Reservoir Computing Approaches to Microsleep Detection

Stephen J Weddell[1]‡, Sudhanshu Ayyagari[1] and Richard D Jones[1,2]

[1]University of Canterbury, Dept. of Electrical & Computer Engineering, Christchurch 8140, New Zealand
[2]New Zealand Brain Research Institute, Christchurch, New Zealand

E-mail: steve.weddell@canterbury.ac.nz

October 2020

Abstract.
Objective. The detection of microsleeps in a wide range of professionals working in high-risk occupations is very important to workplace safety. A microsleep classifier is presented that employs a reservoir computing (RC) methodology. Specifically, echo state networks (ESN) are used to enhance previous benchmark performances on microsleep detection. *Approach.* A clustered design using a novel ESN-based leaky integrator is presented. The effectiveness of this design lies with the simplicity of using a fine-grained architecture, containing up to 8 neurons per cluster, to capture individualized state dynamics and achieve optimal performance. This is the first study to have implemented and evaluated EEG-based microsleep detection using RC models for the detection of microsleeps from the EEG. *Main results.* Microsleep state detection was achieved using a cascaded ESN classifier with leaky-integrator neurons employing 60 principal components from 544 power spectral features. This resulted in a leave-one-subject-out average detection in performance of $\phi = 0.51 \pm 0.07$ (mean $\pm$ SE), AUC-ROC $= 0.88 \pm 0.03$, and AUC-PR $= 0.44 \pm 0.09$. *Significance.* Although performance of EEG-based microsleep detection systems is still considered modest, this refined method achieved a new benchmark in microsleep detection.

Keywords: EEG, microsleep, reservoir computing, classification

1. Introduction

Microsleeps are brief ($\approx$ 0.5–15 s) involuntary lapses in consciousness in which a person has a complete suspension of performance due to falling asleep, momentarily. The neural dynamics of sleep transition, as seen in both the EEG spectral power and task performance measures in drowsy individuals performing an active task, such as

‡ Present address: Dept. of Electrical & Computer Engineering, University of Canterbury, Christchurch 8140, New Zealand.

driving, usually vary rapidly between periods of wakefulness and sleep [1]. Despite the prevalence and dangers of these microsleeps in everyday life [2–5], a complete system-wide understanding of the brain mechanisms underlying microsleeps remains elusive [6].

Over the past 15 years, our research group has carried out several studies using electrophysiological and behavioural tests to investigate the nature and causes of lapses of responsiveness [2, 6–12]. Using experimental data from these studies, several state-of-the-art techniques have been, and continue to be, developed to automatically identify, characterize, detect, and predict microsleeps.

In a previous study, EEG recordings were used from electrodes at 16 scalp locations, while the subjects performed a 1-D continuous visuomotor tracking task for 1 hour, in two separate sessions [2, 7, 8, 13]. Microsleeps were defined operationally as the presence of either a video-microsleep and/or a tracking flat spot. Video-based microsleeps were identified by prolonged eye-lid closure, head nodding, and/or terminated by waking head jerks. The gold standard for identifying microsleeps was generated by manually inspecting the tracking task and the facial video.

Peiris *et al* [2] reported that during microsleeps, an increase in EEG power in the delta, theta, and alpha spectral bands, and a decrease in the beta and gamma bands was observed. They also reported that 14 of the 15 subjects had at least one microsleep, with an overall mean rate of 39.3 per hour and mean duration of 3.4 s [2]. Linear discriminant analysis (LDA) was used to form detection models based on individual subject data and stacked generalization was utilized to combine the outputs of multiple classifiers to obtain the final prediction [8]. The best performance achieved used an LDA and stacked generalization-based technique on EEG-power spectral features [8]. This major focus of our research has continued to be the development of a state-of-the-art microsleep detection system.

The motivation for this research was to investigate classification performance on this earlier microsleep study using a new recurrent neural network architecture, based on reservoir computing (RC). Our approach employs a novel fine-grained architecture for feature classification. Performance results using our method are compared to four commonly-used classifier configurations. This study presents and evaluates architectural enhancements, based on RC structure, that advances the case for an RC microsleep classifer.

2. Methods

2.1. Data

2.1.1. Subjects The data in the current study were recorded in an earlier study [2], in which there were 15 healthy male participants (mean age 26.5 years, range 18-36). Visual acuities were 6/9 or better in each eye. All subjects had slept normally the previous night (mean = 7.8 h, min = 5.1 h) and were considered non-sleep-deprived. Ethical

approval for the study had been obtained from the Canterbury Ethics Committee.

2.1.2. Procedure Subjects performed a visuomotor tracking task for 1 h while EEG, facial video, and tracking performance were recorded. The 1-D tracking task had a continuous pseudo-random preview target (bandwidth 0.164 Hz, 8-s preview) [14, 15] and a steering wheel, sampled at 64 Hz, to control a cursor near the bottom of the screen. Head and facial features were recorded from a video camera 1 m in front of the subject (frame rate 25 Hz). The time-synchronized video provided an independent measure of the presence of microsleeps.

EEG was recorded from electrodes at 16 scalp locations, bandpass filtered ($0.1 - 100$ Hz), and digitized at 256 Hz. Electrodes were placed according to the international 10-20 system. Bipolar derivations were used in feature calculations: Fp1-F7, F7-T3, T3-T5, T5-O1, Fp2-F8, F8-T4, T4-T6, T6-O2, Fp1-F3, F3-C3, C3-P3, P3-O1, Fp2-F4, F4- C4, C4-P4, and P4-O2.

Each subject attended two sessions, at least one week apart (mean 17 days, range 7-50 days), held following lunch between 12.30 pm and 5.00 pm.

2.1.3. Behavioural Analysis - Microsleeps Occurrences of microsleeps (previously termed 'lapses' [2, 7, 8], plus determination of their start and end points with a time resolution of 1.0 s, were determined off-line from flatspots in tracking response (excluding when target velocity was approximately zero) and/or prolonged eye-closure, rated subjectively from facial video. A subset of behavioural and EEG data from the 8 subjects who had at least one unequivocal microsleep comprising a concomitant flatspot AND prolonged eye-closure over the two sessions was selected for the current study, due to its focus on EEG-based detection, as opposed to behavioural characterization, of microsleeps. The subset of 8 subjects had a total of 917 microsleeps [8], i.e., an average microsleep rate of 57.3 /h.

2.2. System overview

An overview of our microsleep detection system is shown in Figure 1. The microsleep detection system incorporates pre-processing/conditioning, feature extraction, feature selection/reduction, and pattern classification stages. Signals from eye movements, eye blinks, ECG, EMG, and line noise can be orders of magnitude larger than brain-generated EEG and are the main sources of artefacts in EEG data. In order to overcome this problem, a comprehensive set of pre-processing methods were implemented on raw EEG data, and this followed the approach taken by Peiris *et al* [8].

