorithms using hidden Markov models, Journal of Neurophysiology 97 (2007) 2525–2532.
[113] A. B. Gardner, A. M. Krieger, G. Vachtsevanos, B. Litt, One-class novelty detection for seizure analysis from intracranial EEG, Journal of Machine Learning Research 7 (2006) 1025–1044.
[114] B. Direito, C. Teixeira, B. Ribeiro, M. Castelo-Branco, F. Sales, A. Dourado, Modeling epileptic brain states using EEG spectral analysis and topographic mapping, Journal of Neuroscience Methods 210 (2012) 220–229.

[115] M. Ihle, H. Feldwisch-Drentrup, C. A. Teixeira, A. Witon, B. Schelter, J. Timmer, A. Schulze-Bonhage, EPILEPSIAE - 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. Petrosian, 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. Sriraam, P. G. Benakop, Automated detection of epileptic seizures using wavelet entropy feature with recurrent neural network classifier, in: Proceedings of the IEEE Region 10 International 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 Neural Networks for seizure detection and early seizure detection systems, arXiv preprint arXiv:1706.03283 (2017).

[125] M. Golmohammadi, S. Ziyabari, V. Shah, E. Von Weltin, 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.2017.8257020.

[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, G. P. 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.2018.8585542.

[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 prosthesis: 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. Solhjoo, A. M. Nasrabadi, 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. Schlögl, 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 IEEE International Joint 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. Maszczyk, 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. Chatzilari, K. Adam,

