DEVELOPING NEW TECHNIQUES TO ANALYSE AND CLASSIFY EEG SIGNALS

A thesis Submitted by

Mohammed Abdalhadi Diykh

For the award of

Doctor of Philosophy

2018

Abstract

A massive amount of biomedical time series data such as Electroencephalograph (EEG), electrocardiography (ECG), Electromyography (EMG) signals are recorded daily to monitor human performance and diagnose different brain diseases. Effectively and accurately analysing these biomedical records is considered a challenge for researchers. Developing new techniques to analyse and classify these signals can help manage, inspect and diagnose these signals.

In this thesis novel methods are proposed for EEG signals classification and analysis based on complex networks, a statistical model and spectral graph wavelet transform. Different complex networks attributes were employed and studied in this thesis to investigate the main relationship between behaviours of EEG signals and changes in networks attributes. Three types of EEG signals were investigated and analysed; sleep stages, epileptic and anaesthesia.

The obtained results demonstrated the effectiveness of the proposed methods for analysing these three EEG signals types. The methods developed were applied to score sleep stages EEG signals, and to analyse epileptic, as well as anaesthesia EEG signals. The outcomes of the project will help support experts in the relevant medical fields and decrease the cost of diagnosing brain diseases.

Certification of Thesis

This Thesis is the work of Mohammed Diykh except where otherwise acknowledged, with the majority of the authorship of the papers presented as a Thesis by Publication undertaken by the Student. The work is original and has not previously been submitted for any other award, except where acknowledged.

Signed
Mohammed Diykh
Date:

Signed
Principle Supervisor
Dr. Shahab Abdulla

Signed
Associate Supervisor
Dr. Khalid Saleh

Student and supervisors signatures of endorsement are held at the University.

List of Contribution from Publication Co-authors

This section presents details of contribution by the various author for each of the paper presented in this thesis by publication. The following detail is the agreed share of contribution for candidate and co-authors in the presented publications in this thesis:

Chapter 2, Diykh et al., (2016a)

Diykh, M. & LI, Y. 2016. Complex networks approach for EEG signal sleep stages classification. *Expert Systems with Applications,* 63**,** 241-248. (Q1)

Author	Percent contribution	Tasks Performed
Diykh, M	70%	Designed the method, simulation, analysis, interpretation, wrote entire draft of paper
LI, Y	30%	Suggested edits to manuscript, interpretation

Chapter3, Diykh et al., (2016b)

Diykh, M., Li, Y. & Wen, P. 2016. EEG sleep stages classification based on time domain features and structural graph similarity. *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* 24, 1159-1168. (Q1)

Author	Percent contribution	Tasks Performed
Diykh, M	70%	Designed the method, simulation, analysis, interpretation, wrote entire draft of paper.
LI, Y and Wen Peng	30%	Suggested edits to manuscript, interpretation

Chapter4, Diykh et al., (2017)

Diykh, M., Li, Y. & Wen, P. 2017. Classify epileptic EEG signals using weighted complex networks based community structure detection. *Expert Systems with Applications*, 90, 87-100. (Q1)

Author	Percent contribution	Tasks Performed
Diykh, M	70%	Developed the method, simulation, analysis, interpretation, wrote entire draft of paper.
LI, Y and Wen Peng	30%	Suggested edits to manuscript, interpretation

Chapter5, Diykh et al., (2018)

Diykh, M., Li Y., Wen, P. & LI, T. 2018. Complex Networks Approach for Depth of Anaesthesia Assessment. Measurement. (Q2)

Author	Percent contribution	Tasks Performed
Diykh, M	70%	Developed the method, simulation performing, analysis, interpretation, wrote entire draft of paper,
LI, Y, Wen Peng and Li T	30%	Suggested edits to manuscript, interpretation, data acquisition.

The following papers co-authored by by Diykh, M during the period of candidacy, but not relevant to the main thesis are presented in Appendices A-C.

Sahi, A., Lai, D., Li, Y. & Diykh, M. 2017. An efficient DDoS TCP flood attack detection and prevention system in a cloud environment. IEEE Access, 5, 6036-6048. (Q1).

Al-Salman, W., Li, Y., Wen, P. & Diykh, M. 2018. An efficient approach for EEG sleep spindles detection based on fractal dimension coupled with time frequency image. Biomedical Signal Processing and Control, 41, 210-221. (Q2).

Al Ghayab, H. R., Li, Y., Abdulla, S., Diykh, M. & Wan, X. 2016. Classification of epileptic EEG signals based on simple random sampling and sequential feature selection. Brain informatics, 3, 85-91.

Acknowledgments

First of all, I would like to thank almighty Allah for his guidance and strength.

I want to express my thanks to my supervisory team Dr.Shahab and Dr.Khalid for their support and guidance.

I would like to offer my gratitude and thanks to Dr.Yan and Dr.Paul for their support, and guidance during my PhD study. Their guidance helped me to achieve my goals in my PhD journey.

I also want to express my personal gratitude to Sandara for proofreading my thesis.

I wish to introduce my appreciation to my honourable mother, for her support during my PhD study. There are not enough words to describe what a powerful influence she continues giving me.

I would like to dedicate this thesis to my Father who passed away when I was 22 years, and also, to my beloved family who supported me through my study.

I also want to express my thanks to republic of Iraq ministry of higher education and scientific research for supporting me.

Finally, I would like to acknowledge the support of the Australian Commonwealth Government through the Research Training Program (RTP) Fees Offset scheme during my research.

Contents

List of Figures
(Excluding publication included in Chapters 2-5)

List of Tables
(Excluding publication included in Chapters 2-5)

1

CHAPTER 1

INTRODUCTION

The human brain is considered the command centre for the human nervous system. It is a complex network composed of billions of neurons which are capable of processing information millions of times faster and more efficiently. In recent years, much research work has been conducted to analyse human brain activity using different techniques such as electroencephalogram (EEG), Electromyography (EMG) and electrooculography (EOG) signals. EEGs are a tool commonly used to measure the electrical activity generated by the brain's cerebral cortex nerve cells. They are measured using electrodes placed on the scalp. EEG signals record the electrical potentials generated by the nerve cells in the brain. Clinical research showed that EEG signals exhibit different patterns of waves depending on the state of a human; whether he/she is awake, asleep, anesthetized. Consequently, they are utilized as an important source for studying human brain activities and the diagnosis of several neurological diseases, such as sleep disorders, epilepsy, and monitoring depth of anaesthesia.

Although, the number of studies investigating EEG signals during sleep stages, epilepsy and anaesthesia is growing, more effort needs to be applied to improve its efficacy in terms of accuracy and time consumption (Al-Qazzaz el al., 2015; Herrera et al., 2013; Bankman et al., 1987; Henao-Idarraga). The main purpose of those studies is the analysis of EEG signals by separating EEGs into small intervals (segments). Then, the EEG segments are classified into different categories or states such as healthy and non-healthy, or to estimate the depth of anaesthesia (DoA).

Usually, EEG signals generate a huge amount of data. Visual analysis this data by experts or neurologists is error-prone, and requires a high cost and efforts (Kayikcioglu et al., 2015; Zhu et al., 2014; Zhang et al., 2016; Amin et al., 2016). Visual analysis is also a subjective process that means a decision made by two experts of EEG signals could be varied even for the same EEG recordings (Kemp et al., 2000; Lee et al., 2004). The demands associated with the development of accurate techniques for EEG signals analysis (to relieve the burdens of visual inspection) have intensified in the last decade. As a result, several single channel and multi-channel EEG analysis approaches have been established to computerise EEG signals analysis (Gao et al., 2016; Nguyen-Ky et al., 2013; Petsiti et al., 2015; Diykh et al., 2016a; Diykh et al., 2016b) . This focuses focus on analysing three most important types of EEG signals: epileptic, sleep and anaesthesia EEG signals. It proposes one algorithms to analyse and classify EEG sleep stages, one algorithm to detect epileptic seizures and one algorithm to monitor the depth of anaesthesia in EEG signals.

The proposed algorithm can detect and analyse the abnormalities in EEG signals as well as classify them into different categories. The proposed algorithms will be useful for identifying brain disorders and accurately monitoring a patient's state during surgery. The outcomes of this study will help doctors and neurologists in the diagnosis and treatment of brain disorders, and for clinical research.

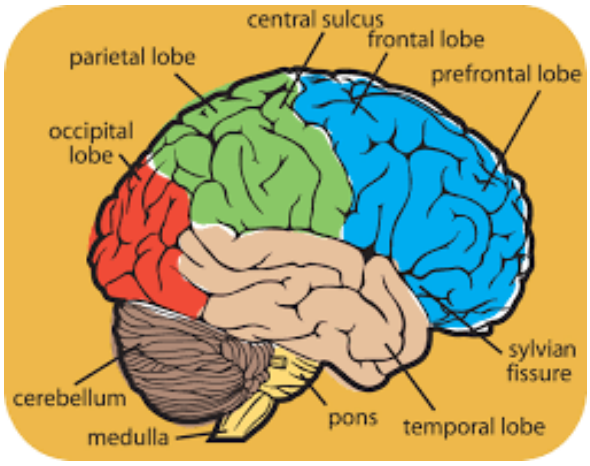


Figure 1.1: The main parts of the brain (Sanei and Chambers, 2007)

1.1 Human Brain

The brain is one of the most complex parts of the human body. It controls on the human nervous system. It receives commands from sensor organs and sends the output to neurons. The human brain consists of three main parts: cerebrum, cerebellum and brainstem, as illustrated in Fig1.1. A brief explanation of each part of the brain is presented below (Carlson, 2002).

The cerebrum is the most important part of the brain. It consists of right and left hemispheres. It performs the most important functions of the brain such as interpreting, thinking, speech emotions and body control. Each hemisphere is divided into four parts called lobes: the frontal lobe, partietal lobe, occipital lobe and temporal lobe. Each Lobe is associated with some functions; for example, the frontal lobe is associated with personality, speech and problem solving (Carlson, 2002, Purves et al., 2006).

The cerebellum is similar to the cerebrum in structure as it has two hemispheres, and it has a folded cortex. It is located in the area under the cerebrum. It is responsible for the control of muscle movement, balance and posture.

The brain stem is located underneath the limbic system. It works as a bridge to connect the spinal cord with the cerebrum. It is associated with the most important life functions such as breathing, mouth movement and consciousness. It is made of the pons, medulla and midbrain.

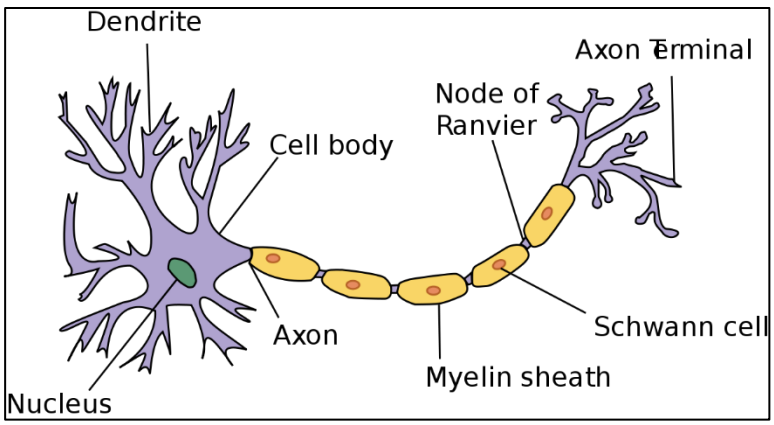


Figure 1.2: A structure of neuron (Sanei and Chambers, 2007)

1.2 Neural System

The brain is one of the most complex parts of the human body (Carlson, 2002, Purves et al., 2006). It consists of billions of neurons which maintain the electrical charge of the brain. A neuron consists of three main parts: cell nucleus, dendrites and long axon (Carlson, 2002, Purves et al., 2006). Figure 1.2 shows a structure of a neuron. Each part of a neuron is responsible for a specific task. The dendrite is a short-branched section which is located at the end of each cell. It consists of many receptors that receive a neurotransmitter from other cells. The long axon transmits the signal from one cell to another. The cell nucleus is the core of the cell providing the cells with instructions by which the cell takes an action.

Neurons form the main unit of the neural system. The neural system is responsible for three main functions: sensory input, integration and motor output. Figure 1.3 depicts the structure of the neural system. The sensory input recognises and monitors the environment and the changes happing with the human body. It describes responses in the skin, eyes, ears and nose (which are considered sense organs), when they receive stimuli. In the integration stage, the central nervous system (CNS), which is formed by the brain and spinal cord, manipulates the information received by sensory input and makes decisions. During motor output, the CNS sends a command to effector organs which could be a muscle or gland.

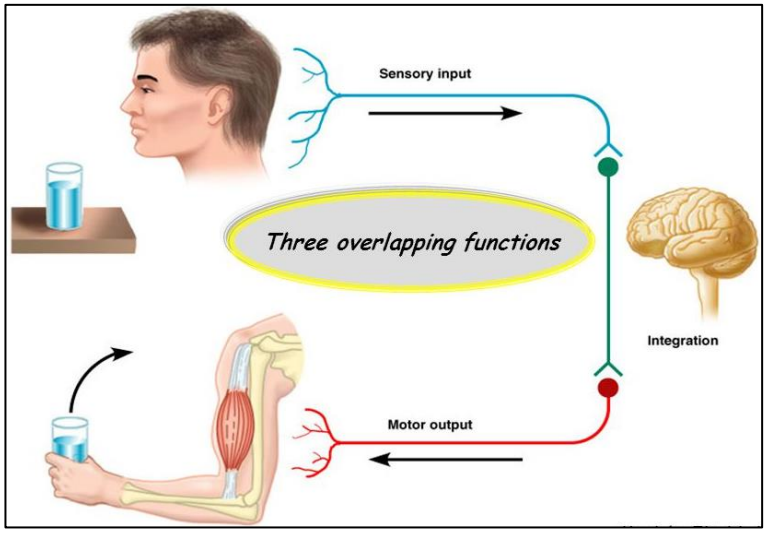


Figure 1.3: A neural system structure (Sanei and Chambers, 2007)

As previously mentioned, the CNS is comprised of two main units: the brain and spinal code. The brain is in control of most functions in the body including speech, memory, and thoughts. The brain is connected to the spinal cord through the brainstem. Figure 1.4 illustrates the nervous system structure. The spinal cord transfers signals between the brain and the body. As the CNS controls all body function, any dysfunctions in the spinal cord can disrupt the exchange of information between the body and the brain. Different lines of defences such as bones, spinal column protect the CNS from injury. In addition, a fluid between the bones absorbs shock and prevent damage to the CNS.

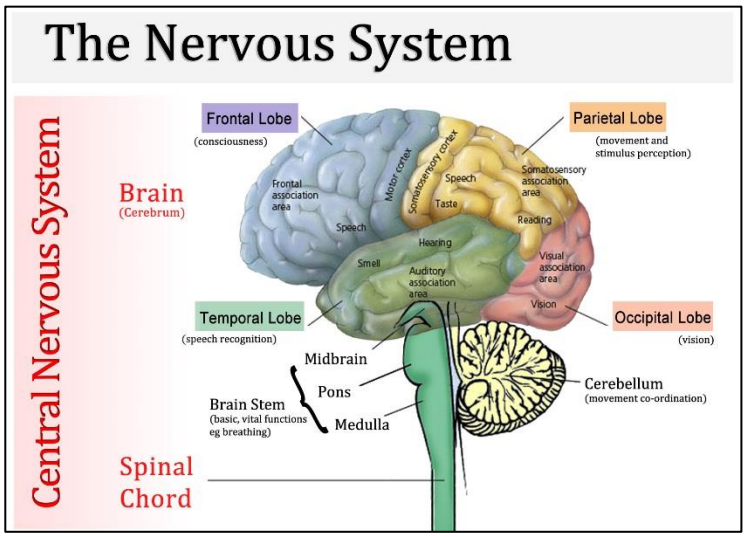


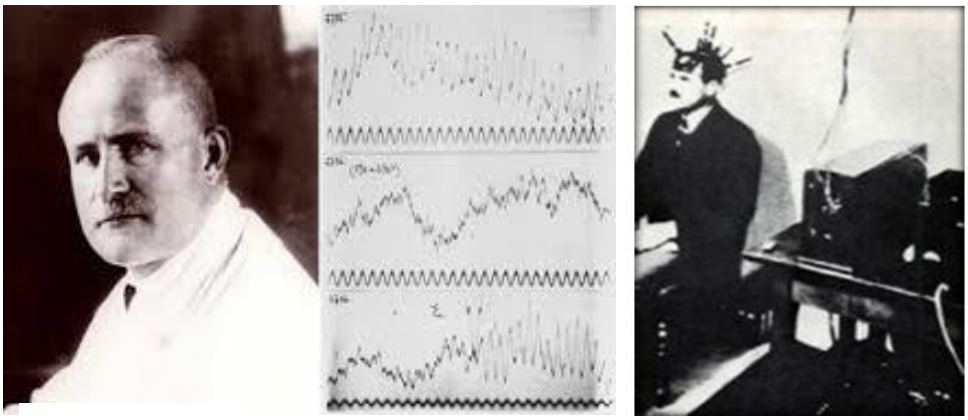
Figure 1.4: Nervous system (Chambers, 2007)

1.3 EEG Overview

The brain consists of two main cells called neurons or nerve cells, and glia which together form the brain tissue (Ramada n et al., 2015; Holmes et al., 2007; Mellinger et al., 2007). The brain tissue is surrounded by a cerebrospinal fluid (CSF) which protects the brain from injuries and strikes. CSF is an aqueous medium for salts of potassium, calcium and magnesium that play an important role in the function of the CNS. The brain tissue and neurons are connected through long cables called axons. These axons spread an electrical signal between the brain neurons and tissue in a non-attenuating manner. When electrical signals are reached, the area where neurons are connected, new axons called dendrites receive those signals. The dendrites are different from axons as they do not have insulating metrical needed for fast spike propagation.

The electrical activity of neurons is probably the most interesting aspect of brain activity (Rao et al., 2012; Van Erp et al., 2012; Felton et al., 2007). Generally, electrodes are placed on the surface of the brain, and can easily record both spikes and synaptic currents.

The cerebral cortex nerve cells of the brain generate variety of electrical brain activities which are recorded as EEG signals. EEG signals were first recorded in 1875 by Caton in 1875 (Collura 1993; Morshed et al., 2014; Lindsley et al., 1936; Loomis et al., 1935). He was able to record electrical activity from the brains of animals and his work was considered to be the first work to deal with brain activity. The study was conducted with over 40 cats, monkeys and rabbits. In this study, Caton observed variations in brain signals associated with different states such as, a sleep, wakefulness and anaesthesia. In 1924, Hans Berger from the University of Jena in Austria, recorded the first human EEG recordings from a 17 year old boy while he was undergoing a surgery to remove a suspected tumour (Millett et al., 2001; Gloor et al., 1994; Kiernan 1886). Figure 1.5 shows the first EEG recording.



Hans Berger | An EEG recording | Patient during recording EEG

Figure 1.5: An EEG signal is being recorded by Hans Berger (Berger, 1929)

Metal strips and a sensitive galvanometer were used as electrodes and recording instruments. In this, irregular and relatively small electrical potentials (i.e., 50 to 100 μV) were observed in the patient's recording.

However, with the developments in the digital technologies, EEG recording systems became more sophisticated. Different types of metal electrodes made from tin, gold platinum or silver, were designed to detect brain signals (Geddes et al., 2003; Tallgren

et al., 2005; Salvo et al., 2012). These electrodes were capable of measuring any small potentials produced by the brain neurons.

The standard system for recording the brain activity is called the 10/20 system (Teplan et al., 2002; Nicolas-Alonso et al., 2012; Mason et al., 2003). It is an international system which describes the locations of electrodes on the scalp. The 10/20 system was designed based on the relationships between the location of an electrode and the underlying area of the cerebral cortex. The numbers 10 and 20 come from the distance between each pair of electrodes that is either 10% or 20% of the total left-right or front-back of the human skull. Each one of those electrodes has a unique number and letter to identify lobe and hemisphere location respectively. Table 1.1 gives a short explanation of those locations.

Table 1.1: Observation for each lobe

Electrode	Lobe
F	Front
T	Temporal
C	Central
P	Parietal
O	Occipital

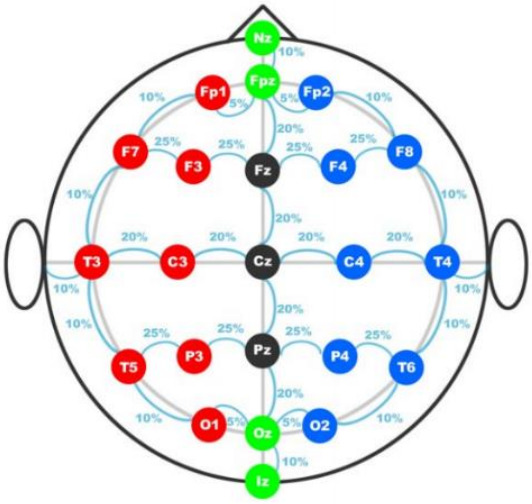


Figure 1.6: 10/20 System electrode distance

The electrodes are placed onto different locations of the scalp's surface. The location depends on the type of signals targeted. Each electrode is associated with an amplifier

and an EEG recording device. Figure 1.6 shows all possible of electrode locations on the scalp.

The acquired EEG signals are transferred into waves form by a computer. The number of possible electrodes that can be used to obtain signals is from 1 to 512 electrodes. These electrodes depend on the scalp location targeted and the required signals. Clinical research has shown that a multi-channel EEG signal can be obtained by employing more than one pair of electrodes while a single channel signal can be recorded using one pair of electrodes. Four forehead electrodes are mostly involved in recording anaesthesia EEG data.

To record sleep EEG signals, the electrodes Fpz-Cz / Pz-Oz are mainly used in recordin Figure 1.7: Different types of EEG rhythms (Lotte, 2008) nent such as C4-A1/ C3-A2 may sometimes be employed. For epileptic EEG signals, most of clinical studies use all the electrodes to record EEG signals (Fonseca et al., 2007; Klimesch et al., 1997; Klem et al., 1999; Kauhanen et al., 2006).

1.4 Nature of EEG Signals

EEG signal are mainly described using two terms of rhythmic activities and transients (Thut et al., 2012; Wang et al., 2010; Schnitzler et al., 2005; Zaehle et al., 2010). The rhythmic activities are classified into different bands based on frequency using a spectral method such as a wavelet transform or Welch. Most brain activities fall in the range of 1-20 Hz. Based on biometric research, the frequency bands are divided into different sub-bands including Delta, Alpha, Theta, Beta, Gamma and Mu rhythm (Liu et al., 2005; Vaughan et al., 1996; Wolpaw et al., 2000).

Delta: the frequency range of alpha activity laies between the range of 0.5-4 Hz with a highest amplitude and lowest waves. These type of waves are associated with deep sleep, awake stages and in some cases of brain disorder.

Theta: the frequency range of theta activity is between 4-8 Hz with an amplitude up to 20 μV. It is normally observed in young children and during drowsiness in adults. Theta activity increases in EEG signals during emotional states such as stress, frustration and disappointment.

Alpha: is in the frequency range of 8-13 Hz with amplitude of 30-50 μV. This range of frequency appears in the posterior region of the head on the both sides. This type of

activity is called posterior basic rhythm and it appears when the subject has closed eyes and is in a state of relaxation.

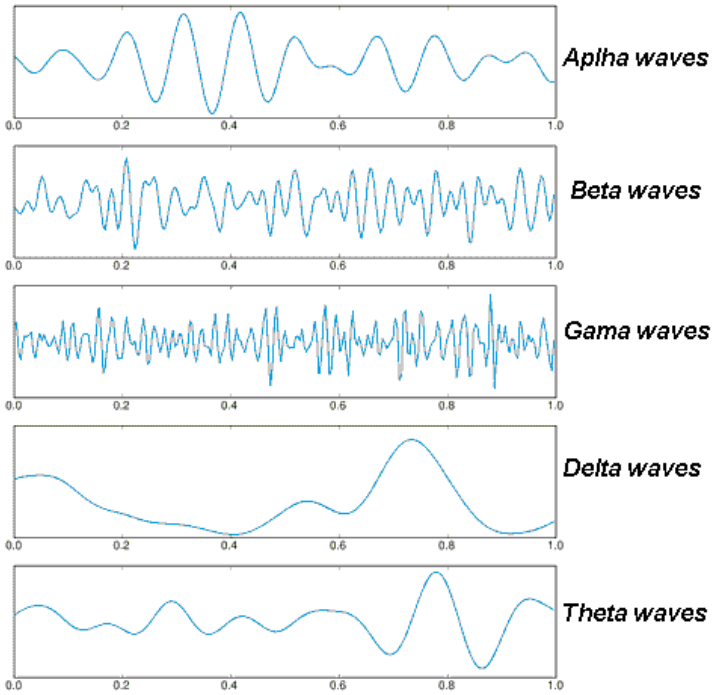


Figure 1.7: Different types of EEG rhythms (Lotte, 2008)

Beta: the frequency range of beta is between 13-30Hz. It is recorded from both sides of the frontal area with low amplitude and varying frequencies. This type of activity appears with motor behaviours and during active movements.