Pre-processing of the dataset comprised stages encompassing EEG data acquisition, artefact removal, mean removal, rescaling, data filtering, and feature matrix generation. For example, independent components analysis (ICA) was used to remove eye blink artefacts [16]. The artefact-free signal was then filtered to remove 50 Hz mains power-line interference using an infinite-impulse response notch filter with a Q-factor of 35.

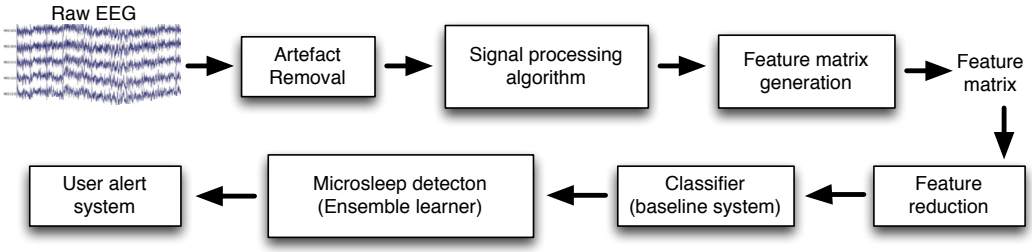

Figure 1. Microsleep detection system.

The mean and standard deviation of the first 2 min of the signal were calculated and z-scores relative to the baseline of the signal were used to allow comparisons to be made between subjects and over multiple sessions. Epochs of 2 s containing samples with an absolute z-score > 3.0 were rejected as artefacts and excluded from analysis [8].

EEG feature reduction algorithms generate meta-features from original features, using, for example, PCA, so as to minimize and optimize the number of features passed to the classifier with minimal loss of significant information from the feature sets. Consequently, the pattern classification stage assigns class labels to given input values based on the training algorithm. In this study all classifiers were trained on concatenated data of seven training subjects and this was tested on data from an eighth (test) subject using the leave-one-subject-out (LOSO) method. Results were taken from each test subject and candidate test subjects were rotated across all 8 subjects, thus forming an average classifier result. The generation of meta-features is discussed in detail by Ayyagari [17].

Validation of training and testing data required properly-labelled states indicating a microsleep. Behavioural metrics of microsleeps were decimated to a resolution of 1 Hz. The presence of a microsleep was treated as a binary state, where "1" indicated the presence of a microsleep at any point in time within the 1 s epoch and "0" indicated the responsive or baseline state. Data rated by human experts served as the gold standard for training and gauging performance of automated classifiers.

2.3. Signal processing

In order to eliminate electrode-pop artefacts, each derivation was normalized into z-scores. The z-scores of each epoch were computed using the mean and standard deviation of the first 2-min of each 1-hour-long record of data of the same session. To exclude unacceptably noisy data, 1-s epochs with absolute z-scores over 3.0 were deleted and removed from further analysis. This process resulted in pruning 580 epochs of the 7200 epochs across both 1-h sessions per subject on average. Corresponding gold standard events were also excluded from analysis at an average of 26 flat-spot epochs and 34 video-lapse epochs per subject in both sessions.

Thirty four spectral features per derivation (13 spectral power + 12 normalized

power + 9 power ratios) were calculated for each of the 16 channels using a 2-s sliding window function, stepping at 1-s intervals, resulting in $34 \times 16 = 544$ spectral features over the 16 channels. Provision was incorporated to ensure that the 2 s window did not overlap the discontinuities.

2.4. Feature reduction

An EEG feature was defined as an arbitrary time series extracted from a single EEG derivation using a given signal processing algorithm [8]. A feature vector was a vector of all feature values for a particular data instance.

Data in each epoch were detrended to remove any DC shifts and the spectrum was estimated using a 40th-order Burg model. Thirty-four spectral features, comprising 13 spectral power (SP), 12 normalized spectral power (NSP), and 9 power ratio (PR) features listed in Table 1 were calculated for each of the 16 derivations, giving a total of 544 spectral features.

Table 1. Spectral features calculated from each EEG derivation.

Feature	**Frequency band (Hz)**
Mean spectral power (normalized values)	
Delta (δ)	1.0 – 4.5
Theta (θ)	4.5 – 8.0
Alpha 1 (α_1)	8.0 – 10.5
Alpha 2 (α_2)	10.5 – 12.5
Alpha (α)	8.0 – 12.5
Beta 1 (β_1)	12.5 – 15.0
Beta 2 (β_2)	15.0 – 25.0
Beta (β)	12.5 – 25.0
Gamma 1 (γ_1)	25.0 – 35.0
Gamma 2 (γ_2)	35.0 – 45.0
Gamma (γ)	25.0 – 45.0
High	> 45.0
Overall	0.1 – 100
Spectral power ratios (absolute values)	
$\theta/\beta, \theta/\alpha, \alpha/\beta, \delta/\theta, \alpha/\delta, \beta/\delta, \beta_2/\alpha, \beta_1/\beta_2$	

Ten feature reduction techniques were evaluated in this study, with principal components analysis (PCA) proving to be the best overall feature reducer [17]. Of the 544 spectral features secured, 60 principal components were used from all of the 8 subjects to train and test a suitable classifier for microsleep detection.

2.5. Classification methods

Several classifiers were considered for comparison with a new artificial neural network from a family known as a reservoir computing, for the detection of microsleeps. A key requirement is to capture the dynamics of input signals. Typically, this requires a complex training process. However, capturing the *fidelity* of such features to ensure accurate classification is a key question that will be addressed in this article. Three, relatively-diverse classifiers were used in this study, one resulting in 5 sub-types.

2.5.1. Linear discriminant analysis Linear discriminant analysis (LDA) was used to determine which continuous variables could discriminate between two or more groups [18]. LDA assumes that the group memberships of the initial cases (training set) are known correctly. This analysis yields information which can then be used to classify a future case with an unknown group membership into a group. LDA maximizes the ratio of between-class variance to the within-class variance in any particular data set, thereby achieving maximal linear separability [18].

LDA was used by Peiris *et al* [8] to form classification models capable of detecting microsleeps and was set as the baseline for the other classifier models.

2.5.2. Support vector machines Support Vector Machines (SVM) are a set of supervised learning techniques used for regression, classification, and outlier detection based on the concept of decision planes, which define decision boundaries [19]. SVM performs classification tasks by constructing hyperplanes in a multidimensional space that separates cases of different class labels [20]. Support vectors turn high dimensionality problems into linear classification problems. SVM algorithms are also well equipped to handle multiple continuous and categorical variables.

SVMs with linear and Gaussian kernel functions have previously been used for microsleep detection [21–23]. More recently, LaRocco [24] applied SVMs for microsleep detection on the same dataset used in this research. Mediocre performance was reported for an SVM with a polynomial kernel, and poor performance was reported with a Gaussian kernel.

SVMs, however, provide a unique solution, since their optimality problem is convex [25]. This is an advantage compared to neural network schemes, which have multiple solutions associated with local minima, and for this reason may not offer a robust solution over different samples.