S. Nikolopoulos, I. Kompatsiaris, Comparative evaluation of state-of-the-art algorithms for SSVEP-based BCIs, arXiv preprint arXiv:1602.00904 (2016).
[154] C. Spampinato, S. Palazzo, I. Kavasidis, D. Giordano, N. Souly, M. Shah, Deep learning human mind for automated visual classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6809–6817.
[155] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. H. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge, International Journal of Computer Vision 115 (2015) 211–252.
[156] T. Hosman, M. Vilela, D. Milstein, J. N. Kelemen, D. M. Brandman, L. R. Hochberg, J. D. Simeral, BCI decoder performance comparison of an LSTM recurrent neural network and a Kalman filter in retrospective simulation, in: Proceedings of the 2019 9th International IEEE/EMBS Conference on Neural Engineering (NER), 2019, pp. 1066–1071. doi:10.1109/NER.2019.8717140.
[157] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, A. H. Caplan, A. Branner, D. Chen, R. D. Penn, J. P. Donoghue, Neuronal ensemble control of prosthetic devices by a human with tetraplegia, Nature 442 (2006) 164–171.
[158] D. Zhang, L. Yao, K. Chen, S. Wang, X. Chang, Y. Liu, Making Sense of Spatio-Temporal Preserving Representations for EEG-Based Human Intention Recognition, IEEE Transactions on Cybernetics 50 (2020) 3033–3044.
[159] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, J. R. Wolpaw, BCI2000: a general-purpose brain-computer interface (BCI) system, IEEE Transactions on Biomedical Engineering 51 (2004) 1034–1043.
[160] S. Tortora, S. Ghidoni, C. Chisari, S. Micera, F. Artoni, Deep learning-based BCI for gait decoding from EEG with LSTM recurrent neural network, Journal of Neural Engineering 17 (2020) 046011.
[161] M. J. Zwarts, D. F. Stegeman, Multichannel surface EMG: Basic aspects and clinical utility, Muscle & Nerve 28 (2003) 1–17.
[162] J. Y. Hogrel, Clinical applications of surface electromyography in neuromuscular disorders, Neurophysiologie Clinique 35 (2005) 59–71.
[163] M. A. Oskoei, H. S. Hu, Myoelectric control systems-A survey, Biomedical Signal Processing and Control 2 (2007) 275–294.
[164] K. Englehart, B. Hudgins, P. A. Parker, A wavelet-based continuous classification scheme for multifunction myoelectric control, IEEE Transactions on Biomedical Engineering 48 (2001) 302–311.
[165] A. D. Chan, K. B. Englehart, Continuous myoelectric control for powered prostheses using hidden Markov models, IEEE Transactions on Biomedical Engineering 52 (2005) 121–124.
[166] K. Englehart, B. Hudgins, A. D. C. Chan, Continuous multifunction myoelectric control using pattern recognition, Technology and disability 15 (2003) 95–103.
[167] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Q. Wang, J. H. Yang, A framework for hand gesture recognition based on accelerometer and EMG sensors, IEEE Transactions on Systems Man and Cybernetics Part a-Systems and Humans 41 (2011) 1064–1076.
[168] K. R. Wheeler, M. H. Chang, K. H. Knuth, Gesture-based control and EMG decomposition, IEEE Transactions on Systems Man and Cybernetics Part C-Applications and Reviews 36 (2006) 503–514.
[169] J. Monsifrot, E. Le Carpentier, Y. Aoustin, D. Farina, Sequential decoding of intramuscular EMG signals via estimation of a Markov model, IEEE Transactions on Neural Systems and Rehabilitation Engineering 22 (2014) 1030–1040.
[170] K. S. Lee, EMG-based speech recognition using hidden markov models with global control variables, IEEE Transactions on Biomedical Engineering 55 (2008) 930–940.
[171] A. D. Chan, K. Englehart, B. Hudgins, D. F. Lovely, Hidden Markov model classification of myoelectric signals in speech, IEEE Engineering in Medicine and Biology Magazine 21 (2002) 143–146.
[172] A. D. Chan, K. Englehart, B. Hudgins, D. F. Lovely, Myo-electric signals to augment speech recognition, Medical & Biological Engineering & Computing 39 (2001) 500–504.
[173] Z. Li, M. Hayashibe, C. Fattal, D. Guiraud, Muscle fatigue tracking with evoked EMG via recurrent neural network: Toward personalized neuroprosthetics, IEEE Computational Intelligence Magazine 9 (2014) 38–46.
[174] P. Xia, J. Hu, Y. Peng, EMG-based estimation of limb movement using deep learning with recurrent convolutional neural networks, Artificial Organs 42 (2018) E67–E77.
[175] F. Quivira, T. Koike-Akino, Y. Wang, D. Erdogmus, Translating sEMG signals to continuous hand poses using recurrent neural networks, in: Proceedings of the IEEE-EMBS International Conference on Biomedical and Health Informatics, 2018, pp. 166–169.
[176] A. Graves, Generating sequences with recurrent neural networks, CoRR abs/1308.0850 (2013).
[177] Y. Hu, Y. Wong, W. Wei, Y. Du, M. Kankanhalli, W. Geng, A novel attention-based hybrid CNN-RNN architecture for sEMG-based gesture recognition, PloS One 13 (2018) e0206049.
[178] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A. G. Hager, S. Elsig, G. Giatsidis, F. Bassetto, H. Muller, Electromyography data for non-invasive naturally-controlled robotic hand prostheses, Scientific data 1 (2014) 140053.
[179] A. Samadani, Gated Recurrent Neural Networks for EMG-Based Hand Gesture Classification. A Comparative Study, in: Proceedings of the 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2018, pp. 1–4. doi:10.1109/EMBC.2018.8512531.
[180] M. Atzori, A. Gijsberts, S. Heynen, A.-G. M. Hager, O. Deriaz, P. van der Smagt, C. Castellini, B. Caputo, H. Müller, Building the Ninapro database: A resource for the biorobotics community, in: Proceedings of the 2012 4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), 2012, pp. 1258–1265. doi:10.1109/BioRob.2012.6290287.
[181] M. Simão, P. Neto, O. Gibaru, EMG-based online classification of gestures with recurrent neural networks, Pattern Recognition Letters 128 (2019) 45–51.
[182] M. Simão, P. Neto, O. Gibaru, UC2018 DualMyo Hand Gesture Dataset, 2018.
[183] S. Pizzolato, L. Tagliapietra, M. Cognolato, M. Reggiani, H. Müller, M. Atzori, Comparison of six electromyography acquisition setups on hand movement classification tasks, PloS One 12 (2017) e0186132.
[184] S. E. Schmidt, C. Holst-Hansen, C. Graff, E. Toft, J. J. Struijk, Segmentation of heart sound recordings by a duration-dependent hidden Markov model, Physiological Measurement 31 (2010) 513–529.
[185] A. D. Ricke, R. J. Povinelli, M. T. Johnson, Automatic segmentation of heart sound signals using hidden markov models, in: Computers in Cardiology, volume 32, 2005, pp. 953–956.
[186] P. Sedighian, A. W. Subudhi, F. Scalzo, S. Asgari, Pediatric heart sound segmentation using Hidden Markov Model, in: Proceedings of the 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2014, pp. 5490–5493.
[187] C. S. Lima, D. Barbosa, Automatic segmentation of the second cardiac sound by using wavelets and hidden Markov models, in: Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2008, pp. 334–337.
[188] E. Sejdić, G. A. Malandraki, J. L. Coyle, Computational deglutition: Using signal- and image-processing methods to understand swallowing and associated disorders, IEEE Signal Processing Magazine 36 (2019) 138–146.
[189] P. B. Shull, W. Jirattigalachote, M. A. Hunt, M. R. Cutkosky, S. L. Delp, Quantified self and human movement: a review on the clinical impact of wearable sensing and feedback for gait analysis and intervention, Gait and Posture 40 (2014) 11–9.
[190] C. Donohue, Y. Khalifa, S. Perera, E. Sejdić, J. L. Coyle, How Closely do Machine Ratings of Duration of UES Opening During Videofluoroscopy Approximate Clinician Ratings Using Temporal Kinematic Analyses and the MBSImP?, Dysphagia (2020).
[191] S. Mao, A. Sabry, Y. Khalifa, J. L. Coyle, E. Sejdić, Estimation of laryngeal closure duration during swallowing without invasive X-rays, Future Generation Computer Systems (2020).

[192] S. Mao, Z. Zhang, Y. Khalifa, C. Donohue, J. L. Coyle, E. Sejdić, Neck sensor-supported hyoid bone movement tracking during swallowing, Royal Society Open Science 6 (2019) 181982.
[193] C. Nickel, C. Busch, S. Rangarajan, M. Möbius, 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. Panahandeh, N. Mohammadiha, A. Leijon, 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: N. Maglaveras, I. Chouvarda, P. de Carvalho (Eds.), Precision Medicine Powered by pHealth 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. Glasmachers, Limits of End-to-End Learning, arXiv preprint arXiv:1704.08305 (2017).
[202] G. Cheron, J.-P. Draye, M. Bourgeios, 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 electromyographic 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:10.1109/DSAA.2015.7344872.
[205] V. G. Sujadevi, 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:10.1007/978-3-319-68385-0_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-fMRI 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. Sejdić, 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, 2015, 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 Neural Systems and Rehabilitation Engineering 27 (2019) 760–771.