Gama: it appears during cognitive and motor functions. The frequency range of the gama activity is 30Hz and up.

EEG signals exhibit different features and patterns such as sleep spindles, k-complexes during sleep stages, spike, and sharp waves during epileptic seizures. These patterns are transients rather than rhythmic. Detecting them is much harder than detecting rhythmic activities.

As mentioned previously, this thesis focuses on developing robust algorithms to analyse and classify three types of EEG signals: epileptic, sleep and anaesthesia EEG signals. The following section provides a brief explanation for each of these signals and how they affect human brain nerves.

1.5 Effects of Epileptic Seizures, and Epilepsy on EEG Signal Behaviour

Epilepsy is a chronic neurologic disorders characterized by recurrent unprovoked and seizures. It affects more than 40 million people around the world; most of them young children and older adults. Developing countries contribute about 85% of epilepsy cases. Several clinical studies have shown that epilepsy can be caused by interactions between several genes and environmental factors. However, people how suffering a brain damage, stroke, toxicity and high fever can endure epileptic seizures. During an epileptic seizure, the brain neurons discharge suddenly, and abnormal activities occur within the cerebellum and cortex. Figure 1.8 shows differences between normal and epileptic EEG signals. A seizure usually lasts about two minute or less depending on the age of the patient and the health of the brain.

Epileptic seizures are classified into two different categories; partial and generalized seizures. The partial seizure effects a particular part of brain while the generalized seizure involves the entire brain. During strong seizures patients can lose consciousness for several minutes and cannot recall what has happened. Epileptic seizures can be controlled by medications and in some severe cases surgery can be a solution. However, challenges remain in curing all epilepsy patients as medication can control only 50-60% of cases.

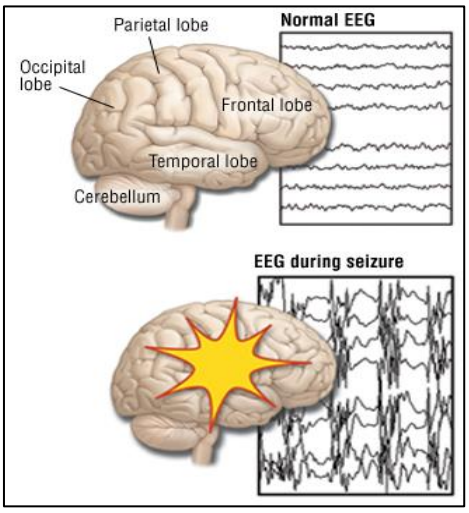


Figure 1.8 Different between normal and epileptic EEG (Harvard health online)

The EEG is a common and widely used tool employed to look at the electrical activity of the brain and the functioning of brain neurons. An abnormality in brain activity can

be detected by an EEG test. The EEG test takes one hour and is usually done with open and close eyes. The advantage of EEG tests is that they enable neurologists and experts to determine what happened during seizure attacks and to compare this with normal EEG recordings. Developing accurate techniques to detect epileptic seizures in EEG signals is vital. In this thesis, a robust methodology is developed to detect the abnormalities in patients' EEG recordings, thus, helping neurologists with early seizure detection.

1.6 Human Sleep and Sleep EEG Signals

Sleep is a dynamic process involving two main stages: rapid eye movement (REM) and non-rapid eye movement (NREM) (Giri et al., 2015; Lanjnef et al., 2015; Lee et al., 2004). The later includes four stages: Stage 1 (S1), Stage 2 (S2), Stage 3 (S3), and Stages 4 (S4). Figure 1.9 shows a normal sleep cycle.

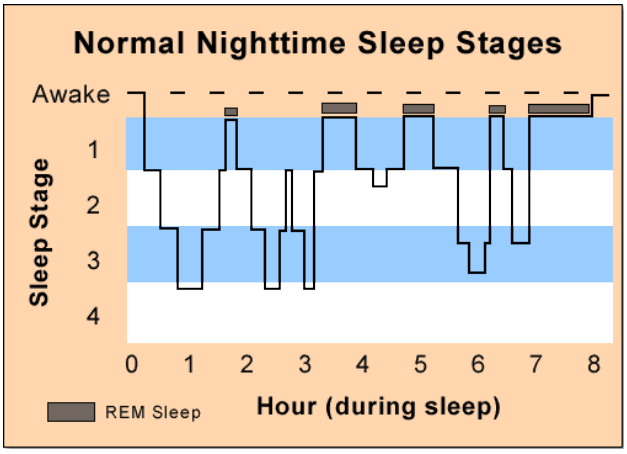


Figure 1.9: A human sleep cycle (Holly Nelson, 2015)

These individual sleep stages are connected through different physiological and neuronal characteristics that are used in sleep identification by experts and researchers. The process of visually discriminating between sleep stages is called sleep scoring or sleep staging. Sleep scoring is normally carried out visually by experts with reference to either the Rechtschaffen, and Kales (R&K) (Rechtschaffen et al., 1968), or the American Academy of Sleep Medicine's (AASM) (Berry et al., 2012) guidelines. The R&K guidelines divide sleep recordings into seven different stages namely: awake (AWA), S1, S2, S3, S4, REM and movement time. The R&K guidelines have been used as the standard for sleep scoring for the last 40 years. It has received much

criticisms for leaving too much room for subjective interpretation with a wide variability in the visual evaluation of sleep stages occurring. In 2007, the AASM guidelines were modified to address some issues in the R&K guidelines. In the AASM guidelines the allocated time for S1 and SWS were changed, and a minimum of three EEG derivations including F4-M1, C4- M1, and O2-M1 from the frontal, central, and occipital regions are recorded. The AASM combines S3 and S4 into one stage, and also considers body movement as a sleep stage.

Stage 3 (S3) and Stage 4 (S4) are considered to be deep sleep that occurs in the first half of the night. During these stages, the human body starts to release hormones that restore muscles damaged from stress and fatigues. Therefore, one is not easily wakened during these sleep stages. These two sleep stages are categorised as slow wave sleep, and the corresponding EEG waves have high amplitudes. The human brain produces similar waves during S3 and S4, and 50% of these are delta waves making it difficult to distinguish them.

The REM stage is associated with a unique brain wave pattern, and the brain waves exhibit a combination of alpha, beta, and de-synchronised waves. During the REM stage, the skeletal muscles are effectively paralysed and breathing becomes more rapid, irregular and slow. During this stage, waking can easily happen and dreams can be remembered if the waking period is too long.

During the AWA stage brain waves become slower, and more synchronous, with an increased amplitude. Thus, EEGs during REM and AWA exhibit different characteristics making the separation of these stages more accurate. The brain waves during S1 transit from unsynchronized beta (12-30Hz) and gamma (25-100Hz) waves to more synchronised beta and gamma waves. During this stage, blood pressure and brain temperature fall. The range of theta waves during S2 are similar to those in S1, and the blood pressure also fall. The main differences between S1 and S2 are that the brain waves during S2 pass through two distinguishing phenomena: sleep spindles and K-complexes. Sleep spindles are defined as short bursts of brain activity in the range of 12-14 Hz for about half a second, while K-complexes exhibit short negative high voltage peaks followed by slower positive complexes.

Developing accurate methods to score a patient's EEG recordings could help neurologists diagnose sleep problems early. In this thesis, a novel technique is developed to classify EEG sleep stages using complex networks coupled with time

domain features. The proposed method shows a high quality in separation an EEG signal into different sleep stages.

1.7 Depth of Anaesthesia and EEG Signals

Monitoring depth of anaesthesia (DoA) during surgical operations is an going challenge. Developing an accurate index of the DoA helps correctly deliver anaesthesia agents to patients, reduces the costs associated with anaesthesia medication and prevents unintended intraoperative awareness. The majority of the devices used to assess DoA are based on clinical signs such as heart rate, blood pressure and sweating. These signs cannot estimate the DoA precisely, and are not reliable. The use of some types of anaesthesia medications, such as muscle relaxants, can make the interpretation of these signs difficult. Clinical research has revealed that abnormalities in these signs are not always present for patients who endure awareness during surgery.

The clinical studies show that, after applying an anaesthesia medication to a patient, the drug targets the central nervous system (CNS) where EEG signals are generated. During anaesthesia, EEG signals patterns change. As long as the level of anaesthesia changes, the amplitude and frequency of EEGs fluctuate rapidly and exhibit changes of the anaesthesia depth. This is the main reasons for the wide use of EEG signals to monitor DoA. As a result, many of the devices developed to monitor DoA are based on EEG signals.

One of these devices, the Bispectral index (BIS) is the most popular. Figure 1.10 illustrates the BIS monitoring system used to assess DoA. The BIS is a complex EEG parameter designed using a combination of power spectrum analysis and time domain analysis. The values of the BIS are between 0 and 100, and change over time. Figure 1.11 shows the BIS index range.

The BIS was inspired by a 1960 geophysics study monitoring the motion of ocean changes due to atmospheric pressure and seismic activity. Following this study, the scope was extended to the study of EEG signals, in particular the coupling of waking and sleeping frequencies.

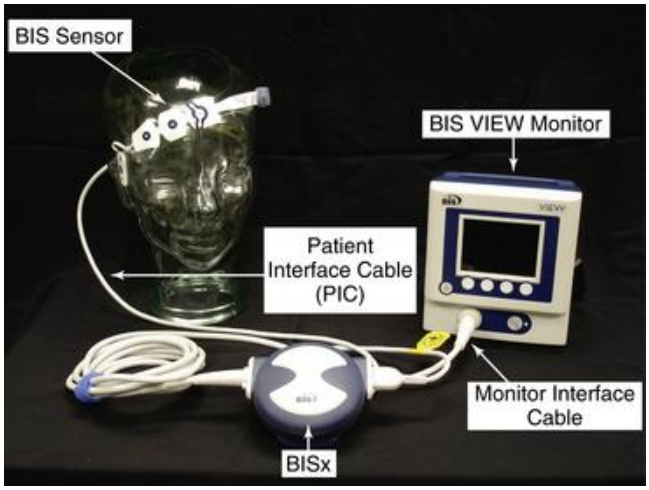


Figure 1.10: The BIS monitoring system (Yeung et al., 2010)

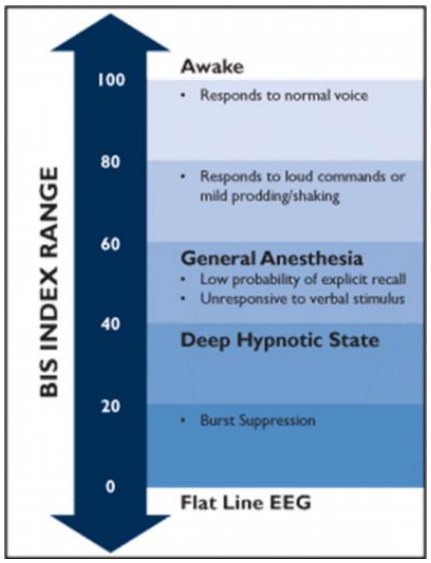


Figure 1.11: The BIS monitoring system index range (Schwary et al., 2005)

In 1992, the BIS was designed by Aspect Medical Systems and it was approved by the Food and Drug Administration as a measure of the depth of anaesthesia induced by sedatives and hypnotics in 1996.

Although the BIS has been used as a standard device to monitor DoA, it has received much criticism such as being delayed, not robust with different medications, and not sufficiently accurate across patients. However, developing an accurate index to monitor DoA is important for the delivery of an accurate amount of medication to patients and helping anaesthesiologists trace the patient's anaesthesia state during

surgery. In this thesis, a new EEG signals based method is developed to trace the anaesthesia state of patients. A statistical model and spectral graph wavelet transform (SGWT) are used to monitor DoA. The results demonstrate that the new index has a high potential to monitor DoA.

1.8 Research Problems and Hypothesis

The main goals of this thesis are to develop novel approaches to discover the human brain and to find out more efficient methods of identifying brain disorders using EEG signals. The performance of the developed methods are compared with their counterparts in time, frequency and hybrid domains. Three types of EEG data are analysed and studied: sleep, epileptic and anaesthesia EEGs. Undirected complex networks, and weighted undirected complex networks are used to analyse epileptic and sleep EEGs. Anaesthesia EEG signals are analysed using spectral graphs wavelet transform. The objectives are as follow:

1. How to efficiently classify EEG sleep stages based on time domain features and undirected complex networks attributes.
2. How to apply community detection algorithms in weighted networks to identify epileptic seizure in multi channels EEG signals.
3. How to assess depth of anaesthesia based on the spectral graph wavelet transform.

The hypothesis is that classification and analysis of EEG signals can be improved using complex networks characteristics. Based on simulation results, complex networks characteristics exhibit different characteristics when they are evaluated from various EEG signals such as sleep, epileptic and anaesthesia EEG signals.

1.9 Contribution of Each Chapter

The work presented in this thesis focuses on how to study human brain behaviour using complex networks through different EEG signals. The behaviour of the human brain in healthy and non-healthy subjects is studied and investigated with different EEG signals acquired from different channels. Different types of brain networks are evaluated through epileptic, sleep and anaesthesia EEG signals. Undirected complex networks, weighted complex networks, spectral graph wavelet transform are employed to study these EEG signals. To investigate the performance of these brain networks in

EEG analysis, thorough investigations are made through the design of extensive experiments. The following contribution are proposed:

1. Development of a structural graph similarity combined with k-means approach to classify and identifying EEG sleep stages using undirected complex networks coupled with time frequent features.
2. Improvement of developed method in No.1 by investigating new complex networks attributes that can improve the classification accuracy of EEG sleep stages.
3. Detection and classification of epileptic seizures in EEG signals based on community detection in weighted complex networks.
4. Implementation of spectral graph wavelet transform to assess the DoA.

These algorithms are entered into Matlab R2013b, and Spectral Graph Wavelet Transform Toolbox[1] and Matlab Tool for Network Analysis[2] are used in designing algorithms. Each algorithm is simulated and evaluated using different EEG signals acquired from different channels. A brief discussion of these four contributions is provided below.

1.9.1 Sleep stages classification based on complex networks

A robust approach based on time domain features and complex network attributes to score EEG sleep stages has been designed. First, sleep EEG signals are segmented into 30 second epochs. Each 30 second epoch is further divided into 75 sub-segments using a proposed segmentation technique. Different sets of statistical features including six, nine and twelve features set, are extracted and investigated from each sub-segment. As a result, each EEG segment is represented by a vector of statistical features. The resulting vector of statistical features is then mapped into an undirected complex network. The results showed that twelve features set gives a high classification accuracy than other sets. In this method, two networks attributes of (Jaccard coefficients, and average degree) are extracted and fed to a classifier. The relationships between graphs behaviours and time domain features are investigated through complex networks. Experimental results show that complex networks constructed

[1] http://wiki.epfl.ch/sgwt
[2] http://strategic.mit.edu/downloads.php?page=matlab_networks

from time domain features are capable of discriminating between the EEG sleep stages.

To increase the accuracy of the EEG sleep stages classification and improve the proposed method, new structural network attributes are investigated and used in a new study. In this study, the same methodology is used. However, different EEG datasets acquired from different EEG channels, Pz-Oz and c3-A1, are used to assess the proposed method with different EEG data.

The degree distribution and clustering coefficients as well as Jaccard coefficients are extracted from each complex network and forwarded to different machine learning algorithms. The network attributes are extensively investigated. The obtained results show that the new complex network attributes increased the classification accuracy and effectively discriminated between the EEG sleep stages compared with the Jaccard coefficients and average degree.

1.9.2 Identifying epileptic seizure in EEG signals based on community structure detection in weighted complex networks

To detect epileptic seizures in EEG signals, weighted complex networks based on community detection are applied to classify seizure from non-seizure EEG events. Five types of EEG signals of healthy with eyes open, healthy with eyes closed, inter-ictal from the epileptogenic zone, inter-ictal from the non-epileptogenic zone and seizure are investigated. The dimensionality of each single channel EEG signal is reduced using a statistical model. As a result, each single EEG channel is represented by 1536 features, and then mapped into a weighted undirected network. To detect the epileptic seizure, the modularity of networks is investigated with other network attributes including average degree, closeness and clustering coefficients. All the network attributes are ranked based on their potential to detect abnormalities in EEG signals. Four well known classifiers, a least squear support vector machine (LS-SVM), K-means, Naïve Bayes, and K–nearest- neighbours, are used classify networks attributes. The proposed method is tested using eight pairs of combinations of EEG signals. The results demonstrate that the proposed method is efficient in detecting epileptic seizures in EEG signals.

1.9.3 Assessment the depth of anaesthesia using spectral graph wavelet transform

Anaesthesia EEG signals are also studied and investigated using a statistical model and spectral graph wavelet transform (SGWT). Based on previous studies, it was found that using complex networks could reveal the hidden patterns in EEG signals. In the developed method, the de-noised EEG signals are partitioned into segments using a sliding window technique, each segment is then divided into sub-blocks to make the signal quasi stationary. A set of 10 features is extracted from each block to construct a vector of statistical features. Each vector of statistical features is mapped into a weighted complex network. Then the graph wavelet transform is applied. The total energy of wavelet coefficients at scale 3 is used to form an index to monitor the depth of anaesthesia. As a result, a new index called $SGWT_{DoA}$ is designed and evaluated using the EEG recordings and bispectral (BIS) from 22 subjects. The proposed index is evaluated and tested using different statistical metrics such as Bland-Altman, regression, Q-Q plot and Pearson correlation. The results demonstrated that the $SGWT_{DoA}$ reflects the transition from unconsciousness to consciousness efficiently.

2

CHAPTER 2

COMPLEX NETWORKS APPROACH FOR EEG SLEEP CLASSIFCIATION

Sleep disorders have a severe effect on the health and quality of human life. Sleep scoring plays an important part in the diagnosis of sleep disorders such as apnea insomnia. To relieve the burdens (e.g. high cost and human error) of visual inspection, a number of automated sleep staging techniques have been developed (Zhu et al., 2014).

Dimension reduction and features extraction are the most commonly used core steps in the development of sleep staging techniques. Computation speed and the quality of the classification results are dependent on how those steps are efficiently undertaken. The majority of sleep scoring methods have been developed under one of the major four domains: time, frequency, graph and hybrid domains (Uğuz, H. 2012); Fraiwan et al. 2010; Şen el al., 2014; Zhu et al., 2014).

According to previous research, graph models provide a simple and popular way to study human brain functions. They have been used as a useful and efficient tool for the investigation of brain disorders such as hyperactivity disorder and Alzheimer Disease.

Diykh et al. (2016) developed a new technique to score EEG sleep stages. Complex network attributes based on time domain features were used to identify the six sleep stages. Two graph attributes were investigated including average degree and Jaccard coefficients. One of the important findings of this research paper is that combining time and graph domains to identify sleep stages achieved a high classification accuracy. The paper's results also show that using degree and Jaccard coefficients together also improves classification accuracy. An average sensitivity and accuracy of 91% and 92% was obtained respectively when these graph's attrivutes were used togather. To compare the proposed method with the SVM, the extracted time domain features were forwared to the proposed method as well as to the SVM. The kernel function of SVM used in this experiment was RBF. To investigate the performance of the SVM, different kernel functions were tested, and it found that the RBF gave a high classification rate than other kernel functions

We followed the same methodology in both Chapter 2 and Chapter 3, however, the main differences between them are that in Chapter 3 new network's attributes are investigated and used to identify EEG sleep stages. It is noticed that the performance of the proposed model is improved significantly. In addition, extensive simulations and experiments are made to explorer the relationship between time domain features and networks behaviours. Also, two datasets, which were acquired from two different EEG channels, were utilized to examine the quality of the proposed method.

2.1 Diykh et al.,(2016)" COMPLEX NETWORK APPROACH FOR EEG SLEEP STAGES CLASSIFCIATION"

The manuscript prepared by Diykh at al., (2016), "Complex network approache for EEG sleep stages classification"

Expert Systems With Applications 63 (2016) 241–248

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Complex networks approach for EEG signal sleep stages classification

Mohammed Diykh*, Yan Li

School of Agricultural, Computational and Environmental Sciences, University of Southern Queensland, Australia

ARTICLE INFO

Article history:
Received 11 March 2016
Revised 28 June 2016
Accepted 1 July 2016
Available online 5 July 2016

Keywords:
Electroencephalography
Complex networks
Sleep stages
Statistical features

ABSTRACT

Sleep stage scoring is a challenging task. Most of existing sleep stage classification approaches rely on analysing electroencephalography (EEG) signals in time or frequency domain. A novel technique for EEG sleep stages classification is proposed in this paper. The statistical features and the similarities of complex networks are used to classify single channel EEG signals into six sleep stages. Firstly, each EEG segment of 30 s is divided into 75 sub-segments, and then different statistical features are extracted from each sub-segment. In this paper, feature extraction is important to reduce dimensionality of EEG data and the processing time in classification stage. Secondly, each vector of the extracted features, which represents one EEG segment, is transferred into a complex network. Thirdly, the similarity properties of the complex networks are extracted and classified into one of the six sleep stages using a k-means classifier. For further investigation, in the statistical features extraction phase two statistical features sets are tested and ranked based on the performance of the complex networks. To investigate the classification ability of complex networks combined with k-means, the extracted statistical features were also forwarded to a k-means and a support vector machine (SVM) for comparison. We also compare the proposed method with other existing methods in the literature. The experimental results show that the proposed method attains better classification results and a reasonable execution time compared with the SVM, k-means and the other existing methods. The research results in this paper indicate that the proposed method can assist neurologists and sleep specialists in diagnosing and monitoring sleep disorders.

1. Introduction

Sleep is an important physiological activity which plays a crucial role in self-repairing and self-recovery for human body. It is a cyclic process that involves various stages regulated by a specific system (Šušmáková & Krakovská, 2008). Monitoring and analysing sleep are very important to human health.

Rechtschaffen and Kales (1968) suggested a new sleep scoring visual inspection method according to the change in the physiological signals. This method has been used for sleep staging for about 40 years. Based on the R&K, an overnight sleep is partitioned into 20–30 s segments and EEG-rhythms with other parameters are calculated for each segment individually. The results are interpreted and each segment is classified into one of the seven sleep stages: awake, rapid eye movement sleep (REM), four non-rapid eye movement (NREM) numbered from 1 to 4 and body movement stage. In 2007, American Academy of Sleep Medicine (AASM) released the sleep scoring criteria (Iber, Ancoli-Israel, Chesson, & Quan, 2007) where the sleep stages are divided into five stages. The main differences between the R&K and the AASM are that the body movement stage is considered as one sleep or awake stage in the AASM. Furthermore, the NREM3 and NREM4 are combined into a single sleep stage in the AASM criteria. However, using these methods for sleep scoring is considered time consuming and also it is prone to human errors (Guo, Rivero, Seoane, & Pazos, 2009; Şen, Peker, Çavuşoğlu, & Çelebi, 2014; Zhu, Li, & Wen, 2014). To overcome the tedious of these methods, many automatic approaches have been developed to classify sleep stages.

Numerous research on developing automatic sleep classification methods use EEG, Electrooculography (EOG) and Electromyography (EMG) signals (Bajaj & Pachori. 2013; Ishizaki, Shinba, Mugishima, Haraguchi & Inoue, 2008; Micheloyannis et al., 2006; Şen et al., 2014; Van Straaten & Stam 2013; Zoubek, Charbonnier, Lesecq, Buguet, & Chapotot, 2007). Different techniques such as support vector machines (SVMs), neural networks and fuzzy systems have been developed to classify sleep stages (Adnane, Jiang, & Yan, 2012; Fang & Wang 2014; Güneş, Dursun, Polat, & Yosunkaya, 2011; Uguz, 2012). Most of EEG research aiming at sleep stages classification consist of two phases: (i) features extraction from EEG data, (ii) categorisation of the extracted features using a classification method. These research strode an important step to develop sleep scoring techniques. However, af-

* Corresponding author.
E-mail addresses: mohammed.diykh@usq.edu.au (M. Diykh), Yan.Li@usq.edu.au (Y. Li).

http://dx.doi.org/10.1016/j.eswa.2016.07.004
0957-4174/

ter decades of investigations, many problems still need to be solved. Improving the accuracy of classification and reducing the complexity time are two main challenges in sleep stages classification (Xiao, Yan, Song, Yang, & Yang, 2013). Further exploration on employing more effective features extraction and classification methods should be made to improve classification performance.

Over the last decade, there has been a growing interest in using complex networks to analyse different time series, such as biomedical and brain signals. (Ahmadlou, Adeli & Adeli, 2012; Liu, Zhou & Yuan, 2010; Panzica, Varotto, Rotondi, Spreafico, & Franceschetti, 2013; Xu, Zhang & Small, 2008; Zhu et al. 2014) analysed sleep stages based on visibility graphs. Zhang and Small (2006) studied behaviours of human speech and electrocardiogram (ECG) time series using complex networks. A complex network is a mathematical model with relational information that can be described by a graph. Graph theory provides a method to capture the topology of a network and to extract the main characteristics across networks which can help better understand the relationships between networks.