2.5.3. Reservoir computing Reservoir computing (RC) denotes a specialised family of recurrent neural networks that support a sparse, randomly-connected 'reservoir' that can realise a dynamical system. Two RC structures are the liquid state machine (LSM) and echo state network (ESN). LSMs are biologically-inspired recurrent networks, whereas ESNs are fashioned as a general engineering tool [26]. Both RC members support a simplified training structure, where recurrent nodal pathways can be used to enhance

system dynamics that can be captured to a memory structure for separation using a supervised training set [27].

2.6. Reservoir computing approaches

ESNs have been used in neuroengineering for detection of epileptic seizures [28]. Furthermore, advanced variations of ESN architectures have been shown to predict cues within preprocessed EEG, such as microsleeps [29, 30]. In this subsection, the basic structure of two RC architectures is outlined, whereas a leaky approach to prolong the 'echo' quality of ESNs for application to relatively low-frequency EEG signals, is described in Subsection 2.6.3.

2.6.1. Liquid state machines The basic principle behind liquid state machines (LSMs) is that the period between neural firing is not constant, as they decode the time between spikes, which is thought to be the main source of information transfer between neurons. LSMs are based on a spiking neural network model [31]. Their main properties are recurrency and that neurons have spiking activity, usually based on a complex synaptic model. The recurrent weights in an LSM are not trained in a supervised manner, unlike recurrent neural networks (RNNs). LSMs are also used in computational neuroscience to study functional properties of neural circuits by abstraction [32].

2.6.2. Echo state networks Characteristically, echo state networks have a simplified training structure and provide a "reservoir of rich dynamics" [33]. Based on a set of discretized, time-varying inputs, a fixed sparse matrix $\mathbf{W}_{\mathrm{DR}}$ is used to implement a recurrent network with enhanced spectral diversity, where a linear readout $\mathbf{W}_{\mathrm{out}}$ is trained to produce an output. The state vector $\mathbf{x}(n)$ maintains the relationship between the input vector $\mathbf{u}(n)$ and output vector $\mathbf{y}(n)$, which can be expressed as

$$\mathbf{x}(n) = \varphi\Big(\mathbf{W}_{\mathrm{in}}\,\mathbf{u}(n)^{\mathrm{T}} + \mathbf{W}_{\mathrm{DR}}\,\mathbf{x}(n\text{-}1)^{\mathrm{T}} + \mathbf{W}_{\mathrm{back}}\,\mathbf{y}(n\text{-}1)^{\mathrm{T}}\Big), \tag{1a}$$

$$\mathbf{y}(n) = \Psi\Big(\mathbf{W}_{\mathrm{out}}\,\mathbf{x}(n)^{\mathrm{T}}\Big), \tag{1b}$$

where $\mathbf{W}_{\mathrm{DR}}$ is the dynamic reservoir matrix, $\mathbf{W}_{\mathrm{back}}$ is the feedback matrix, $\varphi(\cdot)$ and $\Psi(\cdot)$ are input and output activation functions, respectively, $\mathbf{W}_{\mathrm{out}}$ is the output weight matrix, and $n \in \{1, 2, \cdots, N\}$.

With reference to the architecture shown in Figure 2, the operation of an ESN can be described as follows. The input vector $\mathbf{u}(n)$ is mapped into state space $\mathbf{x}(n)$ through the *echo state property* supported by a sparsely connected recurrent matrix, $\mathbf{W}_{\mathrm{DR}}$. Linear regression is used to train the ESN output matrix $\mathbf{W}_{\mathrm{out}}$ to facilitate recombination of output data. Input and optional feedback matrices $\mathbf{W}_{\mathrm{in}}$ and $\mathbf{W}_{\mathrm{back}}$, respectively, are dense randomly-connected matrices that facilitate the distribution of inputs and output data to the dynamic reservoir. Lastly, as with most ANNs, an activation function $\varphi(\cdot)$ is supported to ensure data remains bound and provides non-linear output capability. In essence, the network is acting as a set of finely-tuned, matched filters.

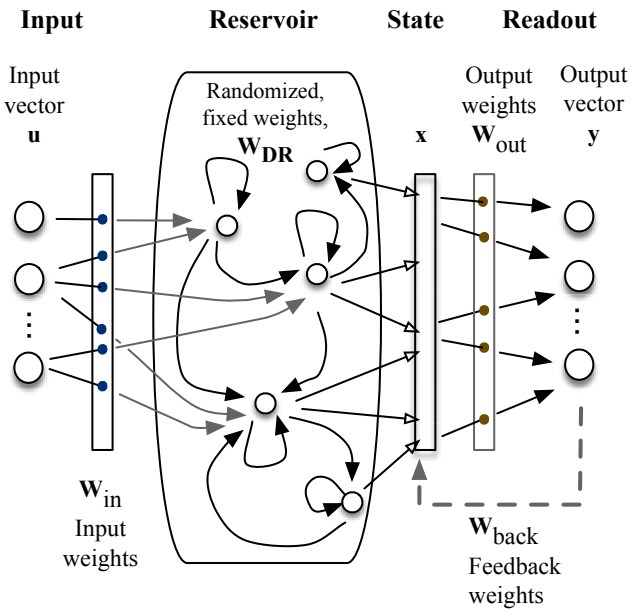

Figure 2. A basic form of the echo state network (ESN) architecture.

2.6.3. ESNs with leaky neurons A disadvantage of echo state networks is that they do not have a time constant which means that their dynamics cannot be 'slowed down', as can be performed, for example, in the dynamics of a differential equation. Therefore, given the relatively slow dynamic of EEG signals, such as low frequency sinusoidal waveforms, the leaky-integrator ESN model [33] was used. By employing leaky-integrator ESNs, the temporal characteristics of a learning task can be exploited by using the individual state dynamics of the system [33].

A leaky integrator neuron network can be interpreted from Jaeger *et al* [34] as,

$$\dot{\mathbf{x}} = \frac{1}{\tau}\Big(-\alpha\mathbf{x} + \varphi\big(\mathbf{W}_{\text{in}}\mathbf{u} + \mathbf{W}_{\text{DR}}\mathbf{x} + \mathbf{W}_{\text{back}}\mathbf{y}\big)\Big), \tag{2}$$

$$\mathbf{y} = \Psi\Big(\mathbf{W}_{\text{out}}[\mathbf{x};\,\mathbf{u}]\Big), \tag{3}$$

where τ is the time constant, α is the leak decay rate, functions $\varphi(\cdot)$ and $\Psi(\cdot)$ use non-linear and linear activations, respectively, and where the semicolon in $[\mathbf{x};\,\mathbf{u}]$ is used to represent the conjoining of input vector $\mathbf{u}$ with state vector $\mathbf{x}$. Thus, reservoir states are extended with input vector $\mathbf{u}$ to essentially 'feeding' output units $\mathbf{y}$ [35].