In this study, an automatic sleep classification method is proposed, which employs statistical features, complex networks and a classifier for classifying an EEG signal into six sleep stages. Each segment of 30 s EEG data was divided into 75 sub-segments. The number of sub-segments was empirically determined. Then a vector of statistical features was extracted from each segment and mapped into a complex network. The average degree and Jaccard similarity coefficients of each complex network were extracted and fed to a k-means classifier.

Two sets of statistical features were extracted, and evaluated in order to explore best representative features for each sleep stage, and to figure out the relationships between statistical features and network properties. The statistical features were ranked and then sorted in descending order based on the accuracy achieved for each feature. The first best nine and twelve features were selected as the key features for the experiments in this work.

The rest of the paper is organized as follows: in Section 2, we describe the prior studies in EEG signals classification. Section 3 depicts the datasets used in this paper. Section 4 illustrates the methodology of the proposed method. Section 5 discusses the experimental results. Finally, Section 6 draws the conclusion for this paper.

2. Related work

Automatic EEG sleep stages classification is a hard and complex problem that requires effective solutions at different levels of EEG processing, from the segmentation, the extraction and the selection of features, to classification. This section provides a brief summary of different techniques.

The Wavelet and Fourier transforms have been widely utilized in EEG sleep stages classification. Guo et al. (2009) used a relative wavelet energy to analyse epileptic EEG signals. The wavelet coefficients were extracted and forwarded to an artificial neural network. The method attained a 95% classification accuracy. Jo, Park, Lee, An, and Yoo (2010) classified sleep stages using EEG signals based on a genetic algorithm and a fuzzy classifier. The fast Fourier transform with a hanning window was utilized to obtain a power spectrum from the bandwidths (delta, theta, alpha, sigma, and beta). The fuzzy classifier combined with the genetic algorithm was used to classify the spectral features. Their method was implemented with four EEG recordings, which were obtained from four healthy youths between 27 and 29 years old. The method achieved an 84.6% classification accuracy. Doroshenkov, Konyshev, and Selishchev (2007) identified EEG sleep stages based on a hidden Markov model (HMM). Each row of an EEG signal was decomposed using a fast Fourier transform filter. Various features were extracted and then classified based on the HMM. The researchers used the PhysioNet international dataset to implement the method. The average accuracies of 88% and 74% for REM, awake and stage1 were achieved, respectively.

More recently, visibility graphs have been employed to identify sleep stages. Zhu et al. (2014) utilized the concepts of visibility graphs and horizontal visibility graphs to extract the features from single channel EEG signals. The different graph features were extracted, and then were forwarded to a support vector machine to classify the six sleep stages. The dataset used in the research was from the Sleep-EDF database (Europe data format). An average of 87% classification accuracy was obtained. Shi et al. (2015) developed multi-channel EEG sleep stages classification method. A collaborative representation model was utilized to develop a multi-learning algorithm. A k-means classifier and dictionary learning method were employed in classification phase.

However, other studies have been tried to employ hybrid features in sleep stages distinguishing. Herrera et al. (2013) extracted different combinations of EEG features by using different features extraction methods, such as wavelet transform, Hjorth parameters and symbolic representation. The extracted features were ranked using a normalized mutual information extraction and fed to a SVM machine. Furthermore, the classification results were improved by applying a stacked sequential learning approach. Şen et al. (2014) made a comparative study on sleep classification using different feature selection and classification methods. Four types of features: time domain features, frequency features, time frequency features and linear features were extracted. The extracted features were used as the input to five different classification algorithms: a random forest, a feed-forward neural network, a SVM, a radial basis function neural network and decision tree. The researchers conducted the comparison based on the datasets provided by St. Vincent's University Hospital and University College Dublin, the same datasets used in this paper.

From the literature, one of the limitations in the existing automatic sleep stages classification techniques is that they were used the same features to identify all the sleep stages. In fact, the features required for a desired accuracy is different. To find most discriminative features for each sleep stage, the extracted features were tested and evaluated in this work, and we found that not all sleep stages can be classified with the same number of features. Secondly, the current studies have been limited in classifying and analysing EEG data in time or frequency domain. However, in this work, we have combined the statistical features with the properties of complex network concept for sleep stages classification. Based on our knowledge, this approach has not been used in sleep stages classification before.

3. EEG data

The datasets used in this work were collected at St. Vincent's University Hospital and University College Dublin from different randomly selected subjects in observations over 6 months (Goldberger et al., 2000).

They are publically available.[1] All 25 subjects were selected from the database. Their demographic features were: 21 males and 4 females, age 50 ± 10 years, range 28–68, BMI: 31.6 ± 4.0 kg/m^2, range 25.1–42.5 kg/m^2; apnea/hypopnea (AHI): 24.1 ± 20.3, range 1.7–90.9. The Jaeger-Toennies system was utilized to obtain Polysomnograms records. 10–20 electrode placement system was a method used to describe the location of scalp electrodes. Each individual record in the database consisted of 2 channels EEGs (C3-A2 and C4-A1), 2 EOG channels and 1 EMG channel. The C3-A2

[1] http://www.physionet.org/physiobank/database/ucddb/

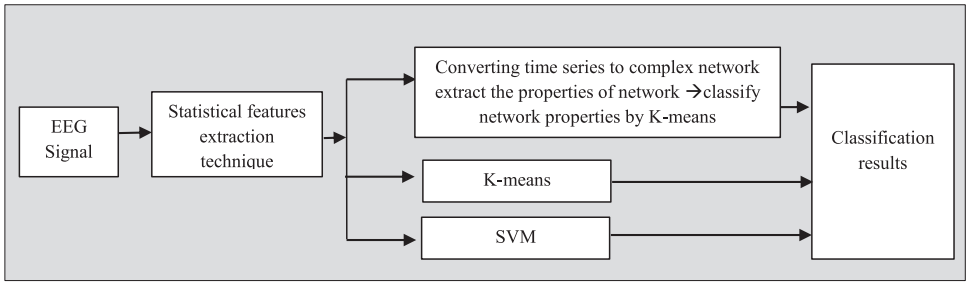


Fig. 1. Automatic sleep classification block diagram.

Table 1
The information of the experimental data.

Sleep stage	AWA	S1	S2	S3	S4	REM	Total
Number of segments	1109	897	988	1078	764	324	5160

EEG channel was used in this study because it gives better classification results compared with C4-A1 channel (Şen et al., 2014). According to the R&K criteria each EEG signal was divided into segments of 30 s (3000 data points), with each segment corresponding to a single sleep stage (AWA, REM, S1, S2, S3, S4, Artifact, Indeterminate). Table 1 presents the number of each sleep stage segments from the 25 subjects during 6.9 h period. In this study, all the segments in Table 1 were used in the experiments.

4. Proposed methodology

This paper proposes an efficient method to classify EEG sleep stages. Fig. 1 illustrates the structure of the proposed method. To decrease the dimensionality of EEG segments, each segment is partitioned into 75 sub-segments. 12 statistical features are extracted from each sub-segment, and then all the extracted features of each EEG segment are mapped into a complex network. The average degree and Jaccard coefficients are pulled out from each network and used as the key features. To investigate from the capability of complex networks builded from statistical features in sleep stages identification, the extracted features are forwarded to the k-means and SVM classifiers separately. The SVM and k-means are selected because they are popular and efficient (Guo et al., 2009; Herrera et al., 2013). The details of the methodology are explained in the following sections.

4.1. Feature extraction

Feature extraction methods aim to reduce the dimensionality of a large volume of signal data without losing the main information. In classification, feature extraction is a crucial step to obtain effective results. In this work, feature extraction is conducted in two stages through partitioning the original EEG data into segments and sub-segments. These stages are described in the following subsections.

4.1.1. Segmentation technique

As EEG signals are non-stationary and non-periodic (Kemp, Zwinderman, Tuk, Kamphuisen, & Oberyé, 2000), we apply the segmentation technique in order to make EEG signals quasi-stationary. The EEG signals are divided into small segments. The interval of each segment is defined as 30 s (3000 data points) in this work. Then, each segment is further divided into 75 sub-segments . The number of sub-segments and the length of each sub-segment is empirically determined during the experiments. To determine an optimum number of sub-segments that makes EEG signals relatively stationary we compute the number of sub-segments according to the following algorithm:

(1) $m = 1$
(2) $m = m + 1$
(3) Each EEG segment is divided into m sub-segments
(4) Extracting the statistical features from each sub-segment
(5) Forward the extracted features to the proposed method
(6) If the classification results are satisfactory then stop segmentation
(7) Else Go to step 2

As the result, each segment is divided into 75 sub-segments with each sub-segment includes 40 EEG data points. Fig. 2 shows the segmentation technique.

4.1.2. Statistical features extraction

After the segmentation, we used a statistical approach to extract the time domain features from each sub-segment. The feature extraction method can reduce the number of data points in each segment. This step is important because decreasing the number of nodes in complex networks can decrease the computational time of the proposed method.

In this paper, 12 statistical features are extracted from each sub-segment and used as valuable parameters, and all the features are put into one vector to represent each EEG segment. The statistical features are *{Mean, Min, Mode, Max, Median, Range, Variation, Skewness, Kurtosis, 1st Quartile, 2nd Quartile, Standard deviation}*.

The extracted features are also ranked according to the performance of the networks during the training phase, and then they are sorted in descending order. The top nine and twelve features are chosen as the most significant features. The order of the importance of the statistical features is {Mean, *Min, Mode, Max, Median, Range, 1st Quartile, 2nd Quartile, Standard deviation, Variation, Skewness, Kurtosis}*. Fig. 3 shows the importance of each feature based on their classification accuracy achieved in experiments.

4.2. Mapping statistical features to complex networks

A network can be described by a graph consisting of a set of nodes and a set of edges. The connection between each pair of nodes refers to the existence of a relationship between the nodes (Blondel, Gajardo, Heymans, Senellart, & Van Dooren, 2004; Nascimento & Carvalho, 2011; Stam & Reijneveld, 2007). In this work, the characteristics of the complex networks are employed to classify an EEG segment into one of the six sleep stages. To build the network, each vector of statistical features, which represent one EEG segment, is transferred into a graph. The structural properties of each graph are then calculated and forwarded to a classifier. Let $\mathbf{X}$ represents a vector of statistical features from an EEG segment $\{x_i\}_1^n$ of n observations. Firstly, each data point in $\mathbf{X}$ is considered as a basic node in an undirected graph.

The connection between two nodes is determined according to Zhang and Small (2006). A basic idea is that every two nodes in a

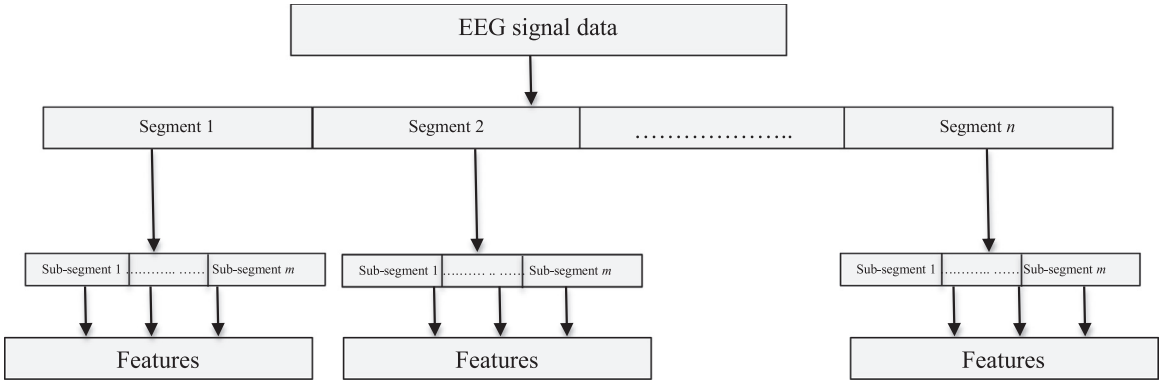


Fig. 2. Features extraction diagram.

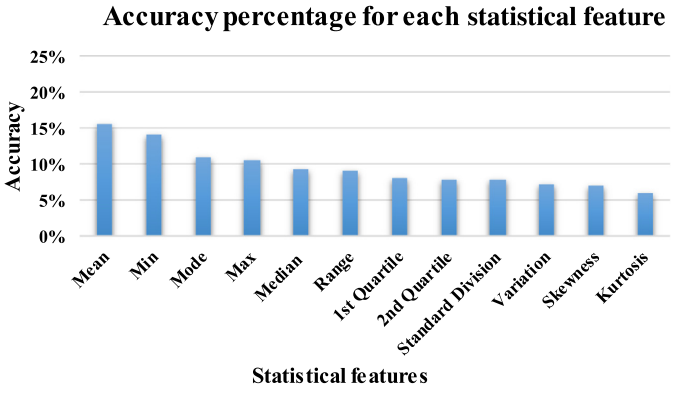


Fig. 3. Classification accuracy based on individual statistical feature.

graph are deemed to be connected if the space distance between the corresponding nodes is less or equals to the predetermined value (adaptive threshold $\boldsymbol{D}$).

$$(v_1, v_2) \in \boldsymbol{E}\ if\ d(v_1, v_2) < \boldsymbol{D},\ \text{where } \boldsymbol{D} \text{ is the adaptive threshold} \tag{1}$$

After a graph is constructed from the statistical features, an adjacent matrix is calculated based on the graph connection, which is defined as:

$$\boldsymbol{A}(v_1, v_2) = \begin{cases} 1, & \text{if } (v_1, v_2) \in E \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Fig. 4 shows a small segment of (2.4, 5.0, 6.2, 6.0, 7.5, 8.5, 3.0, 1.8, 9.5) when being transferred into a complex network.

The number of nodes increases when there are more sub-segments in one EEG epoch, and vice versa. All graphs have a fixed number of nodes.

4.2.1. Graph features

- **Average degree**

  The average degree (**AD**) refers to the number of links connecting a node in a network to other nodes. In an indirect network the total number of links $\boldsymbol{L}$, can be defined as:

$$\boldsymbol{L} = 1/N \sum_{i=1}^{n} k_i \tag{3}$$

  Where L represents the number of links, k_i is the degree of node ith and n is the size of the network.

  In Fig. 4 we have $k_1 = 1$, $k_2 = 2$, $k_3 = 3$, $k_4 = 2$, $k_5 = 3$, $k_6 = 2$, $k_7 = 3$, $k_8 = 0$, $k_9 = 3$, and $\mathbf{AD} = 2$.

  The average degree was used by Artameeyanant, Sultornsanee, and Chamnongthai (2015) to distinguish the sleep stages using weighted visibility algorithm, and by Zhang and Small (2006) to identity different dynamic regimes in time series. In this work, for each graph the degree distribution is calculated and considered as effective features for sleep classification.

- **Jaccard coefficient**

  We investigate the relations between nodes by using Jaccard coefficient. Jaccard coefficient was used by Anuradha and Sairam (2011) to classify digital images, and Iglesias and Kastner (2013) employed Jaccard coefficient to analyse similarity between two time series. The Jaccard coefficient function is calculated based on the following equation:

$$\boldsymbol{w}(v_i, v_j) = |\Gamma(v_i) \cap^{\Gamma} (v_j)| / |\Gamma(v_i) \cup^{\Gamma} (v_j)| \tag{4}$$

  Where $\Gamma(v_i)$ is the set of neighbors of node v_i, $\Gamma(v_j)$ is the set of neighbors of v_j, and $\boldsymbol{w} = [0, 1]$. For each graph, a Jaccard coefficient vector is computed.

4.3. Statistical analysis

In this study, the cross validation, sensitivity (SE) and accuracy (AC) are used to evaluate the performances (Siuly, Li & Wen, 2011; Zhu et al., 2014).

- **K-cross-validation**: is used to estimate the quality of classification. It is calculate by dividing a dataset into k mutual exclusive of subsets of approximately equal size and the method repeated k-times.

$$Performance = 1/K \sum_{1}^{k} accuracy^{(k)} \tag{5}$$

- **Sensitivity**: is a statistical measure which is used to evaluate the performance of a classification tested by measuring the proportion of the actual positive classification.

$$SE = TP/(TP + FN) \tag{6}$$

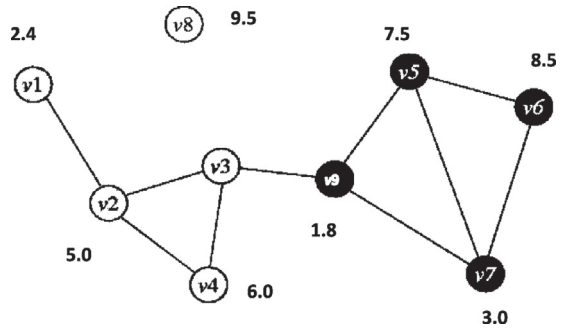
- **Accuracy**: is used to calculate the proportion of the total number of predictions that are correct.

$$AC = (TN + TP)/(TN + FN + FP + TP) \tag{7}$$

Where TP = correctly classified EEG sleep stages, FN = incorrectly classified EEG sleep stages, TN = the number of true negative classification, FP = the number of true positive classification.

4.4. Classification

This section discusses the two algorithms, the SVM and k-means are used in this paper to classify the EEG data, due to



Degree distribution

Degree	Frequency
0	1
1	1
2	2
3	5
4	0
5	0
6	0

Fig. 4. A complex network representing a small segment of EEG features, with nodes representing data (features) and a line represents an edge between nodes, with degree distribution.

their popularity in biomedical data classification (Güneş, Polat, Yosunkaya, & Dursun, 2009; Zhou, Gan, & Sepulveda, 2008; Zhu et al., 2014).

4.4.1. Support vector machine

SVMs have attracted a great deal of attention. The first paper for the SVM was proposed by Boser, Guyon, and Vapnik (1992). It transforms the original data into a high dimension by using a non-linear mapping. Within the new dimension, it searches for the linear optimal separating hyper plane. Zhu et al. (2014) and Herrera et al. (2013) used the SVM classifier to classify EEG sleep stages. In this work, a SVM machine was used as a standard classifier to compare with the proposed method in EEG statistical features classification. The SVM function in Matlab R2013 is used in this paper.

4.4.2. K-means algorithm

K-means is considered as one of most powerful methods in data classification. The algorithm partitions n observations into k groups, so that each observation belongs to one group with the nearest centroid. It has been widely used to classify data in different areas, such as digital images classification, time series and biomedical data analysing (Liao, 2005). In this paper, k-means function in Matlab R2013 is used.

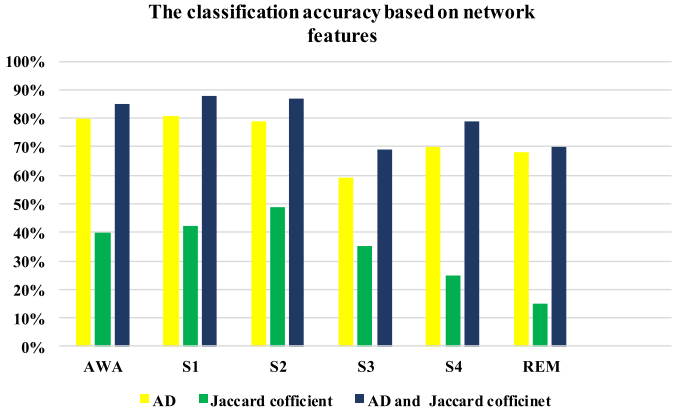


Fig. 5. Accuracy based on complex network properties.

Table 2
Classification accuracy of the proposed method based on 9 features.

Sleep stage	AWA	S1	S2	S3	S4	REM	Average
Accuracy	86.0%	88.0%	87.0%	69.0%	79.0%	70.0%	80.0%
Sensitivity	87.0%	88.0%	87.5%	67.0%	79.0%	71.0%	79.8%

5. Experimental results

In this section, the efficiency of the proposed method to classify sleep stages was evaluated. Two different groups of statistical features: the first nine and all twelve statistical features from each EEG sub-segment as discussed in Section 4, were used. The main purpose of selecting different features sets was to examine the ability of building statistical features as complex networks on EEG sleep stages classification. A vector of statistical features, which contains 900 features (75×12), was also forwarded to a SVM and a k-means classifier for result comparisons. The experiments were conducted using Matlab (Version: R2013) on a computer with the following features: 3.40 GHz Intel(R) core(TM) i7 CPU processor machine, and 8.00GB RAM. All the experiments by the proposed method, k-means and the SVM were evaluated using accuracy and sensitivity.

5.1. EEG sleep stages classification using different number of features

Two different sets of statistical features were used in this section. The performance of the proposed method based on two sets of features was assessed.

5.1.1. Experiment 1: using the first nine statistical features

In this experiment, the top nine features of {*max, min, mode, median, range, 1st quartile, 2nd quartile, standard deviation*}, based on their importance as shown in Fig. 3, were used to represent the original EEG data in each sub-segment. A total of 675 (75×9) features were extracted from each 30 s segment and were then transferred into a complex network. The average degree and Jaccard coefficient from each graph were calculated and forwarded to the k-means classifier.

Firstly, we tested the average degree to investigate how the average degree of a graph was used to distinguish sleep stages. From the obtained results, we observed that, the average degree was a key feature to significantly identify the sleep stage AWA, but not sufficient to represent the other sleep stages {S1-4, REM}. The results obtained were compatible with the results that were reported by Artameeyanant et al. (2015) and Zhu et al. (2014). The AWA sleep stage included the low activity compared with other sleep stages.

Secondly, Jaccard coefficient was calculated to investigate the effectiveness of the relationship between two nodes in sleep stages classification. According to the results, we observed that the Jaccard coefficient had no potential to identify six sleep stages in this case. Based on Fig. 5 it could not be used as a key feature to classify sleep stages due to the overlaps in graph similarities. However, when it was used with the degree distribution it was able to differentiate sleep stages. Fig. 5 illustrates the accuracy based on the graph features.

Table 3
Classification accuracy of the proposed method based on the 12 features.

Sleep stage	AWA	S1	S2	S3	S4	REM	Average
Accuracy	94.0%	92.16%	90.0%	92.3%	93.44%	91.2%	92.%
Sensitivity	96.7%	93.0%	90.0%	95.0%	95.0%	91.0%	93.4%

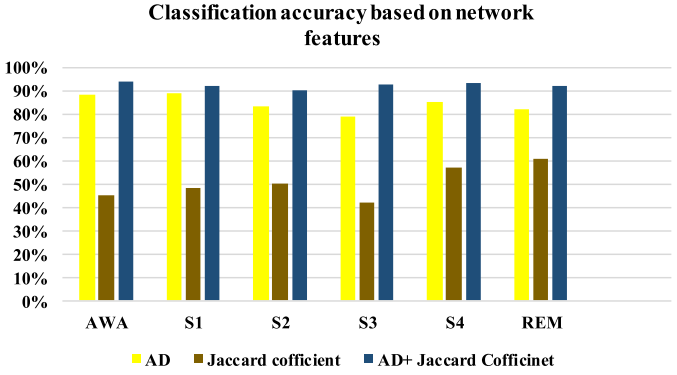


Fig. 6. Classification Accuracy based on network properties.

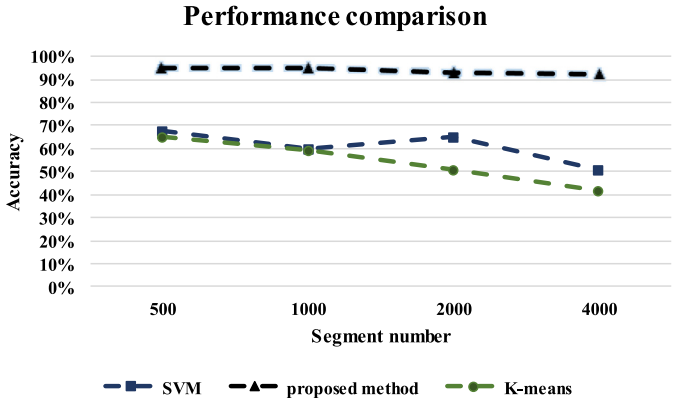


Fig. 7. Performance comparison among the proposed method, k-means and SVM.

Table 2 reports the classification accuracy and sensitivity based on the top nine statistical features. Based on obtained results, we can see that the accuracy rates of the six sleep stages varies. The better performance was achieved for S1, S2 and AWA. The lowest accuracy rate was 69% for S3, and 70% for REM. The experimental results shown in Table 2 confirmed that the nine features set was not sufficient to identify S3, S4 and REM. In summary, the performance of the proposed method was acceptable in terms of accuracy when using the top nine statistical features to classify the sleep stages of AWA, S1 and S2 from EEG signals, while the accuracies were a little bit low for S3, S4 and REM. The average of accuracy for the proposed method after repeating it 10 times using top nine features was 81% with C3-A2 channel EEG. Based on Table 2, the results implied that the proposed method had the ability to classify the stages of AWA, S1, S2 with a good accuracy of 86%, 88% and 87%.

5.1.2. Experiment 2: using twelve statistical features

This experiment was designed to investigate the performance of the proposed method using all twelve statistical features {*max, min, mode, median, range, 1st quartile, 2nd quartile, standard deviation, variation, skewness, kurtosis*}. 900 (12×75) statistical features were extracted from each 30 s segment and mapped into a graph. The main purpose here was to exam the effects of 12 features on the properties of graph and how the increasing in the number of nodes in graphs could improve the discrimination ability of sleep stages classification.

As mentioned in experiment-1, the average degree (AD) and Jaccard coefficient were examined to identify the improvement in classification after using 12 features. We observed that, the AD became the essential property to distinguish the stages of AWA, S1 and S2, significantly, after added the statistical features of {variation, skewness and kurtosis}. Based on Fig. 6 the impact of Jaccard coefficient on the final classification accuracy was improved and also became an important feature to separate the REM sleep stage, probably due to the relationship between network nodes was improved. It was observed that the two properties of the average degree and Jaccard coefficient from the complex networks can significantly distinguish the stage of AWA.

Table 3 shows the classification accuracies achieved by the proposed method using the twelve features set. Based on the results, the classification accuracy for S3, S4 and REM stages was improved by more than 18%, compared to those in Experiment 1. The results confirmed that EEG sleep stages could be represented as graphs. In general, the accuracy was improved for classifying Awake, S1 and S2 by using all the twelve features. However, the accuracy rate for S3, S4 and REM increased significantly when using all the twelve features. It is 92.34%, 93.44% and 93.77%, respectively.

Also as shown in Table 3, the highest classification accuracy rate was 92.16% for Stage Awake, and 93.44% for S4, respectively. The performance was improved for all the sleep stages. The average of accuracy using all the 12 features was 92%. Based on the results in Table 2, the proposed method was efficient to classify the six sleep stages with a good accuracy.

5.2. Comparison study

To investigate the performances of the proposed method with other existing methods, we conducted two types of comparisons in this section. Firstly, the performance of the proposed method was compared with the support vector machine (SVM) and k-means, the two popular classification methods. Secondly, the obtained results in Tables 3 and 4 were compared with other existing methods, proposed by Fraiwan et al. (2010), Güneş et al. (2009), Ozsen, (2013), Zhu et al. (2014) and Hsu, Yang, Wang, and Hsu (2013).

5.2.1. Comparison with the SVM and K-means

In this section, the comparison was made in terms of the classification accuracy and time complexity. The twelve statistical features extracted in Section 4.1.2 were used for the comparison. 4000 randomly selected segments were used in the comparison. 900 statistical features were extracted from each segment and forwarded to the proposed method, k-means and SVM classifiers to classify sleep stages. The SVM is a binary classifier and in order to evaluate the performance of the SVM with the proposed method as well as the k-means, a pair of 2-state sleep stage classification were assessed and the accuracy of each sleep stage was calculated.

Fig. 7 presents the classification accuracies from the three algorithms. The performance of the proposed method was better than these of the SVM and k-means classifiers. The accuracy of the SVM degraded from 68% to 51% when dealing with 4000 segments. In contrast, the proposed method outperformed the SVM. According to the obtained results, the performance of the k-means decreased

Table 4
Performance comparison of different methods with the proposed method.

Author	Sleep stages	Extraction features	Model	Accuracy
Güneş et al., 2009	AWA, S1, S2, S3, S4 & REM	Welch spectral analysis	Decision tree	91.40%
Hsu et al., 2013	AWA, S1, S2 & REM	Energy features	Recurrent neural classifier	87.2%
Fraiwan et al., 2010	AWA, S1, S2, S3, S4 & REM	Time frequency entropy at different bands	Linear discrimination	84%
Zhu et al., 2014	AWA, S1, S2, S3, S4 & REM	Mean degree of DVG and HVG	Different Visibility graph algorithm + SVM	87.5%
Ozsen, (2013)	AWA, S1, S2, S3, S4 & REM	Dependent sequential features	ANN	90.93%
The proposed method	AWA, S1, S2, S3, S4 & REM	Complex network properties	Complex network + k-means	**92.16%**

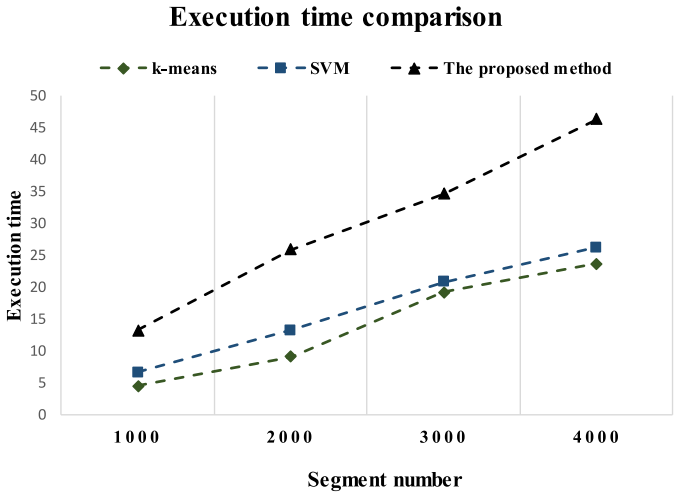


Fig. 8. Comparison of the execution time among the proposed method, k-means and SVM.

significantly when it dealt with 4000 segments. However, with sleep stages classification using complex networks combined with k-means, the discrimination ability was improved significantly. In addition, the effect of the statistical features in sleep classification, are found to be more effective and positive when implemented using complex networks. The average of 10 times accuracy of the proposed method, the k-means and the SVM were 93% 60% and 53% respectively.

In terms of the execution time, from Fig. 8 we observed that the proposed method took an acceptable time although it had more processing steps involved in the algorithm. K-means was recorded the lowest execution time compared with the SVM and the proposed method. Although converting the statistical features to complex networks would take more time it provided more accurate results in sleep stages classification.

5.2.2. Comparison with other single channel sleep stages classification methods

To verify the performance of the proposed method, Table 4 presents the comparison of the performances among the proposed method and some reported methods in the literature. The winning classification accuracy rates among the methods were highlighted in bold font in Table 4. Based on the results in Table 4, the proposed method is the best among the three methods. Güneş et al. (2009) reported their results of the sleep stages classification using a less number of epochs compared with our proposed method. The Welch spectrum was used to extract the important features from EEG data and then fed the features to a decision tree classifier. The proposed method achieved high results compared with one proposed by Güneş et al. (2009). Hsu et al. (2013) identified sleep stages based on energy features. The energy features were extracted from each EEG segment, and a recurrent neural classifier was used to classify the extracted features. The research by Ozsen, (2013) used dependent sequential features extraction and artificial neural network algorithm to distinguish sleep stages. The research reported an average of accuracy 90.93%, about 4% lower than the proposed method in this paper. Fraiwan et al. (2010) focused on designing a system to differentiate sleep stages based on time frequency entropy. The average of the accuracy results obtained was 84%. The accuracy results were also lower than the proposed method. Zhu et al. (2014) developed a method to classify sleep stages. Based on visibility graph approach, the researchers achieved an 87.5% accuracy. The proposed method is outperformed those by Zhu et al. (2014). Based on Table 4, the proposed method can be served as an effective method for EEG sleep stages classification.

6. Conclusions

In this paper, an efficient sleep stages classification method was proposed. The statistical properties of EEG signals were analysed using complex networks to classify a single channel EEG signal into six sleep stages. The statistical feature vector of each EEG segment was mapped into a complex network. The average degree and Jaccard coefficient were calculated from each network to form a feature vector. All the network properties were then forwarded to a k-means classifier for the sleep stages classification. Two sets of the twelve and the first nine features were tested and evaluated. The experimental results show that the twelve features set yielded better performances for all sleep stages, with an average accuracy of 92.16%.

The proposed algorithm was compared with the other existing methods. It was also compared with k-means and SVM classifiers to identify the ability of using complex networks to classify EEG sleep stages. It was demonstrated that the proposed method achieved the best performance in terms of the classification accuracy. The proposed method has the potential to help physicians to accurately diagnose and treat sleep disorders. It can also be applied to different medical data types and be used in different application areas.

References

Adnane, M., Jiang, Z., & Yan, Z. (2012). Sleep–wake stages classification and sleep efficiency estimation using single-lead electrocardiogram. *Expert Systems with Applications, 39*, 1401–1413.

Ahmadlou, M., Adeli, H., & Adeli, A. (2012). Graph theoretical analysis of organization of functional brain networks in ADHD. *Clinical EEG and Neuroscience, 43*, 5–13.

Anuradha, K., & Sairam, N. (2011). Classification of images using JACCARD co-efficient and higher–order co-occurrences'. *JATTI, 34*, 100–104.

Artameeyanant, P., Sultornsanee, S., & Chamnongthai, K. (2015). Classification of electromyogram using weight visibility algorithm with multilayer perceptron neural network, knowledge and smart technology (KST). In *2015 7th international conference on. IEEE* (pp. 190–194).

Bajaj, V., & Pachori, R. B. (2013). Automatic classification of sleep stages based on the time-frequency image of EEG signals. *Computer Methods and Programs in Biomedicine, 112*, 320–328.

Blondel, V. D., Gajardo, A., Heymans, M., Senellart, P., & Van Dooren, P. (2004). A measure of similarity between graph vertices: Applications to synonym extraction and web searching. *SIAM Review, 46*, 647–666.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory* (pp. 144–152). ACM.

Doroshenkov, L., Konyshev, V., & Selishchev, S. (2007). Classification of human sleep stages based on EEG processing using hidden Markov models. *Biomedical Engineering, 41*, 25–28.

Fang, Z., & Wang, J. (2014). Efficient identifications of structural similarities for graphs. *Journal of Combinatorial Optimization, 27*, 209–220.

Fraiwan, L., Lweesy, K., Khasawneh, N., Fraiwan, M., Wenz, H., & Dickhaus, H. (2010). Classification of sleep stages using multi-wavelet time frequency entropy and LDA. *Methods of Information in Medicine, 49*, 230–237.

Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., et al. (2000). Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals. *Circulation, 101*, e215–e220.

Güneş, S., Dursun, M., Polat, K., & Yosunkaya, Ş. (2011). Sleep spindles recognition system based on time and frequency domain features. *Expert Systems with Applications, 38*, 2455–2461.

Güneş, S., Polat, K., Yosunkaya, Ş., & Dursun, M. (2009). A novel data pre-processing method on automatic determining of sleep stages: K-means clustering based feature weighting. In *Proceedings of Complex Systems and Applications ICCSA* (pp. 112–117).

Guo, L., Rivero, D., Seoane, J. A., & Pazos, A. (2009). Classification of EEG signals using relative wavelet energy and artificial neural networks. In *Proceedings of the first ACM/SIGEVO summit on genetic and evolutionary computation* (pp. 177–184). ACM.

Herrera, L. J., Fernandes, C. M., Mora, A. M., Migotina, D., Largo, R., Guillén, A., et al. (2013). Combination of heterogeneous EEG feature extraction methods and stacked sequential learning for sleep stage classification. *International journal of neural systems, 23*, 1350012.

Hsu, Y.-L., Yang, Y.-T., Wang, J.-S., & Hsu, C.-Y. (2013). Automatic sleep stage recurrent neural classifier using energy features of EEG signals. *Neurocomputing, 104*, 105–114.

Iglesias, F., & Kastner, W. (2013). Analysis of similarity measures in times series clustering for the discovery of building energy patterns. *Energies, 6*, 579–597.

Ishizaki, R., Shinba, T., Mugishima, G., Haraguchi, H., & Inoue, M. (2008). Time-series analysis of sleep–wake stage of rat EEG using time-dependent pattern entropy. *Physica A: Statistical Mechanics and its Applications, 387*, 3145–3154.

Jo, H. G., Park, J. Y., Lee, C. K., An, S. K., & Yoo, S. K. (2010). Genetic fuzzy classifier for sleep stage identification. *Computers in Biology and Medicine, 40*, 629–634.

Kemp, B., Zwinderman, A. H., Tuk, B., Kamphuisen, H. A., & Oberyé, J. J. (2000). Analysis of a sleep-dependent neuronal feedback loop: The slow-wave microcontinuity of the EEG. *Biomedical Engineering, IEEE Transactions on, 47*, 1185–1194.

Liao, T. W. (2005). Clustering of time series data—a survey. *Pattern Recognition, 38*, 1857–1874.

Liu, C., Zhou, W.-X., & Yuan, W.-K. (2010). Statistical properties of visibility graph of energy dissipation rates in three-dimensional fully developed turbulence. *Physica A: Statistical Mechanics and its Applications, 389*, 2675–2681.

Iber, C., Ancoli-Israel, S., Chesson, A., & Quan, S. (2007). *The AASM manual for the scoring of sleep and associated events American Academy of Sleep Medicine*. Westchester: American Academy of Sleep Medicine.

Micheloyannis, S., Pachou, E., Stam, C. J., Vourkas, M., Erimaki, S., & Tsirka, V. (2006). Using graph theoretical analysis of multi channel EEG to evaluate the neural efficiency hypothesis. *Neuroscience Letters, 402*, 273–277.

Nascimento, M. C., & Carvalho, A. C. (2011). A graph clustering algorithm based on a clustering coefficient for weighted graphs. *Journal of the Brazilian Computer Society, 17*, 19–29.

Özşen, S. (2013). Classification of sleep stages using class-dependent sequential feature selection and artificial neural network. *Neural Computing and Applications, 23*, 1239–1250.

Panzica, F., Varotto, G., Rotondi, F., Spreafico, R., & Franceschetti, S. (2013). Identification of the epileptogenic zone from stereo-EEG signals: A connectivity-graph theory approach. *Frontiers in Neurology, 4*, 1–6.

Rechtschaffen, A., & Kales, A. (1968). A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects, 55, 305–310.

Şen, B., Peker, M., Çavuşoğlu, A., & Çelebi, F. V. (2014). A comparative study on classification of sleep stage based on EEG signals using feature selection and classification algorithms. *Journal of Medical Systems, 38*, 1–21.

Shi, J., Liu, X., Li, Y., Zhang, Q., Li, Y., & Ying, S. (2015). Multi-channel EEG-based sleep stage classification with joint collaborative representation and multiple kernel learning. *Journal of Neuroscience Methods, 254*, 94–101.

Stam, C. J., & Reijneveld, J. C. (2007). Graph theoretical analysis of complex networks in the brain. *Nonlinear Biomedical Physics, 1*, 1–19.

Siuly, Li, Y., & Wen, P. P. (2011). Clustering technique-based least square support vector machine for EEG signal classification. *Computer Methods and Programs in Biomedicine, 104*, 358–372.

Šušmáková, K., & Krakovská, A. (2008). Discrimination ability of individual measures used in sleep stages classification. *Artificial Intelligence in Medicine, 44*, 261–277.

Uğuz, H. (2012). A hybrid system based on information gain and principal component analysis for the classification of transcranial Doppler signals. *Computer Methods and Programs in Biomedicine, 107*, 598–609.

Van Straaten, E. C., & Stam, C. J. (2013). Structure out of chaos: Functional brain network analysis with EEG, MEG, and functional MRI. *European Neuropsychopharmacology, 23*, 7–18.

Xiao, M., Yan, H., Song, J., Yang, Y., & Yang, X. (2013). Sleep stages classification based on heart rate variability and random forest. *Biomedical Signal Processing and Control, 8*, 624–633.

Xu, X., Zhang, J., & Small, M. (2008). Superfamily phenomena and motifs of networks induced from time series. In *Proceedings of the national academy of sciences: 105* (pp. 19601–19605).

Zhang, J., & Small, M. (2006). Complex network from pseudoperiodic time series: Topology versus dynamics. *Physical Review Letters, 96*, 238701-4.

Zhou, S.-M., Gan, J. Q., & Sepulveda, F. (2008). Classifying mental tasks based on features of higher-order statistics from EEG signals in brain–computer interface. *Information Sciences, 178*, 1629–1640.

Zhu, G., Li, Y., & Wen, P. P. (2014). Analysis and classification of sleep stages based on difference visibility graphs from a single-channel EEG signal. *IEEE Journal of Biomedical and Health Informatics, 18*, 1813–1821.

Zoubek, L., Charbonnier, S., Lesecq, S., Buguet, A., & Chapotot, F. (2007). Feature selection for sleep/wake stages classification using data driven methods. *Biomedical Signal Processing and Control, 2*, 171–179.

2.2 Summary of results

Diykh et al.(2016) demonstrated that integrating the complex networks concept with the statistical model had a high potential in EEG signals analysis. The proposed method classified EEG sleep signals with a high accuracy using a simple approach compared with other state of the art approaches where different transformation techniques were used. Diykh et al. (2016) investigated changes in weighted network's behaviours during the sleep cycle based two network attributes the average degree and Jaccard coefficient. However, using only these networks attributes was not good enough to identify all the EEG sleep stages with a high accuracy. Investigating from other graph attributes, such as degree distribution and clustering coefficients, improved the accuracy of sleep stage identification.

In Figure 2.3, we investigated the importance of each time domain feature. During the training phase, at each iteration, one statistical feature was extracted from each sub-segment of EEG signal. Then, a vector of features was formed to represent each EEG segment and then fed to the proposed model. This process was repeated twelve times and the accuracy at each iteration was recorded. The performances of the proposed model based on each feature was described in Figure 2.3. Then, the features were sorted in descending order. Based on results in Figure 2.3, we picked the most influential features and used them as a combination of nine and twelve features sets. The results showed that twelve features set gives a higher accuracy than other features sets.

New experiments were made to study the correlation between different combinations of features including {Mean and min}, {range, quartiles, median and std.dev}, and {Variance", skew and kurtosis}. The obtained results showed that the proposed model gives higher accuracy than the other two sets. The next chapter will be discussing EEG sleep classification results based on a new combination of network features. The obtained results showed that the proposed model gives slightly different results with these combinations of features. An average accuracy of 70%, 75% and 78% respectively for {Mean and min}, {range, quartiles, median and std.dev}, and {Variance", skew and kurtosis}.

3

CHAPTER 3

EEG SLEEP STAGES CLASSIFCIATION BASED ON TIME DOMAIN FEATURES AND STRUCTURAL GRAPH SIMILARITY

The observations discussed in Chapter 2 show that EEG signals exhibit nonlinear behaviours. One of the effective non-linear methods is complex networks. According to the results in Chapter 2, it was found that complex networks yielded promising results in EEG signals' classification. In this chapter, a new combination of graph attributes is used for further improve the performance in Diykh et al. (2016b). Two new graph features were investigated to improve the sleep stages' classification accuracy including the distribution and clustering coefficient as well as Jaccard coefficients.

Two EEG datasets were used in this study. The data were acquired from two different channels Pz-Oz and C3-A1. It was found that degree distribution and the clustering coefficient could be used to identify the relationships and patterns in EEG sleep stages signals. These relationships were difficult to identify using the network features explored in Chapter 2. The effectiveness of the structural networks features on sleep stages identification was investigated. As a result, individual sleep stages were found to be better classified using a combination of degree distribution, clustering coefficient and Jaccard coefficients.

Although the proposed method achieved a high classification accuracy and sensitivity for AWA, S1, S2, SWS and REM, it showed a relatively low performance in

identifying S3 and S4 due to their similarities. In general, the method has a high noise tolerance without the need to pre-process the original EEG data. However, processing would take a long execution time because of the need to map the all of the features into a complex network for each EEG segment.

3.1 Diykh et al., (2016b) "EEG SLEEP STAGES CLASSIFICATION BASED ON TIME DOMAIN FEATURES AND STRUCTURAL GRAPH SIMILARITY"

The paper published by Diykh et al., (2016b), EEG sleep stages classification based on time domain features and structural graph similarity

EEG Sleep Stages Classification Based on Time Domain Features and Structural Graph Similarity

Mohammed Diykh, Yan Li, and Peng Wen

***Abstract*—The electroencephalogram (EEG) signals are commonly used in diagnosing and treating sleep disorders. Many existing methods for sleep stages classification mainly depend on the analysis of EEG signals in time or frequency domain to obtain a high classification accuracy. In this paper, the statistical features in time domain, the structural graph similarity and the K-means (SGSKM) are combined to identify six sleep stages using single channel EEG signals. Firstly, each EEG segment is partitioned into sub-segments. The size of a sub-segment is determined empirically. Secondly, statistical features are extracted, sorted into different sets of features and forwarded to the SGSKM to classify EEG sleep stages. We have also investigated the relationships between sleep stages and the time domain features of the EEG data used in this paper. The experimental results show that the proposed method yields better classification results than other four existing methods and the support vector machine (SVM) classifier. A 95.93% average classification accuracy is achieved by using the proposed method.**

***Index Terms*—Electroencephalogram (EEG) signal, sleep stages, structural graph similarity, time domain features.**

I. Introduction

ELECTROENCEPHALOGRAM (EEG) signals are measured using electrodes placed on the scalp. They record the electrical potentials generated by the nerve cells in the brain and open a window for the investigation of neural activities and brain functioning [1], [19]. EEG signals are an important source for studying human brain activities and for diagnosing and monitoring neurological diseases, such as sleep disorders and epilepsy.

We spend about one third of our life asleep. Human sleep is a dynamic process which can be divided into two main states: the rapid eye movement (REM) and the non-rapid eye movement (NREM), whereas the latter can be further divided into 4 stages, namely, Stages 1, 2, 3, and 4 [1], [30]. Each sequential stage of the NREM sleep stages is indicative of a deeper sleep, with Stage 1 being the lightest and Stage 4 the deepest [25]. A typical night's sleep consists of 75% NREM sleep. The REM is a status stage where dreams occur and constitutes 25% of a normal sleep night [21], [25].

When used for diagnosis, physicians are in favour of EEGs instead of other biological signals, such as electromyography (EMGs) and electrooculography (EOGs) [7], [9], [19]. As a result, many commonly used medical classification systems are developed based on EEG signals [5], [19], [26], [39], [43]. In recent years, much research work has been conducted in EEG classification. Usually, such classification is completed using single channel EEG signals [15], [43]. Those work focused on classifying sleep stages using different techniques, such as neural networks [19], support vector machines (SVMs) [36], [43]and k-means [11], [26]. But it is often difficult to achieve a high level of accuracy [34].

Graph theory is increasingly used in neuroscience research to study brain diseases and healthy subjects [27], [39]. It is a powerful tool for characterizing the functional topological properties of brain networks involved both in normal and abnormal brain functioning [4], [12], [20], [32], [35], [38]. In addition, graph theory has been proved to be very useful in statistics to describe the relations among random variables [27], [31], [35]. In recent years, it has been widely used for classifying and analyzing the relationships in complex networks, such as biological and brain networks, social networks, signal and image processing. In image processing, graph theory is considered as one of the most powerful tools to classify and analyze digital images. In this work, graph theory is employed to classify sleep stages using EEG signals.

In this paper, the time domain features and structural graph similarity combined with K-means (SGSKM) are used to achieve a high level of classification accuracy, which is much better than the manual results obtained by the experts (about 83%). In this work, each segment from an EEG signal is partitioned to m sub-segments to make the signal stationary. The number of sub-segments (m) is empirically determined by experiments. Then the statistical features are extracted from each sub-segment to reduce the data dimension. A vector of features from each segment is extracted and then mapped into a graph. Finally, the structural graph properties are extracted from each of the graphs to represent the original EEG signal. The K-means algorithm is applied to classify the graph representation features into six sleep stages groups, with each group representing one sleep stage.

In order to evaluate the performance of the proposed method, the comparisons are made among the proposed method and four other existing methods as well as the SVM. The experimental results show that the proposed method gives better accuracy results (95.93%), compared with other methods. This technique has the potential to improve the existing EEG sleep stages classification techniques and help physicians with auto sleep stages scoring.

Manuscript received September 04, 2015; revised January 18, 2016, March 23, 2016; accepted April 04, 2016. Date of publication April 14, 2016; date of current version November 23, 2016.

The authors are with the School of Agricultural, Computational and Environmental Sciences, University of Southern Queensland, Australia (e-mail: mohammed.diykh@usq.edu.au; yan.li@usq.edu.au; peng.wen@usq.edu.au).

Digital Object Identifier 10.1109/TNSRE.2016.2552539

The rest of the paper is structured as follows. Section II presents a brief overview about the current EEG sleep stages classification techniques. In Section III, the EEG datasets used in the paper are described. Section IV introduces the proposed SGSKM method and discusses the performance evaluation measures. Section V presents the experimental results and compares the results from the proposed method and those from other popular methods. Finally, Section VI draws the conclusions from this study.

II. Related Work

In this section, we review some of the related research work. In most research, the representative features are extracted from EEG signals, and then these features are forwarded to a classifier to distinguish the EEG signal into six or seven sleep stages.

In the literature, the wavelet transform has been widely used for sleep EEG stages analysis [6], [11], [24], [25], [42], [44] Oboyya *et al.* [25] classified six sleep stages using a single channel EEG signal. The wavelet transform was utilized to extract the main features from EEG signals. Then a matrix of the main features was forwarded to a fuzzy C-means algorithm to classify the six sleep stages. The average of the classification rate reported was 85%. However, the ages of the subjects used in that work were limited between 35 and 50.