For a discrete time-sampled system, Equations 2 and 3 can be represented as difference equations with step-size δ as

$$\begin{aligned}\mathbf{x}(n+1) =& \Big(1 - \frac{\delta\alpha}{\tau}\Big)\mathbf{x}(n) + \\ & \frac{\delta}{\tau}\Big(\varphi\big(\mathbf{W}_{\text{in}}\mathbf{u}\big((n+1)\delta\big) + \mathbf{W}_{\text{DR}}\mathbf{x}(n) + \mathbf{W}_{\text{back}}\mathbf{y}(n)\big)\Big),\end{aligned} \tag{4}$$

$$\mathbf{y}(n) = \Psi\big(\mathbf{W}_{\text{out}}[\mathbf{x}(n);\,\mathbf{u}(n\delta)]\big). \tag{5}$$

To achieve optimal classification performance, consolidation, in terms of network state dynamics was an important requirement. To achieve, this we introduced a *fine-grained* ESN architecture.

2.7. Fine-grained echo state networks

Echo state networks transform the information at the input to an excited state, where the outputs represent various class hypotheses formulated from the information in state vectors [33, 35]. This is performed through operations and interconnections of the output matrix to inputs, and subsequently, reservoir units to an output weight vector. Following this, linear regression of targets on the input matrix is computed. Using this method as a basis, an enhanced ESN architecture was developed, which achieved a high-level of classification performance.

2.7.1. Fine-grained re-sequenced states A set of small leaky-integrator ESN modules are generated, comprising p neurons, where p is the number of neurons (5 in this case) and K sequences (K = 3), is considered an individual classifier. Initial observations showed that 8 or less neurons of a modified cascaded-leaky-integrator structure presented markedly different dynamic properties across their random instantiations. This is not the case for larger ESNs, where inter-network differences became insignificant with growing network size [36].

To consolidate the results from collections of small fine-grained leaky-integrator modules, a cascaded leaky-integrator approach was employed. This required segmentation of a single, large state matrix into a set of segregated K state sequences of equal size, where K is a small integer. Each sequence reflects a few, equally-spaced 'snapshot's of state vector development when the ESN reads a sample. The regression weights in output matrix $\mathbf{W}_{\text{out}}$ (Eq. 5) are first computed for all segregated states and form class hypotheses.

Class hypotheses H_m^i are generated from 60 power spectral features, where $m \in \{1, \cdots, 7\}$ employed from training ensembles comprising 7 of 8 subjects used in our study and over $i \in \{1, \cdots, L\}$ samples. This training structure was required for the leave-one-subject-out (LOSO) cross validation training method, which was applied to all classifiers used in this study.

Each K segregated training sample represented a sequence vector that was concatenated to form a single state vector, where the number of mappings (connections) from each K sequence to the final state vector was determined by m, where m is the replication factor. Replication of state vector sequences provide a means whereby key dynamic sequences can be more adequately accessed throughout the network, as compared to basic leaky ESN structures. Given the coarse granularity of the sampled input, which may be reflected in terms of phase errors resulting in misclassification of feature vectors, we propose that small replicated (and distributed) state vector sequences can reduce the incidence of misclassifications by introducing small temporal offsets to

capture complete cues, resulting in improved performance.

Rather than use the entire state vector, dimensionality was further reduced through the use of segmentation and re-sequencing of the state vector and input sequences. This allowed smaller, fine-grain networks to compensate for ESNs short-term memory capacity. A schematic of this segmentation and re-sequencing is depicted in Figure 3.

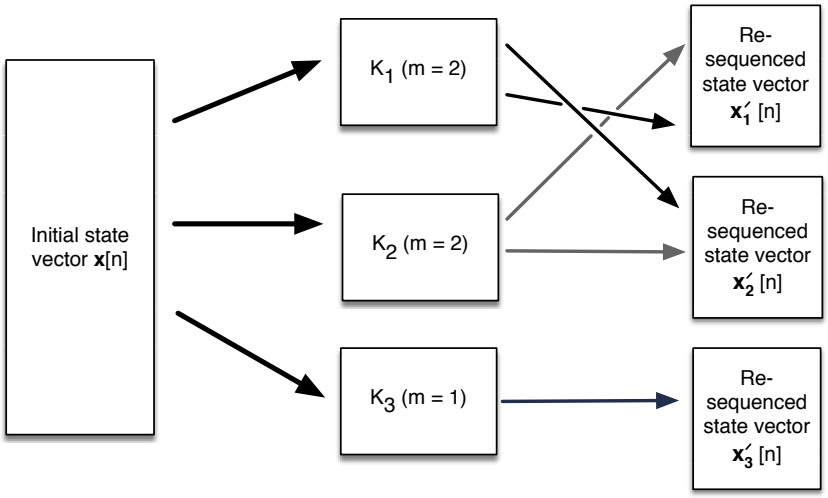

Figure 3. Schematic representation of the state vector $\mathbf{x}[n]$ and its reconstruction, $\mathbf{x}'[n]$, using segmented sequences, K_1, K_2, and K_3.

Therefore, each reservoir in the cascaded-leaky-integrator ESN structure can be considered an individual detector for random, p-dimensional, non-linear, dynamic features of the input signals. Consequently, joining all K extended state vectors creates a series of p-neuron ESNs which can transform an input sequence into a static $(p + \mathbf{m}) \times \mathbf{K}$-dimensional feature. Here, $\mathbf{m}$ is the output vector of dimension M, and is the contribution of all reservoir state components that vary across each of these network instantiations.

The major advantage of this innovative architecture is that small networks, i.e., those of order 5–8 cascaded-leaky-integrator-neuron structures, are capable of outperforming nodal structures that are considerably larger in size. However, a disadvantage of this approach is that, by increasing architectural complexity, the possibility exists of over-fitting the network. This is particularly the case if global network parameters and reservoir size are not chosen carefully.

2.7.2. Optimization and global network parameters The cascaded-leaky-integrator-ESN formulation uses 5 parameters, 3 of which include scaling parameters to the randomly-generated connection weights in the reservoir. The leakage factor or leaking rate is the fourth and the most crucial optimization parameter for this model. Apart from the scaling parameters and the leakage rate, the reservoir size can also be considered as a fifth parameter that must be determined [37].

Three scaling parameters are required for optimization: *bias scaling*, *input scaling*, and *spectral radius*. Bias scaling is used to push the reservoir states closer to -1 or +1, which corresponds to the non-linear region of the hyperbolic tangent function. Therefore, for the higher values of the bias scaling, the reservoir tends to move towards non-linearity [38]. Usually, scaling values set for the bias are either 0, 0.1, and 1.

Similar to the bias parameters, the leakage rate can have a substantial effect on the fading of the reservoir dynamics and can push reservoirs with a spectral radius greater than 1 into more stable conditions. Leakage rate is highly dependent on the input frequency. For example, an input signal with a frequency component substantially below the system cut-off frequency will not be dampened in a linear reservoir [39].