Güneş *et al.* [11] identified sleep stages using the K-means clustering method based on feature weighting. Welch spectral transform was used to extract the features from EEG signals. 196 features were obtained in their study. Some statistical features were used to reduce the number of the obtained features. Finally the K-means and decision tree techniques were used to classify the sleep states into six stages. The accuracy percentage was scored around 83%. The study was conducted with just five male subjects.

Naderi *et al.* [24] also employed Welch spectrum analysis and a neural network to classify sleep stages. The Welch spectrum analysis was used to decompose and extract the features from the EEG signals. The extracted features were forwarded to a recurrent neural network. The data used in that research was from the Sleep-EDF (Europe data format) database.

Zhovna and Shallom [42]used a multi-channel EEG signal modelling method to classify sleep stages. In the first part, a multichannel auto regressive model was used to extract the main parameters from the EEG signals. In the second part, the classification process was achieved using Kullback-Leibler divergence method. The classification accuracy percentage obtained in the study was 90%.

More recently, Zhu *et al.* [43] utilized the concepts of visibility graphs and horizontal visibility graphs to extract the features from EEG signals. The representative graph features were extracted, and then were forwarded to a support vector machine to classify the six sleep stages. The datasets used in the research were from the Sleep-EDF database. It was reported that an average of 87% classification accuracy was achieved.

Suily *et al.* [33] developed and applied a new technique that was called clustering technique based least square support vector machine to classify EEG signals. The classification method was conducted in two phases. In the first phase, the clustering technique was used to extract the features. Then, the least square support vector machine was utilized to classify the extracted features. The proposed method was conducted with three publicly available datasets: epileptic EEG, motor imagery EEG data and mental imagery tasks EEG. The average classification accuracy for the three datasets were 94.55%, 84.52%, and 61.60%, respectively.

Bajaj and Pachori [2] classified the sleep stages based on the time frequency images (TFIs) of EEG signals by utilizing a smooth pseudo Wigner-Ville distribution. The frequency bands of rhythms of EEG signals were used to segment the TFIs. The histograms of the segmented TFIs were forwarded to a multiclass least squares support vector machine for automatically classifying the sleep stages.

Aboalayon *et al.* [1] also classified the sleep stages based on EEG signals. Butterworth band pass filters were used to decompose the EEG signals into five bands (delta, theta, alpha, beta, and gamma). Different features were extracted from the five bands, and these features were fed to a SVM to classify them into six sleep stages. The dataset used in the study was publicly available online from the Sleep-EDF database. The same datasets were used as the ones used in this paper. The study reported a 90% classification accuracy.

Hsu *et al.* [13] introduced a recurrent neural classifier for sleep stages classification. The energy features were extracted from the characteristics of an EEG signal (fpz-cz channel). The recurrent neural classifier distinguished the energy features extracted from each epoch (30 s of an EEG signal) into one of the sleep stages. Eight sleep recording files were used in the work, which were obtained from the sleep-EDF database.

Vural and Yildiz [39] made a study to determine the ability of identifying sleep stages using different feature sets extracted from sleep EEG signals. In this study three types of features were used to represent the original EEG signals. They were time domain, frequency domain (spectral power analysis) and hybrid features which included both the time domain and frequency domain features. The principle component analysis was used to determine the best feature set that could determine the EEG sleep stages correctly.

III. Experimental Data

In this work, two publicly available datasets are used for the proposed method to classify EEG sleep stages. The following section gives brief explanations of the two datasets.

A. Sleep-EDF Database (Dataset-1)

The dataset was obtained from the Sleep-EDF data [17], [10]. The dataset is available for the public to access.[1] In the dataset, there were eight sets of EEG data collected and all were selected for the experiments in this paper. The datasets were recorded from different volunteers in 1989 and 1994, respectively, and stored in the EDF format. The demographics information of subjects were: The first four records (sc4112e0 to sc4002e0) were recorded in 1989 from Caucasian males and females healthy volunteers during 24 h in their normal daily life, aged from 26–101. The other four subjects named (st7132j0 to st7022j0) were recorded in 1994 also from Caucasian males and

[1]http://www.physionet.org/physiobank/database/sleep-edf/

TABLE I
INFORMATION OF THE EXPERIMENTAL DATA (DATASET-1)

Sleep stage	No. of Segments
AWA	7830
S1	603
S2	3621
S3	672
S4	627
REM	1609
Total no. of segments	14963

TABLE II
INFORMATION OF THE EXPERIMENTAL DATA (DATASET-2)

Sleep Stage	No. of segments
AWA	8858
S1	2856
S2	21184
S3	3216
S4	4441
REM	8000
Total no. of segments	48555

female who had mild difficulty falling asleep but were otherwise healthy. Their age was from 18–34 years old [18], [23]. Each file contains one horizontal channel, two EEG signal channels (Fpz-Cz and Pz-Oz), Resporonasal, EMGSubmenta, Tempbody, and Eventmarker.

In this study we used channel Pz-Oz because it provided a better classification performance than Fpz-Cz channel [29], [43]. The original sleep stages of these segments were labelled as AWA, S1, S2, S3, S4, REM MVT (movement time), and UNS (unknown state). Each whole EEG signal was divided into segments in 30 s (containing 3000 points of data). The recordings were also scored in sleep stages according to the R& K criteria [28] with every 30 s of an EEG signal. Table I shows the number of segments that are used in this study.

B. Sleep Spindles Database (Dataset-2)

The EEG sleep stages database which is publically available online[2] is also used in this study. They were recorded at the University of MONS-TCTS laboratory and the University of Libre de Bruxelles—CHU de Charleroi Sleep Laboratory [37]. The recordings were obtained from eight males and females (aged from 31–54). They consist of 20 whole night recordings. Eight recordings were used in this study. The recordings contain two EOG channels (P8-A1, P18-A1), three EEG channels (CZ-A1 or C3-A1, FP1-A1 and O1-A1), and one submental EMG channel.

In this study only C3-A1 channel was used. The data were sampled at frequency of 200 Hz. The EDF standard was utilized for storing. The recordings were also scored in sleep stages according to the R&K criteria [28]. Table II shows the number of segments that are used in this study.

[2]http://www.tcts.fpms.ac.be/~devuyst/Databases/DatabaseSubjects

IV. METHODOLOGY

This paper proposes an efficient method for classifying EEG signals into six sleep stages. Fig. 1 illustrates the structure of the proposed method. Each segment of the EEG signals is divided into m sub-segments. The statistical features from each sub-segment are extracted. The vector of the features from each segment is then mapped into a graph. For each graph the structural similarity properties are extracted. The graph features are then forwarded to a classifier to classify each segment into one of the six sleep stages. The details of the methodology are explained in the following sections. We also apply the SVM to the two datasets for the comparison with the proposed method.

A. Statistical Features Extraction

Feature extraction is the most important part of the classification because if the features are not chosen well the performance of a classification will be degraded. It is even more important to design an effective method to extract features from EEG signals, because they are also periodic. One effective method to quantifying a non-stationary time series such as EEG signals is to view it as consisting of many number of segments that are themselves stationary. In this work, the EEG signals are divided into smaller segments. The interval of each segment is 30 s (3000 data points) in this paper according to the Rechtschaffen and Kales [28]. Then, each segment is further divided into m sub-segments for a shorter period of time to make the signals stationary and $m \geq 1$. The number of sub-segments from each segment and the length of each sub-segment are empirically determined during the experiments. The optimum number of sub-segments is determined according to the following algorithm.

Algorithm

Input $X = EEG\ signal$

1. *let p be a number that each EEG segment should be divided, initially* $p = 0$
2. $k = first\ segment\ of\ X$
3. ***For each*** *EEG segment* ***do***
4. $p = p + 1$
5. ***divide*** k *into p sub-segments*
6. ***extracting*** *the statistical features from each sub-segment*
7. ***forward*** *the extracted features to the proposed method*
8. ***if*** *the classification accuracy is satisfactory then stop segmentation, got to step 10*
9. *Else go to step 4*
10. *End*

Output Set of statistical features

Number of sub-segments

As the result, each segment is separated into 75 sub-segments, with each sub-segment includes 40 data points each.

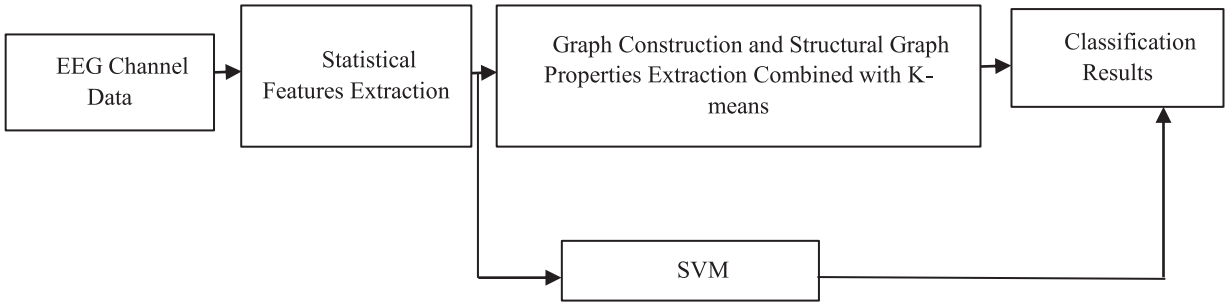


Fig. 1. Diagram of the proposed method.

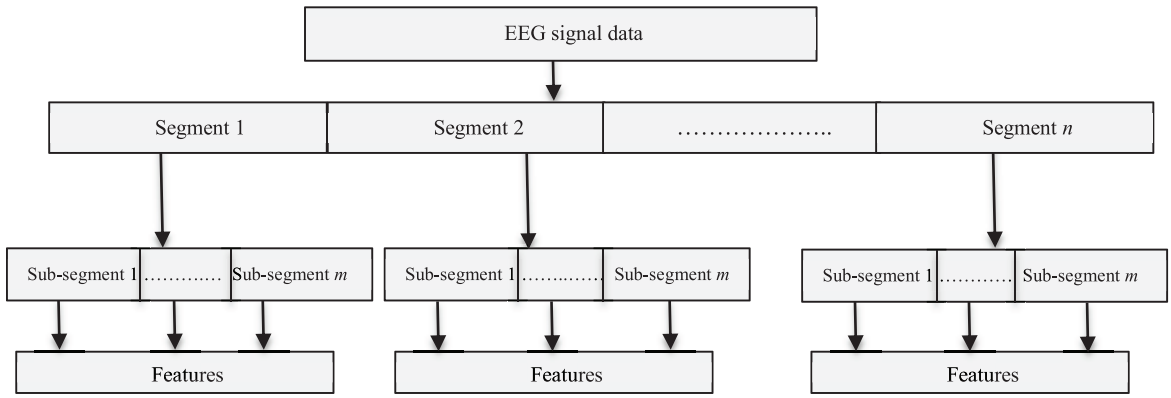


Fig. 2. Segmentation and statistical features extraction.

Fig. 2 shows the segmentation and features extraction technique. In the experiments, we use a statistical approach to extract the time domain features from each sub-segment, and then put all the features from one segment in a vector to represent the segment. It is found that some of EEG data are symmetric distribution and other skewed distribution. The mean and the standard division are considered appropriate measures for a time series with symmetric distribution. While for a skewed distribution median, range and quartile are effective to measure the centre and the spread of a dataset. However, feature mode which defines as the most frequent value is used to measure the locations of a time series. Other statistical features, such as minimum, variation, skewness and kurtosis are used as measures to pull out the important information about a time series. For these reasons, 12 features of *{median, maximum, minimum, mean, mode, range, first quartile, second quartile, standard deviation, variation, skewness, kurtosis}* are considered as key features to represent EEG data in this study. Suily *et al.* [33] used statistical properties as the key features to classify motor images and epileptic data. Sen *et al.* [34] utilized statistical features with time frequency features in a comparative study in EEG classification. Table III provides a short explanation of the features. The 12 features are denoted as $\{X_{Me}, X_{Max}, X_{Min}, X_{Mean}, X_{Mod}, X_{Rand}, X_{Q1}, X_{Q2}, X_{SD}, X_{Var}, X_{Ske,} X_{Kue}\}$. These segment features are then mapped into a graph. The structural graph similarity of all the graphs is used to classify EEG segments into different sleep stages. In Section V, several experiments are designed to test which combination of the features is the best to represent the EEG data.

B. Complex Network and Structural Graph Similarity

Graph theory provides many global and local quantitative measures to analyze the brain network dynamics. In this work, we calculated a topological graph structure property to classify six sleep stages in an EEG signal. Statistical analysis was conducted to extract the main features from graph nodes. According to Zhang and Small [41], each vector of features (one segment) was mapped as an undirected graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ denotes the set of nodes and $\mathbf{E}$ the set of connection among the nodes. The connection between each pair of nodes refers to the existence of a relationships between the nodes [3], [12], [22]. One of the commonly used similarity measuring method for a time series is the Euclidean distance [4], [14], [16]. Let $\{x_{ij}\} = 1, 2, 3 \ldots N$ be a set of time series of N data points. Each data point of the series was assigned to be a node in an undirected graph. Two nodes, v_1 and v_2, in a graph are connected if a distance between the two nodes are less than or equal to an adaptive threshold

$$(v_1, v_2) \in E, if\ d(v_1, v_2) \leq \delta \tag{1}$$

where δ is an adaptive threshold, Fig. 3 shows a time series: $\{2.4, 5.0, 6.2, 6.0, 7.5, 8.5, 3.0, 1.8, 9.5\}$ being transferred into a graph. The adjacency matrix $\mathbf{A}$ of graph $\mathbf{G}$ is calculated for all $\mathbf{V}$ to describe the graph nodes connected. The adjacent matrix of an undirected graph is symmetric, i.e., $\mathrm{A}(v_\mathrm{i}, v_\mathrm{j}) = \mathrm{A}(v_\mathrm{j}, v_\mathrm{i})$

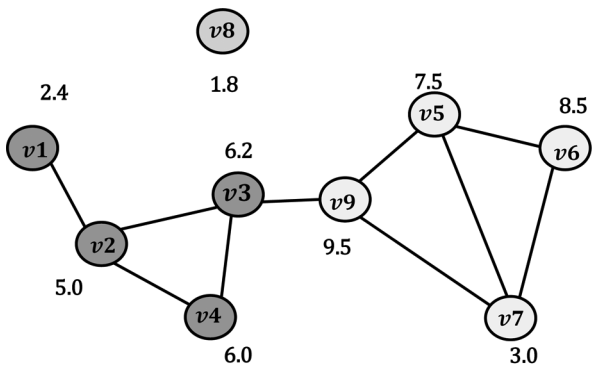
$$\boldsymbol{A}(v_\mathrm{i}, v_\mathrm{j}) = \begin{cases} 1, & \text{if } (v_\mathrm{i}, v_\mathrm{j}) \in \boldsymbol{E} \\ 0, & \text{otherwise} \end{cases}. \tag{2}$$

The number of nodes increases when there are more sub-segments in one EEG segment, and vice versa. All graphs have a

TABLE III
SHORT EXPLANATION OF THE STATISTICAL FEATURES

No.	Feature name	Formula	No.	Feature name	Formula
1	Maximum	X_{Max}=Max[x_n]	7	Minimum	X_{Min}=min[x_n]
2	Mean	$X_{Mean}=\frac{1}{n}\sum_{1}^{n}x_i$	8	Mode	$X_{Mod}=L+\left(\frac{f_1-f_0}{2f_1-f_2}\right)Xh$
3	Median	$X_{Me}=(\frac{N+1}{2})^{th}$	9	Range	X_{Rang}=X_{Max}-X_{Min}
4	First Quartile	$X_{Q1}=\frac{1}{4(N+1)}$	10	Standard Deviation	$X_{SD}=\sqrt{\sum_{n=1}^{N}(x_n-AM)}\,\frac{2}{n-1}$
5	variation	$X_{Var}=\sum_{n=1}^{N}(x_n-AM)\frac{2}{N-1}$	11	Skewness	$X_{Ske}=\sum_{n=1}^{N}(x_n-AM)\frac{3}{(N-1)SD^3}$
6	Kurtosis	$X_{Ku}=\sum_{n=1}^{N}(x_n-AM)\frac{4}{(N-1)SD^4}$	12	Second Quartile	$X_{Q2}=\frac{4}{4(N+1)}$

where X_n=1, 2, 3, …, *n,* is a time series, *N* is the number of data points, AM is the mean of the sample.



A.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
v_1	0	1	0	0	0	0	0	0	0
v_2	1	0	1	1	0	0	0	0	0
v_3	0	1	0	1	0	0	0	0	1
v_4	0	1	1	0	0	0	0	0	0
v_5	0	0	0	0	0	1	1	0	1
v_6	0	0	0	0	1	0	1	0	0
v_7	0	0	0	0	1	1	0	0	1
v_8	0	0	0	0	0	0	0	0	0
v_9	0	0	1	0	1	0	1	0	0

B.

Fig. 3. Example of a time series converted to graph A. Graph G B. Adjacency matrix of graph G.

fixed number of nodes in this paper. From Fig. 3 we can notice that, node v_8 has no connections with other nodes in the graph. This means that node v_8 is an isolated point in the graph.

C. Graph Features

The statistical features of a graph can be obtained from the adjacency matrix of the graph [20], [8], [40].

The following features are extracted and used for classification in this study.

1) Degree Distribution: The degree distribution for a graph, denoted by $\mathbf{P}(k)$, is defined to be the fraction of nodes in the graph with a degree k. The degree distribution can be calculated as

$$\mathbf{P}(k)=\frac{|\{v \mid d(v)=k\}|}{N} \tag{3}$$

where $d(v)$ is the degree of node v, and N is the number of nodes in the graph.

2) Clustering Coefficient: Clustering coefficient (CC) is frequently used to characterize the global and local structures of a graph [20], [22], [35]. Stam *et al.*[35] and Li *et al.* [20] used the CC to analyze brain activities. The clustering coefficient for node v_i in graph $\mathbf{G}$ is defined as $C_{v_i}=L/N_G/2$, where L is the number of the actual links between v_i with its neighbors, and N_G is the number of the neighbors of v_i.The clustering coefficient of the graph is the average of the clustering coefficients of all the nodes

$$C=\frac{1}{N}\sum_{i=1}^{N}C_{v_i} \tag{4}$$

where N is the number of nodes in the graph and C_{v_i} is the CC for node v_i. To calculate the CC of node v_3 in Fig. 3, the three neighbors of v_3 are connected each other $(v_4 \rightarrow v_9, v_3 \rightarrow v_9, v_4 \rightarrow v_3)$. Thus, $\mathrm{CC}=(v_3)=1$.

3) Jaccard Similarity Coefficients: Jaccard coefficient is a statistic function used for comparing the similarity and the diversity of two nodes in a graph [12]. It is evaluated as the set

of the intersection neighbors between two nodes divided by the neighbor set of the union of the two nodes. Jaccard coefficient function is calculated based on the following equation:

$$\boldsymbol{w}(v_1, v_2) = \frac{|\Gamma(v_{\rm i}) \cap \Gamma(v_{\rm j})|}{|\Gamma(v_{\rm i}) \cup \Gamma(v_{\rm j})|} \tag{5}$$

where $\Gamma(v_i)$ is the set of neighbors of node v_j, $\Gamma(v_j)$ is the set of neighbors of v_j, and $\boldsymbol{w} = [0, 1]$. For each graph, a Jaccard coefficient vector is computed.

D. Classification

After the structural graph properties are obtained from each graph, we classify the segment based on the structural graph features. The K-means clustering algorithm [11], [14], [26] is applied to classify the similarity vectors from each graph (segment) into one of the six sleep stages.

E. Performance Assessment

In this study, the cross validation, sensitivity, kappa coefficient and confusion matrix are used to evaluate the performances of the proposed algorithm.

- k-cross-validation: in pattern recognition, k-cross- validation is a very popular measure to evaluate the performance of a classification method. It is used to estimate the quality of a classification method by dividing the number of the correctly classified results by the total of the cases. A dataset is divided into k-alternately exclusive subsets of an equal size. One subset is used as the testing set, while others are considered as the training sets. All the subsets are tested and the accuracy of the classification is calculated. In this work, the 10-cross-validation is used. The average of the overall results for the subset testing is computed

$$\text{Perfomance} = \frac{1}{10}\sum_{1}^{10} \text{accuracy}^{(k)} \tag{6}$$

  where $\text{accuracy}^{(k)}$ is the accuracy for the kth iteration ($k = 1, 2, \ldots 10$).
- Sensitivity: is a statistical measure which is used to evaluate the performance of a classification test by measuring the proportion of the actual positive classification. It is defined as

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{7}$$

  where TP $=$ the number of the positive classification for all the subset testings decisions, FN $=$ the number of the incorrectly classified decisions for the subsets.
- Confusion matrix: is to provide the information about the actual and predication classification results done by the algorithm.
- Classification accuracy: is the number of the correctly classified decisions divided by the total number of the cases.
- Kappa coefficient: is the measurement of performance agreement of two classification products (the proposed method and the experts in this paper) [33], [43].

V. Experimental Results

To evaluate the proposed method, a set of experiments were designed and conducted using the datasets described in Section III. As mentioned before, different sets of different number of statistical features were used and then were transferred into a graph. A number of experiments were also conducted to determine the best number of the features for each sleep stage based on structural graph similarity properties. The results are discussed in the next section. According to the experimental results, the proposed method achieves a 95.4% accuracy and outperforms the other four existing methods. The experiments were conducted using MATLAB (Version: R2013) on a computer with 3.40 GHz Intel core i7 CPU processor machine, and 8.00 GB RAM.

A. Classification Accuracy With Different Number of Features

From the experiment results, we notice that there is a positive relation between the number of the extracted features and the classification accuracy. When the graph nodes increase due to the increase of the number of the extracted statistical features, the discriminative characteristics among the graphs improve, showing more significant differences among sleep stages.

In multi sleep stages classification such as the six sleep stages, it is difficult to obtain high accuracies with the same set of features [42], [43], [13], [19]. In this study, the SGSKM method was also conducted with different sets of statistical features to extract the key characteristic data from the graphs to yield high classification accuracies for six sleep stages.

The SGSKM method was conducted with two datasets to investigate the performances with different channels and a different data size. To determine which set of features can best represent the original EEG signal the best, several experiments were conducted with different sets of features (different number of graph nodes). The 12 statistical features of $\{X_{Me}, X_{Max}, X_{Min}, X_{Mean}, X_{Mod}, X_{Rand}, X_{Q1}, X_{Q2}, X_{SD}, X_{Var}, X_{Ske}, X_{Kue}\}$ were extracted. They were then tested and conducted, separately, to evaluate its classification accuracy from the proposed algorithm. After the proposed method was repeated for 12 times, the importance of each feature was decided based on the classification accuracy results. All the features were then sorted in descending order based on their classification accuracy importance. The final obtained vector based on the results is $\{X_{Mean}, X_{Min}, X_{Mod}, X_{Max}, X_{Me}, X_{Rand}, X_{Q1}, X_{Q2}, X_{SD}, X_{Var}, X_{Ske}, X_{Ku}\}$. Fig. 4 shows that $\{mean\}$ feature is the most important one among the 12 features, with an average accuracy of 16.5% for dataset-1, and 15.7% for dataset-2.

Six-Feature set: The first six features of $\{X_{Mean}, X_{Min}, X_{Mod}, X_{Max}, X_{Me}, X_{Rang}\}$ were used to evaluate the performance of the proposed method. Each segment was represented by a vector of the features containing 450 data points (75×6). The structural graph properties stated in Section IV-B were calculated from each graph. It can be noticed that, the performance of the proposed method significantly improved for several specific sleep stages after having increased the graph nodes with more features. The local graph connectivity significantly increased for stages of: AWA, S1, S2, based on the degree distribution and the average of clustering coefficients. However, the six features were not enough to distinguish the stages: S3, S4,

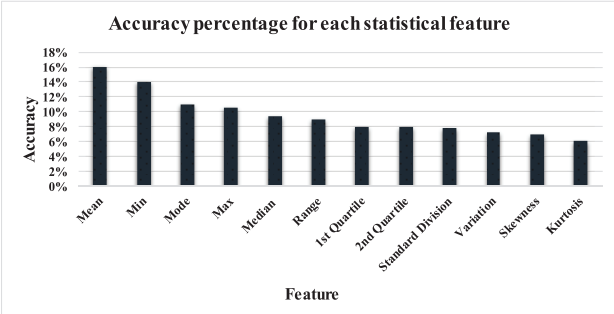


Fig. 4. Order of the statistical features based on the classification accuracy.

TABLE IV
CLASSIFICATION ACCURACY BASED ON SIX FEATURES

Sleep Stages	Dataset-1	Dataset-2
AWA	60.5%	56.5%
S1	59%	55.7%
S2	58.8%	57.5%
S3	53%	52.4%
S4	58%	55.7%
REM	53.7%	50.8%
Average	57.2%	54.8%

and REM, with a high accuracy, due to the overlapping in similarities of the graph nodes for these stages. Table IV provides a summary of the obtained results for the two datasets based on the six features set.