Lastly, reservoir size is related to the network memory and network model complexity. It is identified as one of the most important factors which can influence the overall performance of the system. These disadvantages can be overcome with the use of the leaky-integrator neuron ESN structures as they can scale the reservoir size quadratically. Therefore, a reservoir size is usually selected at a point where increasing the size has little or no effect on the overall performance of the network.

The optimal global parameters used for the cascaded leaky-integrator ESN model used for microsleep detection are given in Table 2.

Table 2. Cascaded leaky-integrator ESN global parameters for optimal microsleep detection.

Parameter	**Value**	**Parameter**	**Value**
Spectral radius	0.88	Leakage rate	0.05
Input scaling	0.2	Neurons/cluster	5–10
Bias scaling	0.5		

The memory capacity of a network is strongly associated with the non-linearity of the system [39]. The more non-linear the reservoir, the shorter the memory. Hence, classification tasks that require a long memory usually require a very large reservoir or a linear reservoir.

2.8. An ESN classifier for microsleep detection

To overcome the disadvantages associated with large, single-structure ESNs for complex classification problems, we proposed an alternative solution for microsleep detection. This includes the fusion of multiple, fine-grained individual networks into a combined classifier model. Accordingly, class hypotheses from each classifier are combined and the mean of the individual votes calculated for each classifier. Calculating the mean of the vote combination is performed, as it averages vote fluctuations due to single classifier biases [40].

Extending the fine-grained ESN methodology outlined in Subsection 2.7.1, a combined (i.e., multiple) classifier approach is taken, with the aim of enhancing

classification performance. The architecture shown in Figure 4 can be described as follows.

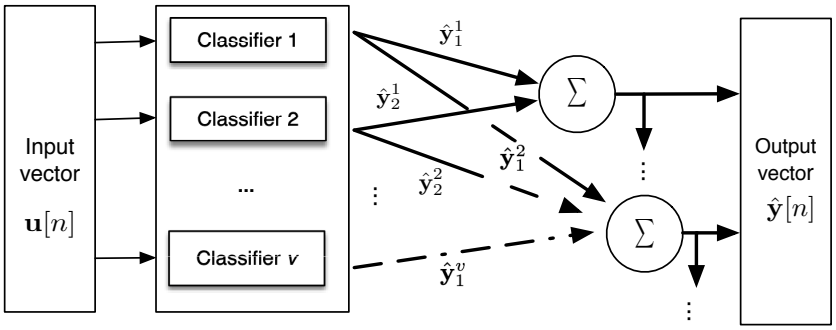

Figure 4. Schematic of a combined classifier model of the cascaded leaky-integrator ESN.

Artefact-removed EEG features discussed in Section 2.2 form the basis of input vector $\mathbf{u}[n]$. A series of baseline classifiers are formed from fine-grained ESN structures during training. Each classifier used a discrete leaky ESN, as defined by Equations 4 and 5, and where the sigmoidal and linear activation functions were used for recurrent nodes and readout layers, respectively. Integration and re-ordering of state vectors described in Section 2.7 provides the basis for this refinement. For each classifier output m, vector $\hat{y}_{\mathbf{C}}^{\mathbf{m}}$ is formed, where $\mathbf{m} \in \{1 \dots M\}$ and $\mathbf{C} \in \{1 \cdots K\}$. This level-0 process is refined using a stacking model, where combined outputs are passed to a meta-learner before being combined to form output, $\hat{y}[n]$, as shown in Figure 4.

The stacking framework depicted in Figure 5 consists of level-0 and level-1 generalizers. The level-0 models are formed by base classifiers which are trained using the input data and the target output. The level-0 outputs are then presented as an input to the level-1 generalizer (meta-learner) which is also trainable.

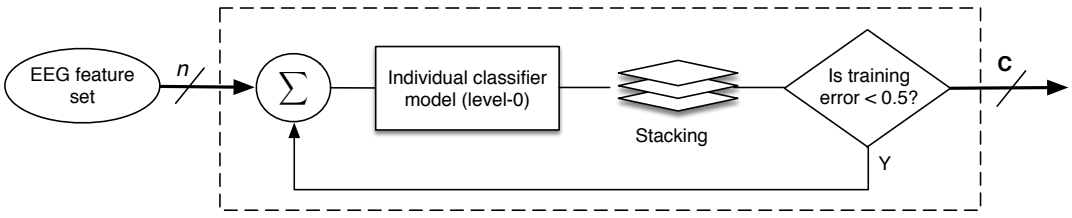

Figure 5. Schematic of a level-0 classifier model of the cascaded leaky-integrator ESN.

The classification phase of the stacking system is shown in Figure 6. New classification cases are generated for level-0 models, each producing a classification value at their output. Subsequently, the resulting base model predictions are passed to the level-1 model and combined linearly. The linear combination scales the output of each model according to its weight, adds the new scaled model outputs, and applies a threshold to the added model output to maximize classification performance.

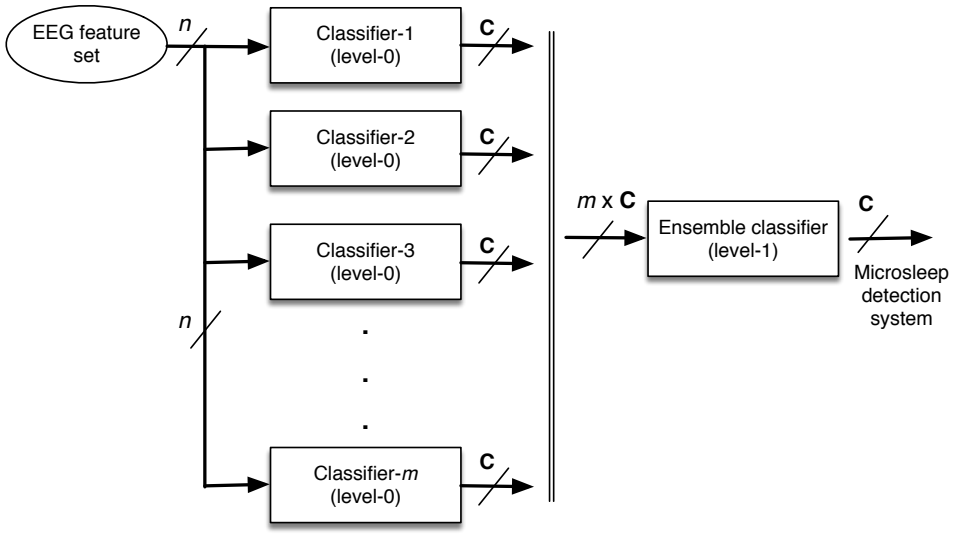

Figure 6. The internal structure of the meta-learner used in the stacking framework. The outputs of the classifiers are denoted by **C** and m represent the number of level-0 models passed to the meta-learner before combining them to form an overall prediction.

In this stacking framework, some of the test data are held back and used to train the level-1 model, while the level-0 models are trained on the rest of the data. Only after all of the level-0 models are trained, the data that was held out is classified using the level-0 models, which then form the training data for the subsequent level-1 model. As the held-out data is not used to train the level-0 models, their predictions are unbiased and, therefore, the level-1 training data accurately reflects the true performance of the level-0 models.