The average of 10-cross validation were 7% and 53% with the six features set for dataset-1 and dataset-2, respectively.

In order to further increase the accuracy, we increased the number of the features to ten in the next experiment.

Ten-Feature Set: Top 10 features $\{X_{Mean}, X_{Min}, X_{Mod}, X_{Max}, X_{Me}, X_{Rang}, X_{Q1}, X_{Q2}, X_{SD}, X_{\mathrm{Var}}\}$ were selected to represent the EEG data. From each segment, 10×75 statistical features from all sub-segments, were extracted and then were mapped into a graph. A very interesting observation in this experiment is that the accuracies by the proposed method for all the six sleep stages significantly improved when the features were increased to ten. The classification accuracies for stage: AWA, S1 and S2 exceeded 85%, increased by more than 25% from the previous experiment with the six feature set for the both datasets. Also, the performances of the proposed method were elevated to 69% for the stages of S3, S4, and REM. Based on the results, the differences in the accuracies between the stages group of AWA, S1, and S2, and the group of the stages of S3, S4 and REM, indicated that it was difficult to use the same set of features to classify all six sleep stages.

Table V presents the classification accuracies for the two datasets. It can be seen that the proposed method yields a high performance by using the ten features. There are no significant differences in the results obtained for both datasets. Also, the kappa coefficients for stages: AWA, S1, S2, are very close for the two datasets.

The averages of classification accuracies of the proposed method for 10 times for dataset-1 and dataset-2 are 79% and 75%, respectively. From the experimental results for the both datasets, it is clear that the 10 features set is sufficient to classify the stage of: AWA, S1, and S2 from the six sleep stages. The results in Table V demonstrates that the ten features set yields an improved performance with the stage of: AWA, S1, and S2, for the different channel datasets, compared with stages: S3, S4, and REM.

TABLE V
CLASSIFICATION ACCURACY BASED ON TEN FEATURES

	Dataset-1		Dataset-2	
Sleep Stages	Accuracy Rate	Kappa coefficient	Accuracy Rate	Kappa coefficient
AWA	86%	0.36	85%	0.31
S1	88%	0.46	89.6%	0.46
S2	87%	0.25	86.8%	0.29
S3	69%	0.20	68%	0.19
S4	69%	0.21	71.2%	0.20
REM	70%	0.23	73.8%	0.25
Average	78.1%	0.29	79.1%	0.28

TABLE VI
CLASSIFICATION ACCURACY BASED ON 12 FEATURES

	Dataset-1		Dataset-2	
Sleep stages	Accuracy rate	Kappa coefficient	Accuracy rate	Kappa coefficient
AWA	97%	0.96	95%	0.92
S1	96 %	0.95	94.6%	0.77
S2	97%	0.92	96.5%	0.95
S3	94.34%	0.67	94%	0.63
S4	95.44%	0.48	95%	0.47
REM	95.77%	0.96	95.8%	0.97
Average	95.93%	0.82	95.15%	0.79

Twelve Features: In this case, all the 12 features of $\{X_{Mean}, X_{Min}, X_{Mod}, X_{Max}, X_{Me}, X_{Rang}, X_{Q1}, X_{Q2}, X_{SD}, X_{Var}, X_{Ske}, X_{Ku}\}$ were used to classify the six sleep stages. From each segment 900 features were extracted and were transferred to a graph. Table VI shows the classification results and kappa coefficients for the six sleep stages with the two datasets.

The most noticeable results in this experiment were that the accuracies for all the six sleep stages were exceeded 90% for the both datasets. According to the obtained results in Table VI, the accuracies for the stages of: S3, S4, and REM, improved by more than 19% compared to the previous results in Table V. More importantly, the results proved that the features {skewness, kurtosis} have significantly increased the accuracy of the classification results for the stages (S3, S4 and REM). Based on the results, the stages: AWA, S2, S1, and REM, recorded the highest accuracy compared with the other stages in the two datasets. Also, the kappa coefficient of the six sleep stages increased nearly more than half in the two datasets compared with the previous results in Table IV. The averages of 10 times classification accuracies for dataset-1 and dataset-2 were 95%

TABLE VII
CONFUSION MATRIX AND SENSITIVITY OF SIX SLEEP STAGES (DATASET-1)

		Expert's Scoring					
		AWA	S1	S2	S3	S4	REM
The proposed method	AWA	780	10	120	4	9	24
	S1	10	484	36	15	47	8
	S2	20	18	3612	4	0	50
	S3	15	78	89	578	3	75
	S4	9	0	90	52	547	14
	REM	12	14	95	19	21	1458
Sensitivity		99%	80%	88%	86%	87%	89%

TABLE VIII
CONFUSION MATRIX AND SENSITIVITY OF SIX SLEEP STAGES (DATASET-2)

		Expert's Scoring					
		AWA	S1	S2	S3	S4	REM
The proposed method	AWA	8750	120	137	2	45	12
	S1	18	2765	78	63	17	7
	S2	78	10	19030	3	611	5
	S3	2	9	129	3216	247	95
	S4	3	17	850	245	3504	921
	REM	7	85	960	114	17	8000
Sensitivity		98.7%	83%	89.8%	86%	79%	87%

and 94%, respectively, Tables VII and VIII illustrate the confusion matrix and sensitivity for the six sleep stages classification. From the obtained results there are no significant differences in the classification accuracy between the two datasets although the size and the channel of the two datasets are different. The sensitivity of AWA and REM in Tables VII and VIII is 99.1%, 98.7%, and 89.3%, 87.1%, respectively, which reveals that the proposed method for Pz-Oz and C3-A1 channels is effective to classify Awake AWA and REM stages from the six sleep stages using the twelve features. Fig. 5 shows the classification accuracy against the number of the features used in the classification. It can be noticed that the twelve features set yields the highest accuracy for the six sleep stages with both datasets.

B. Comparative Study

To investigate the performances of the proposed method, we made two types of performance comparisons in this section. Firstly, the performance of the proposed method was compared with the support vector machine. Secondly, the obtained results in Table V were compared with other four existing methods from the literature. For a fair comparison, we evaluated the proposed method with other studies that used the same datasets and the same EEG channels. The obtained results for dataset-1 were listed in Table VI and were used in the comparison study. There were no reported experimental results for dataset-2.

1) Comparison With the SVM: In this section, the comparison was made in terms of the classification accuracy. The 12 statistical features extracted in Section IV-A were used for the comparison. For the classification accuracy, Fig. 6 presents the comparison between the SVM classifier and the proposed method. The same number of segments were used in the both methods. The segments were selected randomly from dataset-1 and dataset-2.

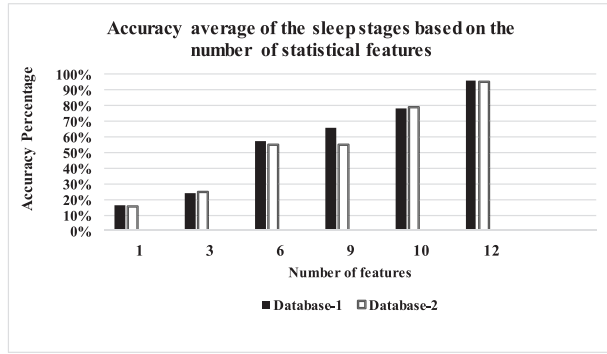


Fig. 5. Classification accuracy based on the number of statistical features.

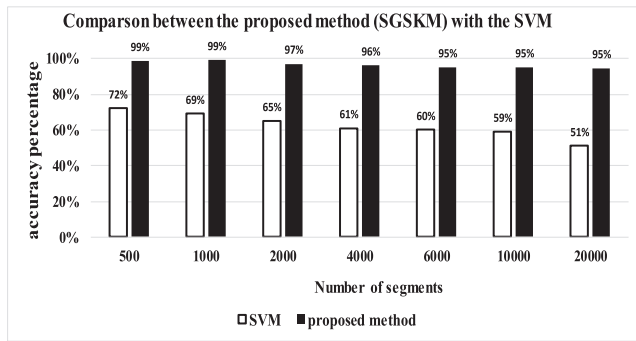


Fig. 6. Accuracy comparison between the proposed method and the SVM.

The selected segments were divided into two sets, the training and testing sets. Then, the extracted features (12) from each segment were forwarded to the SGSKM and the SVM. From Fig. 6, it can be noticed that the performance of the proposed method is better than that of the SVM. The accuracy of the SVM was degraded from 72% to 51% when dealing with 20000 segments.

In contrast, the accuracy of the proposed method slightly changed and yielded more accurate and stable results, compared with the SVM. The average of 10 times accuracies was calculated for each of the two methods.

2) The Comparison With Other Existing Sleep Stage Classification Methods: To evaluate the performance of the proposed method, the classification results were compared with those of other four existing methods. Table IX presents the performances comparison among the proposed method and four other reported methods (Zhu *et al.* [43]; Bajaj *et al.* [2]; Aboalayon *et al.* [1] and Güneş *et al.* [11]. All those studies were used the same dataset (dataset-1) as discussed in Section III. The winning classification accuracy rates among the four methods were highlighted in bold font in Table VIII. Based on the results in Table VIII, the proposed method is the best among the five methods. Bajaj *et al.* [2] reported their results of the sleep stages classification with the same dataset, but using lesser segments compared with the proposed method. The average of the accuracy results they achieved was 92.3% for the stages of {AWA, S1, S2, S3, S4, and REM}. The research by Bajaj *et al.* [2] used a smoothed pseudo Winger-Villa distribution to obtain a segmented time-frequency image (TFI). The TFI divided the EEG signal into five sub-images which corresponded to frequency bands. Then, different features from the histogram of

TABLE IX
ACCURACY COMPARISON WITH OTHER METHODS

Authors	Sleep stages, No. of segments	Method	Accuracy
Bajaj et al. [2]	AWA, S1, S2, S3, S4, REM (4700)	Smoothed pseudo Wigner Ville distribution based on time frequency representation and segmentation.	92.3%
Zhu et al. [43]	AWA S1, S2, S3, S4 REM (14963)	Difference visibility graph.	87.5%
Aboalayon et al. [1]	AWA, S1 (200)	Designed band pass filter to decomposed EEG signal.	92%
Güneş et al. [11]	AWA, S1, S2, S3, REM (4195)	K-means clustering based features weighting combing with k-nearest neighbour.	82.15%
The Proposed method	AWA, S1, S2, S3,S4, REM (14963)	Statistical features and structural graph similarity.	**95.93%**

the TFI were extracted. The average accuracy rates were lower than that obtained in this study. Zhu *et al.* [42] focused on designing a system to differentiate sleep stages using difference visibility graphs. The average of the accuracy results obtained was 87.5% with 14963 segments. The accuracy rates were also lower than that of the proposed method although the number of segments was the same as used in this paper. Aboalayon *et al.* [1] proposed an efficient method to distinguish between AWA and stage1 by designing infinite impulse response (IIR) Butterworth band filters. Firstly, the Butterworth band filters was utilized to filter an EEG signal. Then, the filtered EEG signal was decomposed into five frequency bands (delta, theta, alpha, beta and gamma). Different statistical characteristics were pulled out from each band and then were fed to a SVM. They achieved an average of 92% accuracy for the stages of, AWA and S1. They used only 200 segments from eight subjects in the dataset. Based on our results, the proposed method yields the highest accuracy with a huge number of segments compared with Aboalayon *et al.* [1]. Güneş *et al.* [11] proposed an automatic sleep stages scoring method using K-means clustering based on weighted features. The Welch method was used to extract 129 frequency domain features from an EEG signal. Then, the features were weighted and reduced to four. The reduced features were forwarded to the K-means clustering algorithm. They used a lesser number of segments compared with that of the proposed method. In addition, they reported a classification accuracy rate less than the accuracy obtained by the human experts, which is $83 \pm 3\%$. We can see from the above results, the proposed method achieved the highest accuracy compared with the four other existing methods using the same different EEG channels, while a huge amount of data were used in this paper.

VI. CONCLUSION

In this paper, the statistical features and structural graph properties are used to classify the sleep EEG signals. We have investigated the classification ability of graphs constructed from different statistical features sets in sleep EEG classification. It is found that the 12 features set yields a better performance for all sleep stages (95.93%), with a 5% improvement compared with the recent studies even the proposed method in this paper is applied to a huge amount of data. It is also observed that using a ten features set gives reasonable classification accuracies for stages AWA, S1, S2. This study suggests that the graph theory combined with time domain features can be used to classify EEG signals efficiently without pre-processing. The proposed method is compared with the four other existing methods and the SVM. It is demonstrated that the proposed method can achieve the best performance in terms of the classification accuracy. This method can help physicians to accurately diagnose and treat sleep disorders. It can also be applied to different medical data types and be used to different application areas. With large amounts of EEG recordings available, we will apply big data technologies to classify EEG sleep stages.

REFERENCES

[1] K. Aboalayon, H. T. Ocbagabir, and M. Faezipour, "Efficient sleep stage classification based on EEG signals," in *IEEE Long Island Syst., Applicant. andTechnol. Conf. (LISAT)*, 2014, pp. 1–6.
[2] V. Bajaj and R. B. Pachori, "Automatic classification of sleep stages based on the time-frequency image of EEG signals," *Comput. Methods Programs Biomed.*, vol. 112, pp. 320–328, 2013.
[3] V. D. Blondel, A. Gajardo, M. Heymans, P. Senellart, and P. Van Dooren, "A measure of similarity between graph vertices: Applications to synonym extraction and web searching," *SIAM Rev.*, vol. 46, pp. 647–666, 2004.
[4] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," *Phys. Rep.*, vol. 424, pp. 175–308, 2006.
[5] M. R. Daliri, "Kernel earth mover's distance for EEG classification," *Clin. EEG Neurosci.*, vol. 44, pp. 182–187, 2013.
[6] L. Doroshenkov, V. Konyshev, and S. Selishchev, "Classification of human sleep stages based on EEG processing using hidden Markov models," *Biomed. Eng.*, vol. 41, pp. 25–28, 2007.
[7] F. Ebrahimi, M. Mikaeili, E. Estrada, and H. Nazeran, "Automatic sleep stage classification based on EEG signals by using neural networks and wavelet packet coefficients," in *Proc. 30th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2008, pp. 1151–1154.
[8] Z. Fang and J. Wang, "Efficient identifications of structural similarities for graphs," *J. Combinatorial Optimizat.*, vol. 27, pp. 209–220, 2014.
[9] L. Fraiwan *et al.*, "Classification of sleep stages using multi-wavelet time frequency entropy and LDA," *Methods Inf. Med.*, vol. 49, p. 230, 2010.
[10] A. L. Goldberger *et al.*, "Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, pp. E215–E220, 2000.
[11] S. Güneş, K. Polat, and Ş. Yosunkaya, "Efficient sleep stage recognition system based on EEG signal using k-means clustering based feature weighting," *Expert Syst. Appl.*, vol. 37, pp. 7922–7928, 2010.
[12] Y. He and A. Evans, "Graph theoretical modeling of brain connectivity," *Curr. Opin. Neurol.*, vol. 23, pp. 341–350, 2010.
[13] Y.-L. Hsu, Y.-T. Yang, J.-S. Wang, and C.-Y. Hsu, "Automatic sleep stage recurrent neural classifier using energy features of EEG signals," *Neurocomputing*, vol. 104, pp. 105–114, 2013.
[14] X. Huang and W. Lai, "Clustering graphs for visualization via node similarities," *J. Vis. Languages Comput.*, vol. 17, pp. 225–253, 2006.
[15] S. A. Imtiaz and E. Rodriguez-Villegas, "A low computational cost algorithm for rem sleep detection using single channel EEG," *Ann. Biomed. Eng.*, vol. 42, pp. 2344–2359, 2014.

3.2 Summary of results

Diykh et al. (2016b) identified the EEG sleep stages using structural graph features. The efficiency of the degree distribution, clustering coefficient and Jaccard coefficients in the sleep stages classification were investigated. One of the most important findings in this paper is that the behaviours of networks can vary from one sleep stage to thenext. The effectiveness of the proposed method was tested with two datasets acquired from different EEG sources.

Both Diykh et al. (2016a) discussed in Chapter 2 and Diykh et al. (2016b) showed that the proposed method yields a better performance for all sleep stages compared with the recent studies. Diykh et al. (2016b) suggests that the structural graph features combined with time domain features can be used to classify EEG signals efficiently and without pre-processing.

4

CHAPTER 4

CLASSIFY EPILETIC EEG SIGNALS USING WEIGHTED COMPLEX NETWORKS BASED COMMUNITY STRUCTURE DETECTION

The EEG is a tool used to look at the electrical activity of the brain and the functioning of brain neurons. An abnormality in brain activity can be detected by an EEG test. Much EEG signal based research has been conducted to identify and trace the abnormities produced by epileptic seizures.

Epilepsy is one of the chronic neurologic disorders characterised by recurrent unprovoked seizures. Detecting epileptic seizures in EEG signals using transformation techniques, such as a wavelet transform, Fourier transform or hybrid transform does not give the promising results that EEG signals do. This is because EEG signals have a nonstationary and nonlinear nature.

Nonlinear features have been investigated by many researchers. To identify sleep stages, Acharya et al. (2010) compared and analysed 29 nonlinear measures such as high order spectra and recurrence quantification analysis. In this study, the extracted nonlinear features were ranked based on f-value. Acharya et al. (2013) also used a nonlinear technique based on high order spectra (HOS).

Diykh et al. (2016a, 2016b) found that complex networks yield promising results in EEG signals' classification. They showed that mapping the statistical features of EEG signals into complex networks improves classification results.

To analyse epileptic EEG signals, Diykh et al. (2017) employed a weighted complex network to analyse multi-channel EEG signals. This study found that the complex networks based community structure detection are capable of identifying abnormalities in EEG such as epileptic seizures.

4.1 Diykh et al., (2016b) "CLASSIFY EPILETIC EEG SIGNALS USING WEIGHTED COMPLEX NETWORKS BASED COMMUNITY STRUCTURE DETECTION"

The paper published by Diykh et al., (2016b), Classify epileptic EEG signals using weighted complex networks based community structure detection.

Expert Systems With Applications 90 (2017) 87–100

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Classify epileptic EEG signals using weighted complex networks based community structure detection

Mohammed Diykh[a,b,*], Yan Li[a,c], Peng Wen[a]

[a] School of Agricultural, Computational and Environmental Sciences, University of Southern Queensland, Australia
[b] Thi-Qar University, College of Education for Pure Science, Iraq
[c] School of Electrical and Electronic Engineering, Hubei University of Technology, China

ARTICLE INFO

Article history:
Received 11 May 2017
Revised 4 August 2017
Accepted 5 August 2017
Available online 9 August 2017

Keywords:
Epileptic EEG signals
Modularity
Statistical features
Weighted complex networks

ABSTRACT

Background: Epilepsy is a brain disorder that is mainly diagnosed by neurologists based on electroencephalogram (EEG) recordings. Epileptic EEG signals are recorded as multichannel signals. A reliable technique for analysing multi-channel EEG signals is in urgent demand for the treatment and diagnosis of patients who have epilepsy and other brain disorders.

Method: In this paper, each single EEG channel is partitioned into four segments, with each segment is further divided into small clusters. A set of statistical features are extracted from each cluster. As a result, a vector of all the features from each EEG single channel is obtained. The resulting features vector is then mapped into an undirected weighted network. The modularity of the networks is found to be the best to detect epileptic seizures in EEG signals. Other local and global network features, including clustering coefficients, average degree and closeness centrality, are also extracted and studied. All the network attributes are ranked based on their potential to detect abnormalities in EEG signals.

Results: Eight pairs of combinations of EEG signals are classified by the proposed method using four well known classifiers: a least support vector machine, *k*-means, Naïve Bayes, and *K*-nearest. The proposed method achieved an average of 98%, 96.5%, 99%, rand 0.012, respectively, for its accuracy, sensitivity, specificity and the false positive rate. Comparisons were made using several existing epileptic seizures detection methods using the same datasets. The obtained results showed that the proposed method was efficient in detecting epileptic seizures in EEG signals.

1. Introduction

Epilepsy is a general term that is used to describe a patient's condition with frequent and spontaneous seizures (Du, Dua, Acharya, & Chua, 2012; Hassan, Siuly, & Zhang, 2016; Swami, Gandhi, Panigrahi, Tripathi, & Anand, 2016). Based on World Wide Web information, over 1–2% of the world population endure neurological abnormalities by which the quality of their lives is impaired (Andrzejak et al., 2001 and Bhardwaj, Tiwari, Krishna, & Varma, 2016). In 2009 the World Health Organization announced that over 50 million people would experience epileptic seizures, and over 100 million were suffered an epilepsy episode (Orosco, Correa, Diez, & Laciar, 2016; Siuly, Li, & Wen, 2011; Swami et al., 2016). The exact cause of the most epilepsy cases is still unknown and mysterious. However, some cases of epilepsy could be the result of a brain injury, brain tumour or genetic factors (Delanty, 2014). Severe epileptic seizures can cause excessive synchronized or abnormal activities in the brain cortex (Orosco et al., 2016; Pippa et al., 2016; Upadhyay, Padhy, & Kankar, 2016). This happens because brain neurons produce an abrupt surge of electrical activities by which the cognition and behaviour of patients change (Al Ghayab, Li, Abdulla, Diykh, and Wan, 2016 and Alam et al., 2011). The clinical signs of an epileptic seizure range from an uncontrollable movement of the arms and legs to lose consciousness and unawareness (Bhardwaj et al., 2016). Hence, an untimely seizure occurrence could make a patient experiencing stress and anxiety for a long time period.

Electroencephalogram (EEG) signals are an important tool in the diagnosis of different neural disorders and diseases (Gotman, Flanagan, Zhang, & Rosenblatt, 1997; Tzallas, Tsipouras, & Fotiadis, 2009). They are the electrical signals recorded from the brain by means of voltage fluctuations using electrodes placed on a subject scalp (Nguyen-Ky, Wen, & Li, 2009 and Zamir, 2016). Based on clinical research findings, any irregular activity in the brain neurons leaves a signature on EEG signals (Bhardwaj et al., 2016). Con-

* Corresponding author at: School of Agricultural, Computational and Environmental Sciences, University of Southern Queensland, Australia.
E-mail addresses: mohammed.diykh@usq.edu.au (M. Diykh), Yan.Li@usq.edu.au (Y. Li), Peng.Wen@usq.edu.au (P. Wen).

http://dx.doi.org/10.1016/j.eswa.2017.08.012
0957-4174/

sequently, many advanced algorithms have been developed based on EEG signals to identify and trace the abnormities produced by epileptic seizures (Guerrero-Mosquera, Trigueros, Franco, & Navia-Vázquez, 2010; Kelly et al., 2010; Klatchko, Raviv, Webber, & Lesser, 1998).

Epileptic seizures detection using neural networks has been explored by Gabor and Seyal (1992), the authors detected epileptiform patterns in EEGs. A feed-forward and back-propagation artificial neural network was employed to identify those epileptiform patterns. Özdaa, Zhu, Yaylali, and Jayakar (1992) designed a real online system to detect spikes in EEG signals. A neural network was used to design that system. Webber, Lesser, Richardson, and Wilson (1996) developed a detector to identify seizures in EEG signals. The detector system was designed using an artificial neural network. Eberhart, Dobbins, and Webber (1989) designed a spike detector system to analyse epileptic EEG signals. In that study, a neural network was used in the training phase.

Time domain analysis has been widely applied in EEG research, especially for epileptic seizures detection (Du et al., 2012; Gajic, Djurovic, Gligorijevic, Di Gennaro, & Savic-Gajic, 2015). Kabir and Zhang (2016) detected epileptic seizures using an optimum allocation technique (OAT) and a statistical model. Du et al. (2012) applied principle component analysis (PCA) with a classification method to classify epileptic EEG signals. Siuly and Li (2015) integrated the PCA with the OAT to detect epileptic seizures in EEG signals. The PCA was employed to make EEG data uncorrelated, and reduced the dimensionality of signals, while the OAT was used to figure out the most representative features from EEG signals. Different classifiers were utilized in that study to find out the most powerful classification method for the extracted features. Al Ghayab et al. (2016) reported an epileptic seizure detection approach based upon a random sampling technique and a feature selection technique. Although, the reported studies achieved high classification accuracies, some of the recent studies have shown that the frequency domain is not robust for analysing all types of EEG data due to its non-stationary and chaotic behaviours of EEG signals.

Wavelet and Fourier transforms have been used as an appropriate technique to analyse EEG signals by many researchers (Adeli, Zhou, & Dadmehr, 2003; Bhardwaj et al., 2016; Das, Bhuiyan, & Alam, 2016; Guler and Ubeyli, 2007; Guo, Rivero, & Pazos, 2010; Khan and Gotman, 2003; Murugavel & Ramakrishnan, 2016; Ocak at al., 2009; Samiee, Kovács, & Gabbouj, 2015; Torres, Colominas, Schlotthauer, & Flandrin, 2011; Xie and Krishnan, 2013; Zarjam, Mesbah, & Boashash, 2003). Kumar, Dewal, and Anand (2014) detected epileptic seizures in EEG signals by using a wavelet transform, and an approximate entropy. The EEG signals were decomposed into five-level and the approximate entropy was calculated. Two machine learning approaches: a support vector machine (SVM) and a neural network, were used in that study. Orosco, Correa, Diez, and Laciar (2016) applied a stationary wavelet transform combined with a features selection technique to identify epileptic seizures in EEG signals. The spectral features from EEG signals were extracted and investigated in that research. A stepwise analysis was employed to find out the best features set to discover epileptic seizures. Bajaj and Pachori (2012) used empirical mode decomposition (EMD) as a features extractor to classify EEG signals into epileptic and non-epileptic segments. A set of bandwidth features were used as the input to a SVM. Hassan et al. (2016) propounded a tunable-Q factor wavelet transform (TQWT) and a bagging technique for detecting epileptic seizures in EEG signals. In that study, six spectral features were adopted and extracted after applied the TQWT. A combination of multi-classifiers was used as an ensemble method to classify the extracted features. Guler and Ubeyli (2007) proposed an automatic seizure detection method using wavelet coefficients and Lyapunov exponent. A SVM was employed in that study to distinguish EEG categories. All the reviewed studies in this section were conducted using Bonn University data. Based on those work, the wavelet transform was used as a fundamental technique for features extraction and analysing EEG signals. However, one of the major disadvantages of using wavelet transforms is that any slight shift in EEG signals can cause perturbation to the wavelet coefficients (Fu, Qu, Chai, & Dong, 2014).

Nonlinear features have been also employed in analysing epileptic EEG signals. Acharya et al. (2013) analysed epileptic EEG signals using a continuous wavelet transform (CWT), high order spectra (HOS) and textures. In that study, the HOS and textures were extracted from the CWT plots, and then they were used to detect seizures in EEG signals. Murali, Chitra, Manigandan, and Sharanya (2016) used recurrence quantification analysis (RQA) to detect epileptic seizures. EEG signals were pre-processed using notch, wavelet and adaptive filters. Different RQA parameters were extracted and investigated from the filtered EEG signals. Chandran, Acharya, and Lim (2007) studied a set of HOS features in seizures detection. The ANOVA test was used in that study to examine the extracted HOS features. Song, Crowcroft, and Zhang (2012) utilized two nonlinear features: sample entropy and approximation entropy, to detect epileptic seizures. The extracted features were optimized using an optimization algorithm. Sharma, Pachori, and Acharya (2017) used an analytic time-frequency flexible wavelet transform (ATF-FWT) for detecting seizures in EEG signals. In that paper, EEG signals were decomposed using the ATF-FWT to obtain the desired bands. Fractal dimension was calculated from each band to form the final features set. The features were ranked and reduced using *t*-test, and then were fed to a LS-SVM. Bhattacharyya, Pachori, and Acharya (2017) employed a tunable-Q wavelet to decompose multichannel EEG signals into different bands. Multivariate fuzzy entropy was computed from sub-band EEG signals falling in the same oscillatory level. The extracted features were fed to a random forest and a LS-SVM. Patidar and Panigrahi (2017) also applied a tunable-Q wavelet to classify epileptic EEG signals. Kraskov entropy was calculated from a specific band after decomposing EEG signals by the tunable-Q wavelet. A LS-SVM was used to classify the extracted features.

Detecting epileptic seizures have been also investigated using visibility graphs. Tang et al. (2013) applied the visibility graph concept to discover the high frequencies in electrocorticography (ECoG) signals. A sequence degree was used in epileptic seizures recognition. Zhu, Li, and Wen (2014) proposed a weighted horizontal visibility graph approach to identify seizures in EEG signals. Two graph characteristics including clustering coefficients and mean degree were extracted. The obtained results were compared with those by a Fast Fourier transform. All the previous research based on visibility graphs included all the data points of EEG signals and considered only structural properties of visibility graphs in the detection phase. Using all EEG data require huge computation memory and also slow down the speed of detection epileptic seizures in EEG signals.

EEG signals exhibit nonlinear behaviours. One of the effective non-linear methods is to apply complex networks concept. Based on our previous studies (Diykh & Li, 2016; Diykh, Li, & Wen, 2016), we found that unweighted complex networks yielded promising results in analysing single channel sleep EEG signals. In this paper, we used modularity detection from a weighted complex network to further improve the performance of analysing multichannel EEG signals. From the obtained results, it was found that the modularity from the weighted complex networks was capable of identifying seizures in EEG signals. In this paper, the statistical characteristics of EEG signals were extracted by using a segmentation technique. It was reported in Diykh and Li (2016) that network attributes constructed from statistical characteristics of EEG

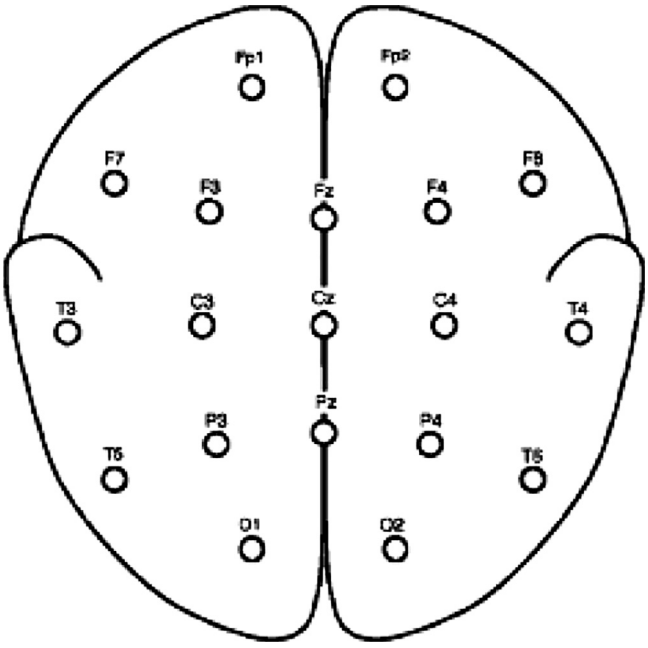


Fig. 1. Scheme of the 10–20 system.

signals reflected abnormal behaviours in EEG signals. In this paper, after making a thorough investigation, each single channel EEG signals is segmented into windows of 1024, 1024, 1024 and 1025, in order to consider signals quasi stationary, and each segment is also spilt into 32 clusters. 12 statistical characteristics are extracted from each cluster. As a result, the dimensionality of each segment is reduced from 1024 to 384 data points. A vector of statistical features is then mapped into a weighted undirected network. To detect the abnormal patterns of epileptic seizures in EEG signals, the modularity and other network characteristics are extracted and analysed. Different machine learning techniques, including unsupervised machine learning, such as least support vector machine (LS-SVM), Naïve Bayes, and supervised algorithms, like *k*-means and *K*-nearest, are used to categorize networks characteristics into different EEG cases. The core objective of this paper is to improve the detection accuracy of epileptic seizures in EEG signals by integrating the complex networks concept with a statistical model, and also to investigate the best network features for analysing multichannel EEG signals.

2. EEG data description

The epileptic EEG data used in this research are available online from the URL link of http://epileptologie-bonn.de/cms/front_content.php?idcat=193&lang=3. The datasets are considered clinical EEG benchmark dataset for most of the epileptic research. The details of the datasets are available in Andrzejak et al. (2001).

The EEG recordings were collected at epilepsy department in Bonn University, Germany. The international 10–20 system was used to acquire EEG signals. Fig. 1 shows the scheme of the locations of surface electrodes based on the 10–20 system. The names of the electrode positions were from their anatomical locations. Datasets A and B were taken from all the electrodes. Depth electrodes were implanted symmetrically into the hippocampal formations (top). Sets C and D were acquired from all the depth electrodes. On the lateral and basal regions (middle and bottom) of the neocortex, strip electrodes were implanted. Set E were taken from all the depicted electrodes. Fig. 2 illustrates the scheme of intracranial electrodes implanted for pre-surgical evaluation of epileptic patients.

Fig. 2. Intracranial electrodes implanted for presurgical evaluation of epileptic patients (Andrzejak et al., 2001).

A 128-channel amplifier with an average common reference was used with all the EEG recordings. The EEG recordings were sampled at 173.61 per second using a 12 bit resolution. A bandpass filter with 0.3–40 Hz was used. Each EEG group includes 100 recordings (100 single EEG channels) of 23.6-s. Data of sets A–E were collected from 10 patients. Groups A and B were acquired from five healthy participants relaxing in an awake stage with eyes open in group A, and eyes closed in group B. Groups C, D and E were from EEG recordings of five epileptic patients. Signals in groups C and D were obtained from five participants during seizure free. Group D was recorded from epileptogenic area, while group C was from hippocampal area. Group E was selected from all recording sites exhibiting ictal activity. All mentioned groups were used in this study to evaluate the proposed method. Fig. 3 shows an example of different EEG group signals.

3. Methodology

A novel epileptic seizure detection method is presented in this paper. Firstly, the dimensionality of EEG signals is reduced. Reducing the dimensionality of EEG signals before transferring them to complex networks is a very important step to eliminate the irrelevant information, and to reduce the computation time of the proposed method. EEG signals are periodic, non-stationary and contain a lot of redundant data. Based on our previous work (Diykh & Li, 2016; Diykh et al., 2016), it was found that mapping an entire EEG segment into a network without dimensionality reduction increased the complexity time of the proposed method and also degraded the detection accuracy.

To distinguish EEG signals, 12 statistical features are pulled out. So that the irrelevant data of signals can be eliminated. Each single channel EEG signal is parted into four segments of 1024, 1024, 1024 and 1025. Each segment is then partitioned into clusters, the number of clusters is empirically determined. A vector of statistical features is extracted representing one channel of EEG data. The vector of features is then transferred into a weighted undirected complex network. The behaviour of the network is examined to determine whether it reflects the changes in the EEG signals during epileptic seizures and non-epileptic seizures events. The modularity of the networks and other global and local network characteristics are investigated. Those network characteristics are analysed using statistical measurements to identify their ability and strength to detect epileptic seizures in EEG signals. Different classifiers including supervised machines learning and non-supervised machines learning algorithms are used with the proposed method to test its reliability and stability to identify epileptic seizures as well as to figure out an effective classifier for categorizing network's characteristics into different EEG categories. Fig. 4 depicts the outline of the proposed method. All the experiments are conducted with the datasets described in Section 2.

3.1. EEG dimensionality reduction

EEG signals are periodic, non-stationary and contain a lot of redundant data. It is necessary to search for more accurate approach

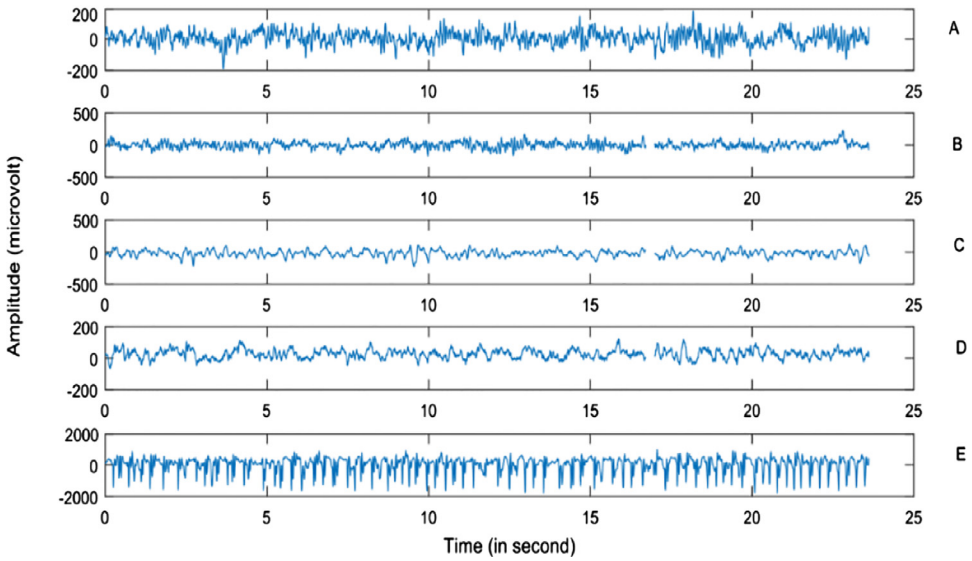


Fig. 3. An example of different EEG cases (A–E).

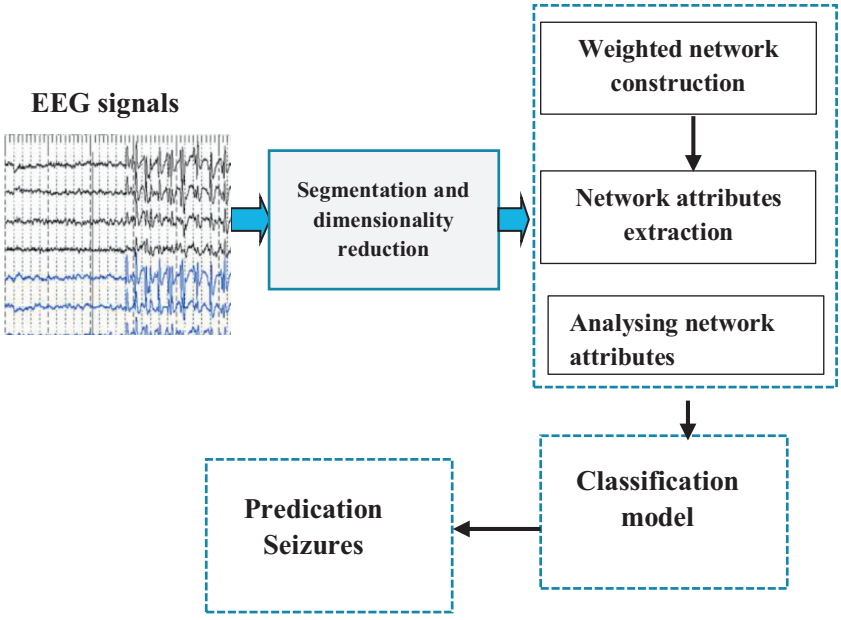


Fig. 4. The methodology of the proposed method.

to reduce the dimensionality of EEG signals. So that the important information are kept and the irrelevant data are eliminated. Based on our previous work (Diykh & Li, 2016; Diykh et al., 2016), it was found that there is a big difference in the classification accuracy when different statistical features of EEG signals are used. Each single EEG channel which contains 4097 data points is partitioned into four segments of 1024, 1024, 1024 and 1025, respectively. This segmentation is carried out based on a specific period of time, which is 5.9 s for each segment. To extract the statistical features, each segment is also spilt into 32 clusters. The number of the clusters was empirically selected during the training session. This means that at each stage of the segmentation the statistical features are selected and sent to the proposed method. The segmentation process is stopped until the desired classification obtained or there were no further improvement in the classification results. 12 features are pulled out from each cluster. As a result, the dimensionality of each segment was reduced from 1024 to 384 data points. The 12 features of {*median, maximum, minimum, mean, mode, range, first quartile, second quartile, standard deviation, variation, skewness, kurtosis*} are considered as the key features to represent EEG data in this study. Table 1 provides a short explanation for the statistical features. They are denoted as $\{X_{Me}, X_{Max}, X_{Min}, X_{Mean}, X_{Mod}, X_{Rand}, X_{Q1}, X_{Q2}, X_{SD}, X_{Var}, X_{Ske}, X_{Kue}\}$. After the segmentation, each single EEG channel contains 1536 (4×384) data points. Fig. 5 shows the segmentation and dimensionality reduction schema.

3.2. From EEG statistical features to weighted complex networks

Complex networks have the potentials to characterize the hidden patterns in a signal as it can reflect the dynamic properties of data (Bashan, Bartsch, Kantelhardt, Havlin, & Ivanov, 2012; Diessen, Diederen, Braun, Jansen, & Stam, 2013). The prior research have showed that complex networks are robust to noise (Diykh & Li, 2016). Our previous work revealed that the behaviour of complex networks vary based on EEG sleep stages. The relationship be-

Table 1
Mathematical details of the statistical features.

No.	Feature name	Formula	No.	Feature name	Formula
1	Max	X_{Max}=Max[x_n]	7	Min	X_{Min}=min[x_n]
2	Mean	$X_{Mean}=\frac{1}{n}\sum_{1}^{n}x_i$	8	Mode	$X_{Mod}=L+(\frac{f_1-f_0}{2f_1-f_2})Xh$
3	Median	$X_{Me}=(\frac{N+1}{2})^{th}$	9	Range	X_{Rang}=X_{Max}-X_{Min}
4	1st Quartile	$X_{Q1}=\frac{1}{4(N+1)}$	10	Standard deviation	$X_{SD}=\sqrt{\sum_{n=1}^{N}(x_n-AM)}\ \frac{2}{n-1}$
5	variation	$X_{Var}=\sum_{n=1}^{N}(x_n-AM)\frac{2}{N-1}$	11	Skewness	$X_{Ske}=\sum_{n=1}^{N}(x_n-AM)\frac{3}{(N-1)SD^3}$
6	Kurtosis	$X_{Ku}=\sum_{n=1}^{N}(x_n-AM)\frac{4}{(N-1)SD^4}$	12	2nd Quartile	$X_{Q2}=\frac{4}{4(N+1)}$

where $X_n = 1, 2, 3, \ldots, n$, is a time series, N is the number of data points, AM is the mean of the sample.

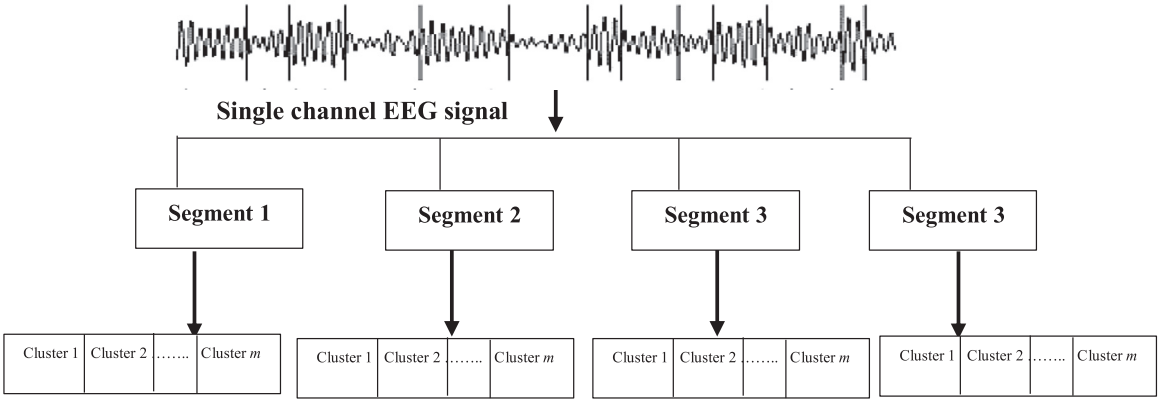


Fig. 5. The segmentation and dimensionality reduction schema.

tween the network topology characteristics and how they change with transitions across epileptic EEG signals are still unknown. In this work, it is demonstrated that epileptic seizures are associated with the behaviours of the networks, and can be reflected through different topological characteristics.

In this study, we adopt a weighted undirected network to represent EEG data whose edge weights are a natural number. Let **G** be a weighted undirected graph (***V***, ***E***, ***W***), where ***E*** is a set of all links among the graph nodes, $\boldsymbol{E} \in \boldsymbol{V}$ with a weight belonging to ***W***. We assume that all the weights in a network are non-negative. Two nodes, v_i and v_j, are connected, and their edge is defined by their Euclidean distance.

Therefore, the weight of each edge is calculated according to the following formula

$$w_{v_i,v_j}=\frac{d(v_i,\ v_j)}{d_{max}} \tag{1}$$

where d_{max} is the longest distance among all the points, and $d(v_i, v_j)$ was the distance between nodes v_i and v_j.

Consider a vector of statistical features, denoted by $R=\{x_1, x_2, x_3, \ldots\ldots,\ x_n\}$, that represents a single EEG channel. Based on Zhang and Small (2006) and our previous work (Diykh & Li, 2016), vector R is mapped as a weighted network, considering each data point in R as a node in graph G.

Fig. 6 shows a time series of $R=\{x_1=32,\ x_2=8.3,\ x_3=7.9,\ x_4=3.1,\ x_5=8.2,\ x_6=9.5,\ x_7=1.0,\ x_8=0.8,\ x_{10}=6.2,\ x_{11}=2.9,\ x_{12}=7.8,\ \ldots\ldots\}$, being transferred into a weighted network. x_1 is the first node in the network, corresponding to the first point (the first statistical feature in R) with a value of 32. The edge between nodes x_1 and x_9 has a weight corresponding to a value of 1.9. Based on the proposed method, the network can be characterised with its modularity, average degree, closeness and cluster coefficients.

In this paper, a threshold is defined to eliminate the nodes with poor connections. Each connection that is lower or equal to a predefined threshold (δ) is eliminated.

$$(v_i,\ v_j)\ \in\ \boldsymbol{E},\ if\ d(v_i,\ v_j)\le\ \delta \tag{2}$$

It can be noticed that, nodes x_{11} and x_{12} in Fig. 6 have no connections with other nodes in the networks. This means that nodes x_{11} and x_{12} are isolated nodes in the network.

The adjacency matrix **A** of graph **G** is calculated for all **V** to describe the connections of the network nodes. The adjacent matrix of an undirected weighted graph is symmetric, i.e. **A** (v_1, v_2)= **A** (v_2, v_1)

$$\boldsymbol{A}(v_i,\ v_j)=\begin{cases} w(\boldsymbol{e}_{v_i,v_j}),\ \text{if}\ (v_i,\ v_j)\in E \\ 0,\ \text{otherwise}\end{cases} \tag{3}$$

One of many interesting observations in this paper is that the network connectivity is significantly stronger in epileptic signals than in non-epileptic signals. The relationship between the networks and how their behaviours change with EEG signals have not been studied through different characteristics. The next section provides more details in regarding to the network characteristics.

3.3. Topological and structural network properties

The following network characteristics are considered in epileptic seizures detection in this paper (Antoniou & Tsompa, 2008; Barrat, Barthelemy, Pastor-Satorras, & Vespignani, 2004; Bullmore and Sporns, 2009; Clauset, Newman, & Moore, 2004; Diykh & Li, 2016; Song, Havlin, & Makse, 2005).

3.3.1. Community detection in networks (modularity)

Modularity is used to measure the strength of partitioning a network into groups of nodes. Each group of nodes in a network must have a high density of connections among them, and a lower density of connections with other groups. This feature is commonly used in social networks to detect different communities and

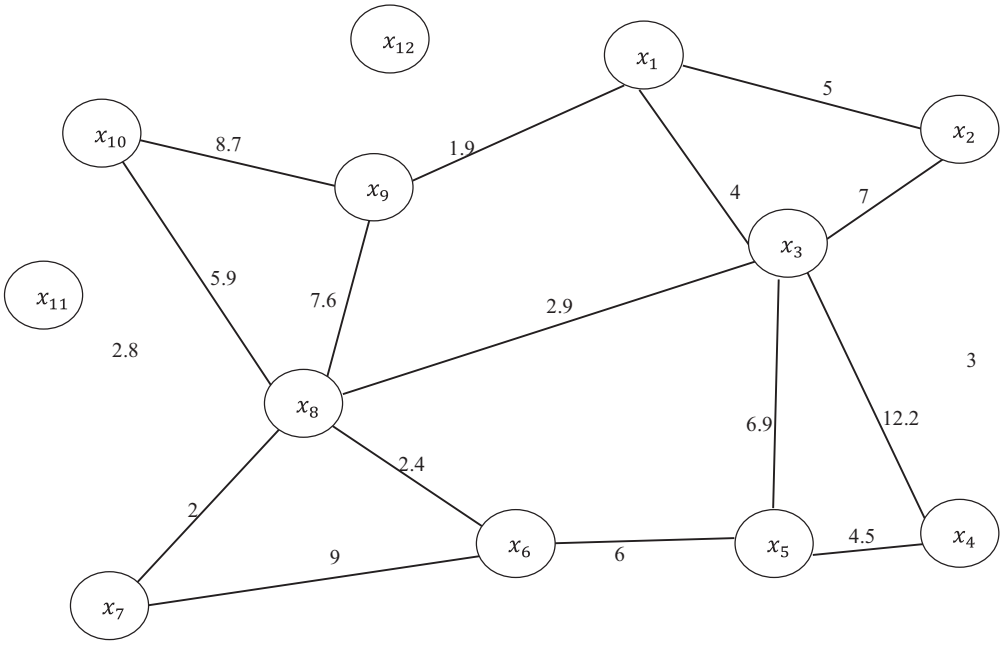


Fig. 6. A vector of statistical features is being transferred into a weighted undirected network.

the relationships among communities. Suppose $A_{i,j}$ is an adjacency matrix of a graph G, and $m = 1/2\ \sum_i k_i$ denotes the total number of the links. Where k_i refers to the degree of node i, and C_i is the cluster name of node i. The mathematic formula of modularity is defined as

$$Q = \frac{1}{2m}\sum_{i,j}\left(A_{i,j} - \frac{k_i k_j}{2m}\right)\delta(C_i C_j) \tag{4}$$

where $\delta(c_i c_j)$ takes value 1 if two nodes v_i, v_j belong to the same cluster, otherwise it is equal 0.