For the classification phase of the stacking system, new cases are generated for level-0 models, each producing a classification value at their output. Subsequently, resulting base model predictions are passed to the level-1 model and are combined linearly. The linear combination scales the output of each model according to its weight, sums outputs of the new scaled model, and applies a threshold to the summed model output to obtain an overall prediction.

Lastly, an 8-fold cross-validation approach has been adopted which ensures that all of the training data are used to train the level-1 model. Therefore, each instance of the training data is used in one test-fold of the cross-validation and predictions from the models built from the corresponding training fold are used to build the level-1 training set, thus generating a level-1 training set for each level-0 training set.

3. Results

Table 3 summarizes microsleep state detection performances of standard ESN, LSM, stacked LDA, SVMP, and cascaded-leaky-integrator ESN classifiers, trained using PCA meta-features on both the pruned and unpruned Study-A EEG datasets. These results are averages of individual detection performances (phi correlation coefficient, ϕ) for each

of the 8 subjects. The corresponding ROC and precision-recall graphs for the cascaded-leaky ESN are shown in Figures 7 and 8, respectively.

Table 3. Detector performances of microsleep states on pruned and unpruned datasets for standard ESN, stacked LDA, LSM, SVMP, and cascaded-leaky integrator ESN-based classifiers.

Classifier	**Microsleeps**
Unpruned EEG Study-A dataset, ϕ(Mean $\pm$ SE)	
Standard ESN	0.30 $\pm$ 0.09
Stacked LDA	0.35 $\pm$ 0.05
LSM	0.38 $\pm$ 0.04
SVMP	0.42 $\pm$ 0.03
Cascaded-leaky ESN	0.44 $\pm$ 0.06
Pruned EEG Study-A dataset, ϕ(Mean $\pm$ SE)	
Standard ESN	0.31 $\pm$ 0.09
LSM	0.40 $\pm$ 0.03
Stacked LDA	0.41 $\pm$ 0.06
SVMP	0.44 $\pm$ 0.06
Cascaded-leaky ESN	0.51 $\pm$ 0.07

The nonparametric Wilcoxon signed-rank test also confirmed that microsleep detection by the cascaded-leaky-integrator ESN was statistically superior to that of the previous LDA-based approach reported by Peiris [8] for the pruned Study A dataset (Phi = 0.51 vs. 0.39, z = 2.54, p = 0.012 (2-tail)).

In summary, the model showed a mean correlation of $\phi = 0.51 \pm 0.07$ (AUC-ROC = 0.91 $\pm$0.04; AUC-PR = 0.47 $\pm$0.09) and this result is the best achieved in this study.

4. Discussion

This is the first study to use an ESN architecture for detection of microsleep states. Furthermore, using a novel modified ESN architecture we achieved an unparalleled detection performance for microsleep detection on pruned data.

The results in Table 3 indicate that the detection performance of the cascaded-leaky-integrator ESN classifier is higher than any other classifier scheme evaluated for microsleep state detection ($\phi = 0.51$). Standard ESN classifiers recorded the lowest performance, which was somewhat expected due to the relatively low frequencies characteristic of EEG signals. Furthermore, we found that the cascaded-leaky-integrator ESNs had a lower value of ϕ on the unpruned datasets ($\phi = 0.44$) than has been achieved using joint entropy ($\phi = 0.50$) [41]. Joint entropy was also used to achieve a 0.25-s prediction of microsleep states at a mean of $\phi = 0.47$ [12].

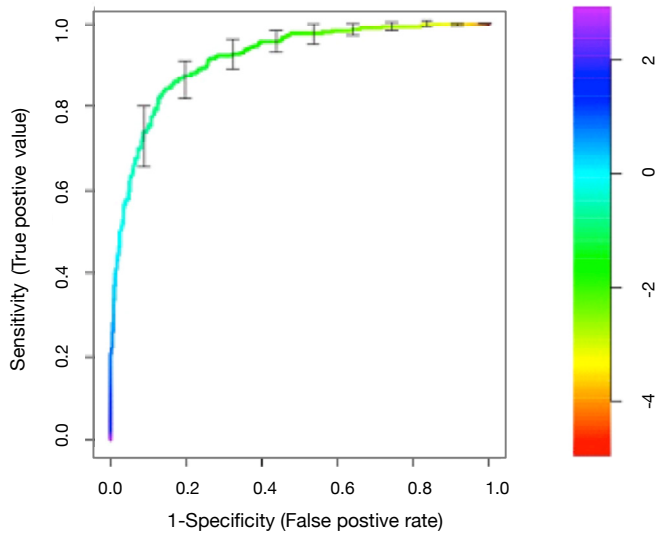

Figure 7. This graph illustrates the ROC analysis, in terms of sensitivity and false positive rate, plotted on the Y- and X-axis, respectively, and is based on the PCA feature selection for a cascaded-leaky-integrator ESN classifier. The vertical bars indicate standard error, and the colour bar on the right indicates which classification threshold results at a certain point on the curve, i.e., from a pair of sensitivity and false positive rate values.

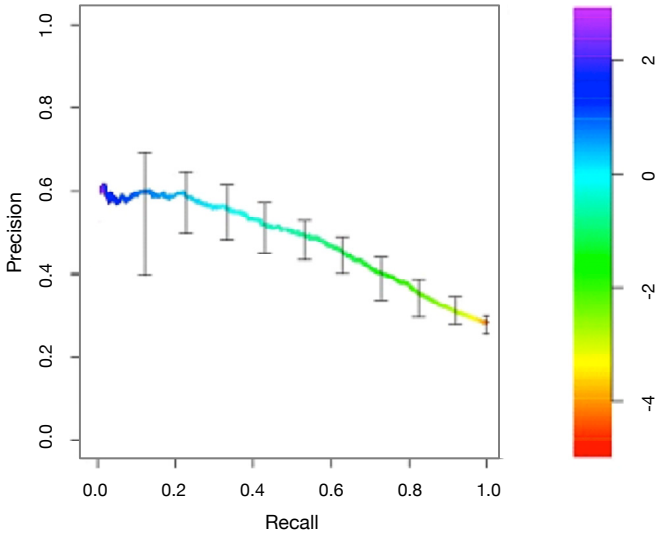

Figure 8. This graph indicates the AUC-PR analysis precision and recall, plotted on the Y- and X-axis respectively, for the PPCA-based leaky integrator ESNs. The vertical bars indicate standard error on both graphs. The colour scale on the right hand side of the graph indicates which classification threshold results in a certain point on the curve, i.e., from a pair of precision or recall values.

From the initial observations, it appeared that certain subjects had consistently low detection values across multiple system configurations (3 classifiers and 6 feature

reduction schemes). Even across the Study-A feature sets, certain subjects consistently scored higher or lower than the average Phi values. Understanding the reasons for successful individual subject classification was thought to provide insights into improving overall detection performance. Consequently, 8-fold cross-validation was used to validate within-subject classifier models on individual subjects. In this type of cross-validation scheme, the data set is divided into 8 subsets and the holdout method was repeated 8 times. Each time, one of the 8 subsets was used as the test set and the other 7 subsets were put aside to form a training set. Finally, the average error across all 8 trials is computed.

Figure 9 provides a within-subject summary of system performance of a stacked-LDA classifier using PCA for microsleep detection on the pruned Study-A EEG dataset. This shows the effect of within-subject classifier variances on the overall lapse detection system. The correlation values for each of the 8 subjects are depicted in terms of phi. The highest phi values across all of the subjects on the pruned Study-A dataset were seen in Subject 804 ($\phi = 0.78$), while the lowest phi values were seen on Subject 810 ($\phi = 0.09$). Overall, there were 5 subjects (804, 809, 814, 817, and 820) who consistently scored higher phi values than the mean phi of the overall lapse detector. The detection performances on the remaining 3 subjects (810, 811, and, 819) was found to be sub-optimal across all of the classifier modules evaluated.

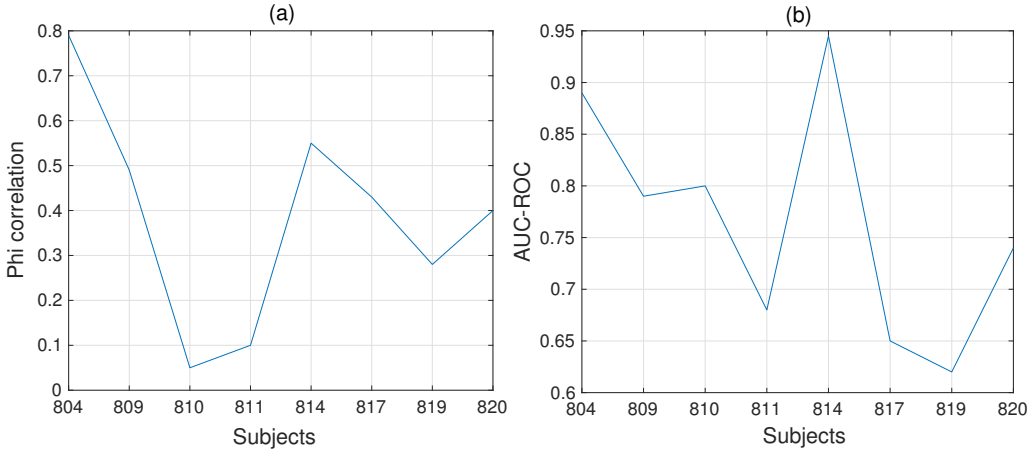

Figure 9. Comparison of within-subject mean detection performances for (a) Phi, and (b) AUC-ROC, on the pruned dataset for the LDA classifier used in this study. The results for each of the remaining four classifiers used in this comparison, as shown in Table 3, showed consistent variances between subjects.

The ROC analysis presented interesting results as several of the classifier modules with low phi values had reasonably high AUC-ROC values. This can be mostly attributed to the large microsleep-responsive imbalance in the data, resulting in reasonably high sensitivities and specificities (hence high AUC-ROC values) but lower precisions due to false detections (hence lower AUC-PR values). For instance, although demonstrating the lowest phi values across all the subjects, Subject 810 reported higher true positive rates (TPR = 0.85) than subjects such as 817, 819 and 820 (TPR = 0.30,

0.47 and, 0.71, respectively). Overall, the highest ROC values were reported in Subject 814 (AUC-ROC = 0.92) and the lowest in Subject 819 (AUC-ROC = 0.63).

All five of the classifiers evaluated used a stacking framework and showed a substantial increase in phi values compared to their base (level-0) models. Stacked LDA modules reported the second greatest increase in detector performance after cascaded-leaky ESNs, with mean phi correlation of 0.41 compared to mean phi of 0.31 on the single classifier modules.

Using eight level-0 models and a meta-learner, the cascaded-leaky-integrator ESN model resulted in a mean phi correlation of 0.51, whereas lumping all the features from seven subjects to create a single model and validating the lumped model on the eighth subject resulted in a mean phi of 0.38. A reason for the performance of the cascaded-leaky-integrator ESNs to be higher than any other model is considered to be due to the leaky ESN approach was built into a multiple classifier scheme (similar to mini-bagging) depicted in Figure 6. The individual classifier output from the cascaded-leaky-integrator structures Figure 4 was later applied to the stacking framework shown in Figure 5, making the overall classifier model robust. Therefore, the predictive outputs of the cascaded-leaky ESN models are much stronger due to rigorous learning that occurs due to these meta-learner models.

Generally, lumping of data prior to model formation resulted in the model being biased towards certain subjects, particularly the ones with the most lapses, resulting in a loss of generalisation ability. Therefore, to prevent such bias, a stacking approach was applied. The stacking approach resulted in improved performance, as the level-0 models were adjusted by the meta-learner according to how well they generalized over the training set. It was also expected that the mean detector performance would increase proportionally with the size of the training set.

In summary, we believe the effectiveness of our classification approach is due to: i) the type of neuron used, where principal features have relatively slow timescales that are more compatible ("tuneable") using leaky-integrator ESNs, ii) the remapping of the state vector to provide each fine-grained network access to specific microsleep features as a function of input vectors, i.e. "snapshots", based on a set of hypotheses, iii) the use of multiple, feature-based, fine-scaled networks which helps support sharp adaptation to a training set, and iv) by effectively combining small (fine-gain) networks from various layers within the stacking framework, such that a "cloud" vote will more likely be directed towards the correct hypothesis.

5. Conclusion and Future Work

Results have been presented for microsleep detectors which were trained using the meta-features from both linear techniques, non-linear techniques, and combinations thereof for our pruned and unpruned Study A EEG dataset. The best detector performance (in terms of the highest mean ϕ) was achieved using the detector model created using fine-grained cascaded-leaky-integrator ESN models with 60 principal spectral components

from the PCA for pruned data.

Ongoing and future work includes the development of a real-time microsleep detection system and predicting the onset of microsleeps, as distinct to state detection.

References

[1] S. Makeig, T. P. Jung, and T. J. Sejnowski. Awareness during drowsiness: Dynamics and electrophysiological correlates. *Can. J. Exp. Psychol.*, 54:266–273, 2000.

[2] M.T.R. Peiris, R.D. Jones, P.R. Davidson, G.J. Carroll, and P.J. Bones. Frequent lapses of responsiveness during an extended visuomotor tracking task in non-sleep-deprived subjects. *J. Sleep Res.*, 15:291–300, 2006.

[3] J. Horne and L. Reyner. Vehicle accidents related to sleep: a review. *Occup. Environ. Med.*, 56:289–94, 1999.

[4] T.A. Dingus, S.G. Klauer, V.L. Neale, A. Petersen, S.E. Lee, J. Sudweeks, and R.R. Knipling. The 100-carnaturalistic driving study. Phase II - results of the 100-car field experiment. Technical Report DOT HS 810593, US Department of Transportation, 2006.

[5] T. Akerstedt. Consensus statement: Fatigue and accidents in transport operations. *J. Sleep Res.*, 9:395, 2000.

[6] G.R. Poudel, C.H.R. Innes, P.J. Bones, R. Watts, and R.D. Jones. Losing the struggle to stay awake: Divergent thalamic and cortical activity during microsleeps. *Hum. Brain Mapp.*, 35(1):257–269, 2014.

[7] P.R. Davidson, R.D. Jones, and M.T.R. Peiris. EEG-based lapse detection with high temporal resolution. *IEEE Trans. Biomed. Eng.*, 54:832–839, 2007.

[8] M.T.R. Peiris, P.R. Davidson, P.J. Bones, and R.D. Jones. Detection of lapses in responsiveness from the EEG. *J. Neural Eng.*, 8:1–15, 2011.

[9] Y. Jonmohamadi, G.R. Poudel, C.R.H. Innes, and R.D. Jones RD. Microsleeps are associated with stage-2 sleep spindles from hippocampal-temporal network. *Int. J. Neural Syst.*, 26:12, 2016.

[10] G.R. Poudel, C.R.H. Innes, and R.D. Jones. Temporal evolution of neural activity and connectivity during microsleeps when rested and following sleep restriction. *NeuroImage*, 174:263 – 273, 2018.

[11] R. J. Buckley, W. S. Helton, C. R.H. Innes, J. C. Dalrymple-Alford, and R. D. Jones. Attention lapses and behavioural microsleeps during tracking, psychomotor vigilance, and dual tasks. *Conscious. Cogn.*, 45:174 – 183, 2016.

[12] A. B. Buriro, R. Shoorangiz, S.J. Weddell, and R.D. Jones. Predicting microsleep states using EEG inter-channel relationships. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 26(12):2260–2269, 10 2018.

[13] M.T.R. Peiris, R.D. Jones, P.R. Davidson, and P.J. Bones. Event-based detection of lapses of responsiveness. In *Proc. Ann. Int. Conf. IEEE Eng. in Med. Biol. Soc.*, pages 4960–4963, 2008.

[14] R. D. Jones and I. M. Donaldson. Measurement of sensory-motor integrated function in neurological disorders: Three computerised tracking tasks. *Med. Biol. Eng. Comput.*, 24(5):536–540, 1986.

[15] R. D. Jones. Measurement and analysis of sensory-motor performance: Tracking tasks. In J. D. Bronzino and D. R. Peterson, editors, *Biomedical Engineering Handbook*, chapter Medical Devices and Systems, pages 31–37. CRC Press, 4 edition, 01 2015.

[16] A. Delorme and S. Makeig. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *J. Neurosci. Meth.*, 134(1):9–21, 2004.

[17] S.S.D.P. Ayyagari. *Reservoir computing approaches to EEG-based detection of microsleeps.* PhD thesis, Electrical & Computer Engineering, University of Canterbury, 2017.

[18] B.D. Ripley. *Pattern recognition and neural networks.* Cambridge University Press, 1996.

[19] S. Ruping. Incremental learning with support vector machines. In *Proc. IEEE Int. Conf. on Data Mining*, pages 641–642, 2001.

[20] T. Hill and P. Lewicki. *Statistics: methods and applications: a comprehensive reference for science, industry, and data mining*. StatSoft, 2006.

[21] M. Golz, D. Sommer, M. Chen, U. Trutschel, and D. Mandic. Feature fusion for the detection of microsleep events. *J.VLSI Signal Process.*, 49:329–342, 2007.

[22] J. Krajewski, A. Batliner, and R. Wiel. Multiple classifier applied on predicting microsleep from speech. In *Proc. 19th Int. Conf. Pattern Recogn.*, pages 1–4, 2008.

[23] M. Golz and D. Sommer. Monitoring of drowsiness & microsleep. In *Proc. Ann. Int. Conf. IEEE Eng. in Med. Biol. Soc.*, pages 1787–1791, 2010.

[24] J. LaRocco. *Detection of microsleeps from the EEG via optimized classification techniques*. PhD thesis, Electrical & Computer Engineering, University of Canterbury, 2015.

[25] N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[26] H. Jaeger. Adaptive nonlinear system identification with echo state networks, 2003.

[27] M. Lukoševičius and H. Jaeger. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.*, 3(3):127–149, August 2009.

[28] P. Buteneers, D. Verstraeten, B.V Nieuwenhuyse, D. Stroob&t, R. Raedt, K. Vonck, P. Boon, and B. Schrauwen. Real-time detection of epileptic seizures in animal models using reservoir computing. *Epilepsy Res*, 103(2-3):124–34, 2012.

[29] S.S.D.P. Ayyagari, R.D. Jones, and S.J. Weddell. EEG-based microsleep detection using supervised learning. *Australas. Phys. Eng. Sci. Med.*, 38:194, 2015.

[30] S.S.D.P. Ayyagari, R.D. Jones, and S.J. Weddell. Optimized echo state networks with leaky integrator neurons for EEG-based microsleep detection. In *Proc. Ann. Int. Conf. IEEE Eng. in Med. Biol. Soc.*, pages 3775–3778, 2015.

[31] W. Maass, T. Natschager, and H. Markram. Real time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.*, 14:2531–2560, 2002.

[32] H. Ju, J. Xu, E. Chong, and A. VanDongen. Effects of synaptic connectivity on liquid state machine performance. *Neural Netw.*, 38:39–51, 2013.

[33] H. Jaeger. The echo state approach to analysing and training recurrent neural networks. *GMD Report*, 148, 2001.

[34] H. Jaeger. The echo state approach to analysing and training recurrent neural networks with an erratum note. Technical Report GMD Report 148, German National Research Center for Information Technology, 2001.

[35] H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert. Optimization and applications of echo state networks with leaky- integrator neurons. *Neural Netw.*, 20(3):335 – 352, 2007.

[36] H. Jaeger. Echo state network. *Scholarpedia*, 2(9):2330, 2007. revision #183563.

[37] M. Lukoševičius and H. Jaeger. On self-organizing reservoirs and their hierarchies. Technical Report, Jacobs University, 2010.

[38] H. Jaeger. Tutorial on training recurrent neural networks, covering BPTT, RTRL, EKF and the "echo state network" approach. *GMD Report*, 159, 2002.

[39] D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt. An experimental unification of reservoir computing methods. *Neural Netw.*, 20(3):391 – 403, 2007.

[40] R. P. W. Duin. The Combining Classifier: To Train or Not to Train? In *Proc. 16th Int. Conf. Pattern Recogn.*, volume 2, pages 765–770, 2002.

[41] A. B. Buriro. *Prediction of microsleeps using EEG inter-channel relationships*. PhD thesis, Electrical & Computer Engineering, University of Canterbury, 2018.