The main drawback of the above formula is that it requires a high computation time, especially when it deals with a network of a huge number of nodes (Newman, 2004, 2003). A fast algorithm for optimizing modularity was suggested by Clauset et al. (2004) to avoid the drawback. However, one of the weaknesses of the algorithm by Clauset et al. (2004) is that it produces a large fraction of nodes and the value of the modularity is lower than the modified version by Blondel, Guillaume, Lambiotte, and Lefebvre (2008). In this paper, the algorithm of Blondel et al. (2008) is used to calculate the value of the modularity. The gain in the modularity is computed by

$$\Delta Q_{ab} = \left[\frac{\sum_{bn} k_{a,\mathrm{bn}}}{2m} - \left(\frac{\sum_{tot}+k_a}{2m}\right)^2\right] - \left[\frac{\sum_{bn}}{2m} - \left(\frac{\sum_{tot}}{2m}\right) - \left(\frac{k_a}{2m}\right)^2\right] \tag{5}$$

where $\sum_{bn}$ refers to the total weight of the links for all nodes in community b; $\sum_{tot}$ denotes the sum of the weights of the links that connect nodes in the community; $k_{a,bn}$ is the total weights of the links from community a to community b; k_a is the total weights of the links connected to node a; m is the total weights of all the links in the network. During the simulation phase, the aforementioned formulas were tested and it was found that the algorithm by Blondel et al. (2008) gave better results than the one by Clauset et al. (2004).

3.3.2. Average degree of the networks

The degree of node i in a weighted network is defined as the sum of all the weights of all the nodes connected to node v_i. The following equation is the mathematic formula of node degree:

$$d_i = \sum_{j\in\prod(l)} w_{ij} \tag{6}$$

where j represents all the connected nodes to v_i, and w_{ij} refers to the weight of the link between nodes v_i and v_j. The node degree formula is extended to express the degree of the entire network. The degree of network is formulized as

$$D = \sum_i d_i \tag{7}$$

3.3.3. Closeness centrality

Closeness Centrality refers to an important node in a network that can be quickly reached by other nodes in the network. The closeness centrality is formulized as:

$$C_{clos} = \frac{1}{\sum_{i\in V} dist(v_i, v_j)} \tag{8}$$

where, $dist(v_i,v_j)$ indicates the shortest path between nodes v_i,v_j.

3.3.4. Cluster coefficients (CC)

The local group cohesiveness in networks can be measured using the clustering coefficients CC and it is defined as the fraction of the connected neighbours to node v_i. The global formula of clustering coefficients is defined as

$$CC_{v_i} = 1/2\left(\frac{\mathrm{M1}}{\mathrm{M2}}\right) \tag{9}$$

where $m1$ is the number of the actual links between node v_i with its neighbours, and $m2$ is the number of the neighbours of v_i. The average of clustering coefficients is used to measure the density of interconnected nodes by triples. It can be expressed as

$$CC_{ave} = \frac{1}{N}\sum_{i=1}^{l} CC_{v_i} \tag{10}$$

The above formula could not be effective enough with weighted networks. In this work, we considered the weighted clustering formula that is defined by Opsahl and Panzarasa (2009).

$$C_i^w = \frac{total\ value\ of\ closed\ triples}{total\ value\ of\ triples} = \frac{\sum_{t\Delta} w}{\sum_t w} \quad (11)$$

where, $\sum_t w$ is the total number of triples, and $\sum_{t\Delta} w$ is a subset of triples that are close to each other's

3.4. Machine learning techniques

This section discusses the four algorithms, the LS-SVM, *k*-means, Naïve Bayes and *k*-nearest. Based on the literature (Diykh et al., 2016; Gular et al., 2007; Sahi, Lai, Li, & Diykh, 2017; Siuly et al., 2011; Wang, Li, Wen, & Lai, 2016; Zhu et al., 2014), we found that those four classifiers are the most popular and effective methods in biomedical signals classification.

3.4.1. Least square support vector machine (LS-SVM)

The LS-SVM was developed by Suyken and Vandewalle (Guler and Ubeyli, 2007) based on the last version of a support vector machine. Recently the LS-SVM has been widely used to classify various types of biomedical signals due to its superior performance to differentiate data with a high classification accuracy and a minimum time execution. Al Ghayab et al. (2016), Siuly and Li (2014) and Siuly, Li, and Wen (2009), Siuly et al. (2010, 2011) utilized the LS-SVM to detect seizure patterns in EEG signals.

To obtain high classification performances with the LS-SVM, two main parameters, γ *and* σ, should be appropriately chosen. Those two parameters can influence the classification accuracy negatively if they are selected improperly. In this paper, the LS-SVM was used to classify epileptic EEG categories. The values of γ *and* σ were empirically set during the training session. More discussions regarding selecting those parameters are in the Experimental Results and Simulations section.

3.4.2. Naive Bayes

Naïve Bayes (NB) is a statistical classifier which can predict the probabilities of class memberships. In biomedical signal processing, the NB yields a high accuracy and low computation time with large databases. To simplify the computation, the NB depends on an assumption that the effect of an observation in a given class is independent of the values of the other observations. This assumption defines as a class conditional independence. Assume $\boldsymbol{Y} = \{y1, y2, \ldots\ldots yn\}$ is a sample that comprises a set of *n* points. Let $\boldsymbol{H}$ represent a hypothesis of each set of $\boldsymbol{Y}$ that belongs to a specific class. In Naive Bayes rules, it considers $\boldsymbol{Y}$ as an evidence and seeks to assign each point of $\boldsymbol{Y}$ *to* the highest posteriori probability class (John and Langley, 1995). In this paper, Naive Bayes is also used to classify the characteristics of the complex networks.

3.4.3. K-nearest neighbour classifier

It is a simple and common classification method as it is robust with noisy and large datasets. It is also adaptive in nature because of using local information for the prediction of unknown data. It performs the classification task on the basis of the frequent class of its nearest neighbours in the feature space (Wilson and Martinez, 2000). Several distance metrics are used to define the distance in the KNN algorithm. Based on the training session, the Euclidean distance is used in this paper. In order to classify an instance, the similarities with *K*-nearest neighbour are computed and the class corresponding to the maximum number of votes is assigned as the output class of the instance.

3.4.4. k-means

k-means is one of the unsupervised classification methods, which is mainly designed to solve clustering problems. The method separates a given population into a number of clusters based on defining *k* centroids for each cluster. The process is achieved by minimizing the Euclidean distance between an observation and the cluster centroid. Each observation belonging to the given population is associated to the nearest centroid. This step is repeated at each iteration to obtain the first level of clustering, and the new *k* centroids are calculated. In this paper, *k*-means is used to classify the attributes of the graphs.

3.5. Performance evaluation measurements

In order to evaluate the performance of the proposed method with different EEG categories, the following statistical measures are utilized.

- Sensitivity: a statistical measure by which the performance of a classification algorithm is assessed by computing the proportion of the actual positive classification. It is defined as

  $$Sensitivity = \frac{TP}{TP + FN}$$

  where TP= true positive, and FN= false negative.
- Specificity: it measures the rate of negative cases that are correctly defined. The main formula of specificity is defined as

  $$Specificity = \frac{TN}{TN + FP}$$

  where TN= true negative, and FP=false positive.
- Accuracy: it is a statistical measure by which the performance of a classification algorithm is evaluated. It is computed through dividing the number of the samples correctly classified by the total number of all the samples.

  $$Accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

- False positive rate (*FPR*): it represents the rate of non-seizure healthy volunteers being categorised as seizure patients.

  $$FPR = \frac{FP}{FP + TN}$$

4. Experimental results and simulations

To assess the ability of the proposed method to detect the epileptic seizures in EEG, a number of experiments were made. Different network features were tested and evaluated. The main purpose of selecting different network characteristics was to examine the relationships between complex network features and the changes in EEG signals during epileptic seizures and non-seizures events. The statistical features of EEG signals were extracted and they were transferred into a weighted network. It was found that the differences among various cases of EEG signals can be revealed through the behaviours of the complex networks. Those differences can be captured using network characteristics. Different network attributes were tested. The datasets used to evaluate the proposed method consists of five EEG data sets (A–E) collected from 10 patients. All the datasets were used to evaluate the proposed method. More details regarding the EEG sets were explained in experimental data section. The experiments were conducted using Matlab Version: R2013, on a computer with the following features: 3.40 GHz Intel(R) core(TM) i7 CPU processor machine, and 8.00GB RAM. The proposed method was evaluated using the above different statistical measurements.

In this section, eight different experiments were designed in order to obtain a clear picture of the performance. The experimental results were obtained under the following experiments:

1. The first experiment (Exp.1) for {A vs E}: it was designed to investigate the ability of the proposed method to distinguish the EEG data in group A (healthy subjects, eyes open) and group E (epileptic seizures).
2. Exp.2 was made to recognize the EEG data in group B (healthy, eyes closed) and group E: {A vs E}.
3. Exp.3 was carried out for group C (seizure free from hippocampal area) and E: {C vs E}.
4. Exp.4 was designed to identify the EEG data in group D (seizure free from epileptogenic area), and group E: {D vs E}.
5. Exp.5 was made in which a combination of EEG groups of A, B and E was used to assess the performance of the proposed method. The EEG data were classified into two categories: non-ictal interval belonging to groups {A and B}, and ictal interval belonging to group E: {(A, B) vs E}.
6. In Exp.6, a combination of groups C, D and E was used. As mentioned before, non-ictal interval belonging to groups {C and D}, and ictal interval belonging to group E: {(C, D) vs E}.
7. In Exp.7, a combination of EEG groups of A, C, D and E was used. The EEG data are classified into two categories: non-ictal interval belonging to groups (A, C and D), and ictal interval belonging to group E: {(A, C, D) vs E}.
8. In Exp. 8, a combination of EEG groups of A, B, C, D and E was used. The EEG data are classified into two categories: non-ictal interval belonging to groups (A, B, C and D), and ictal interval belonging to group E: {(A, B, C, D) vs E}.

The EEG datasets were divided to equal numbers of segments for each group to avoid inconsistency. In each experiment, 100 segments from each group were used. For example, to differentiate groups {A and B} vs group E, 50 segments each were selected from group A and group B, and 100 segments were chosen from group E.

4.1. Analysing network features for identifying EEG categories

The selection of optimal features is considered a significant part for any classification problem. It was observed that when the features were not appropriately selected from a non-stationary and chaotic signals, the obtained classification results were not good enough. However, an important investigation in this paper was to figure out the impact of each network feature on classification accuracy. We investigated the accuracy by each feature to show that how each feature reflected the sudden abnormality in EEG signals. The datasets were divided into the training and testing sets. The number of segments in each group was balanced to avoid inconsistency as mentioned before. The four characteristics of the complex networks, namely *modularity, clustering coefficients, closeness centrality* and *average degree*, were extracted to identify different EEG signals categories. It was noticed that most of the network characteristics could reflect the sudden fluctuations in EEG signals during epileptic seizure activity. Our findings showed that network features were able to reveal the abnormalities in EEG signals, and to identify different EEG groups but with different levels of accuracy.

As a result, the network features were ranked as *modularity, average degree, clustering coefficients* and *closeness centrality* based on their potential to identify the EEG signals cases. To show how the network features can differentiate among EEG cases, Figs. 7–10 show the box plots of the four features of modularity, average degree, clustering coefficients and closeness centrality for each group of A–E.

Fig. 7 shows the network modularity of five EEG groups. The value of the network modularity ranges from 1 to −1. The maximum value of modularity was 1, representing that the community was strong and the network could be partitioned into small communities. One can notice that group E gained the lowest value while group D obtained the highest value among the five groups. Modularity showed a high potential to classify the EEG groups. Clustering coefficients was the second network feature that was used to characterize the set of A–E. Based on the obtained results, clustering coefficients could reflect the global and local connectivity of the networks. Fig. 8 shows the box plots of the five EEG groups of A–E using clustering coefficients. The values of clustering coefficients were in between [0, 1]. As presented in Fig. 8, set C had the highest clustering coefficients among the five groups, and the feature had the ability to identify the EEG groups of A–E. However, group E was recorded having the lowest values for clustering coefficients among EEG groups. The feature reflected the capability of the complex networks to detect the epileptic seizures in EEG signals.

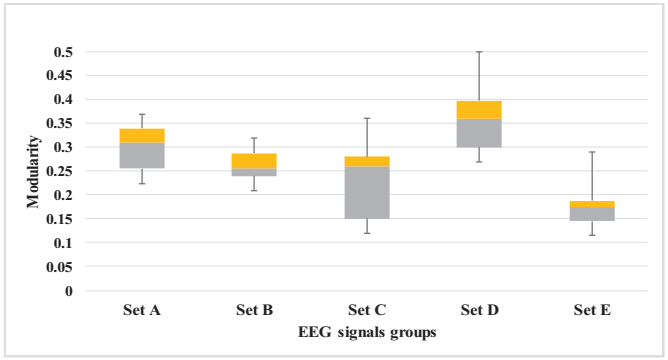


Fig. 7. Box plots of the modularity feature for each EEG case.

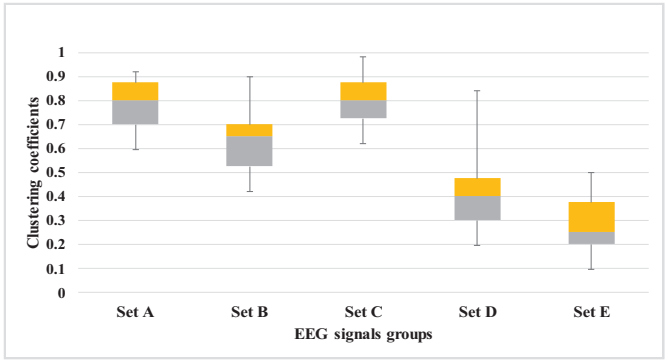


Fig. 8. Box plots of the clustering coefficients feature for each EEG case.

The node degree of the networks was also studied. It showed a positive reflection to detect the epileptic seizures in EEG signals. Fig. 9 presents the box plots of node degrees for the EEG groups of A–E. One can observe that there is a big difference among the five groups in regarding to the average degrees of the networks. Group E was recorded having the highest average degree among the five groups, with its values were in between 395 and 790. However, group D showed having the lowest values of the average degrees. This feature can be used to detect seizure activity in EEG signals and considered as a key feature in the networks. The network centrality was also tested to show whether it could reflect the changes in EEG signals for epileptic patterns. Fig. 10 shows the box plots for each EEG group using closeness centrality.

Based on the results in Fig. 10 the network centrality can be used to distinguish EEG cases. The closeness centrality for group E is between 0.7 and 1, which is very different from other EEG groups. A further investigation was made using Mann–Whitney *U*

Table 2
Results of Mann–Whitney *U* test of the networks characteristics for each pair of EEG cases.

p-value	{A vs E}	{B vs E}	{C vs E}	{D vs E}	{(A, B) vs E}	{(C, D) vs E}	{(A, C, D) vs E}	{(A, B, C, D) vs E}
Modularity	2.20e−16	5.201e−03	1.25e−15	0.0012	0.0035	0.0067	4.98e−10	1.9532-15
Clustering coefficients	3.21e−14	1.64e−12	0.0024	0.0014	1.89e−13	3.35e−14	0.0036	1.89e−10
Closeness	0.00456	0.00140	6.52e−12	0.00640	0.00780	2.97e−19	0.0078	0.00102
Average degree	4.21e−17	3.78e−19	0.0054	3.21e−19	0.0098	1.98−e210	1.12e−114	2.34e−17

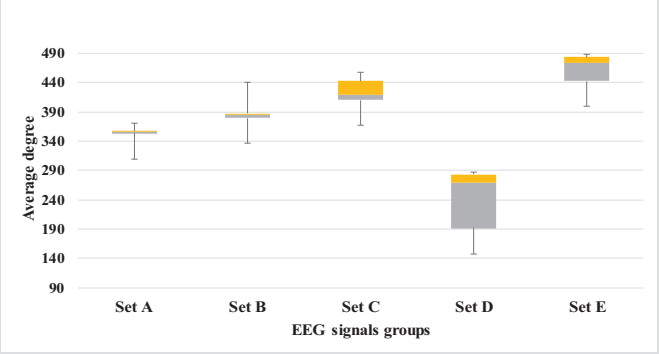


Fig. 9. Box plots of the average degree feature of each EEG case.

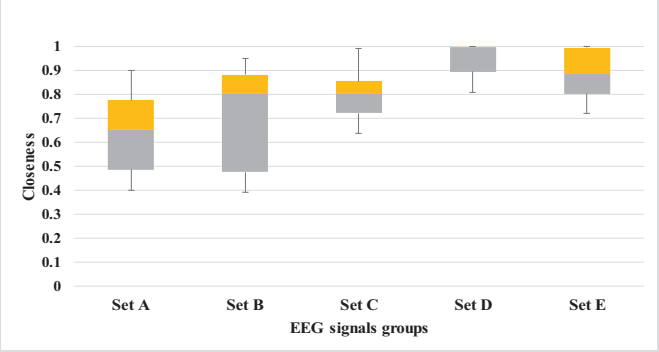


Fig. 10. Box plots of closeness centrality of each EEG case.

Test to identify the ability of the characteristics of the networks to recognize different EEG categories. The network characteristics of each EEG pair, including {A vs E}, {B vs E}, {D vs E}, {C vs E}, {(A, B) vs E}, {(C, D) vs E}, {(A, C, D) vs E}, and {(A, B, C, D) vs E} were investigated. The results of the testing were reported in Table 2.

It was found that all the characteristics of the networks were significant, with $p < 0.05$, and also different for various types of EEGs, reflecting their epileptic activities

4.2. Classification results

To evaluate the performance of the proposed method, extensive experiments were conducted. The network features were assessed separately to examine the ability of each network feature to recognize EEG groups. The LS-SVM was used as an unsupervised machine learning classifier. It was found that the RBF kernel function provided better results than the polynomial kernel for the LS-SVM. Its γ *and* σ^2 values were empirically chosen, and they were set to $\gamma = 1$ *and* $\sigma^2 = 1$.

The data were divided into training and testing sets, and all the results were recorded. Fig. 11 shows the accuracy of the proposed method based on each network feature as well as using all the network features. The performance of the proposed method was tested based on the network features. It was found that the modularity yielded a higher accuracy compared with the other network

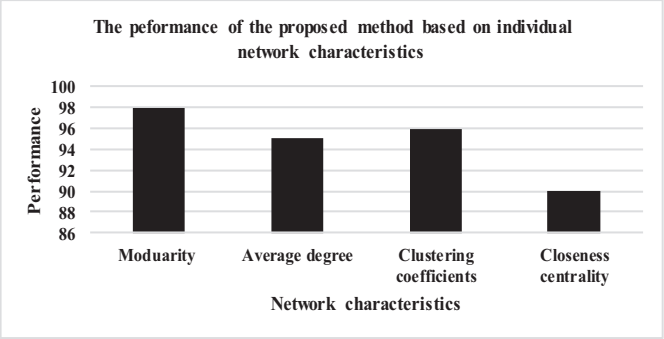


Fig. 11. The Performance of the proposed method based on the network features.

Table 3
Classification results using LS-SVM.

EEG group	Accuracy	Specificity	Sensitivity	FPR
{A vs E}	100%	100%	100%	0
{B vs E}	98%	95%	100%	0
{C vs E}	97%	95%	99%	0
{D vs E}	97%	95.5%	99%	0
{(A and B) vs E}	98%	98%	98%	0.025
{(C and D) vs E}	97.8%	96%	98%	0
{(A, C and D) vs E}	98%	96%	98%	0
{(A, B, C and D) vs E}	97.9%	98%	97%	0.037

features. It had the best discriminative capacity among the networks features. An average of 98% accuracy was achieved for all the EEG cases by modularity, although most of the other network features could also provide the acceptable classification results. Our findings showed that combining all the network features yielded a high classification accuracy for all the EEG groups with an average accuracy of 98% for epileptic EEG signal classification

Fig. 12 shows the performance of the proposed method across all the EEG groups based on the network attributes. The average accuracy for all the EEG groups was calculated based on each network feature. The best performance was achieved by modularity with a 98% accuracy, followed by average degree, clustering coefficients, and closeness centrality.

To evaluate the proposed method with other statistical measures, Table 3 reports the classification results in terms of accuracy, sensitivity, specificity and the FPR for all the EEG cases. Based on the results, we considered the combination of the network characteristics as the key features in all the experiments to differentiate EEG cases.

The results in Table 3 show that the proposed method with the LS-SVM yielded a high accuracy over all the cases. To assess the performance of the proposed method, different EEG cases were combined and classified against group E. the proposed method achieved an average of accuracy, sensitivity, specificity and the FPR of 98%, 99%, 97% and 0.007, respectively. The results showed the efficiency of the proposed method to classify {non-ictal EEG signals vs ictal EEG signal}. The reported classification results in Table 3 showed that the ability of the proposed method to detect {individual EEG groups vs group E}, and {a combination of groups A–D vs group E}.

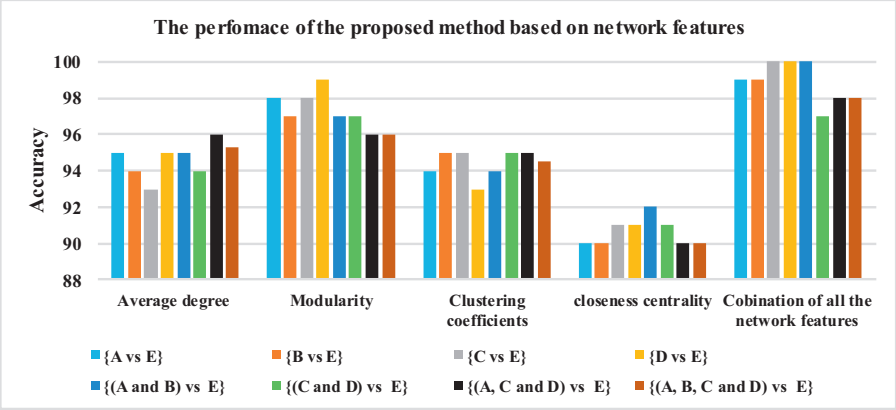


Fig. 12. Classification accuracy based on network characteristics.

Table 4
The parameters of the classifiers used during the experiments in this paper.

Classifiers	Parameters and values
LS-SVM	$(\gamma = 10,\ \sigma^2 = 1), (\gamma = 10,\ \sigma^2 = 10), (\gamma = 2,\ \sigma^2 = 1), (\gamma = 10$ *and* $\sigma^2 = 10), (\gamma = 1,\ \sigma^2 = 1), (\gamma = 2,\ \sigma^2 = 1), (\gamma = 10, \sigma^2 = 1), (\gamma = 2,\ \sigma^2 = 1)$ for the pairs, {A vs E}, {B vs E}, {C vs E}, {D vs E}, {(A and B) vs E}, {(C and D) vs E}, {(A, C and D) vs E}, and {(A, B, C and D) vs E. RBF kernel used for all the pairs
k-nearest	$k = 7$ is used, which denotes the number of the nearest neighbours
k-means	$k = 2$
Naïve Bayes	Class node denotes the EEG cases, and feature nodes represent the network characteristics

Table 5
Classification results using *k*-means.

EEG group	Accuracy	Specificity	Sensitivity	FPR
{A vs E}	100%	100%	100%	0
{B vs E}	97.6%	94.2%	95.5%	0
{C vs E}	96%	96%	97%	0.042
{D vs E}	93.7%	94.7%	94.7%	0.035
{(A and B) vs E}	96.4%	96%	93%	0.10
{(C and D) vs E}	94.5%	94%	95%	0
{(A, C and D) vs E}	95.5%	94.8	93.1%	0.052
{(A, B, C and D) vs E}	94%	93%	94%	0.172

Table 6
Classification results using NB.

EEG group	Accuracy	Specificity	Sensitivity	FPR
{A vs E}	98%	97%	98%	0.1
{B vs E}	96%	93%	92%	0
{C vs E}	94%	96%	97%	0.062
{D vs E}	92%	93%	94%	0
{(A and B) vs E}	93.7%	91.4%	92%	0
{(C and D) vs E}	92.2%	91%	90%	0.041
{(A, C and D) vs E}	91%	91%	92%	0
{(A, B, C and D) vs E}	91%	90%	91%	0.173

Table 7
Classification results using *k*-nearest.

EEG group	Accuracy	Specificity	Sensitivity	FPR
{A vs E}	100%	100%	98%	0
{B vs E}	97%	96%	95%	0
{C vs E}	95.5%	94%	93.5%	0.023
{D vs E}	93.2%	93%	92%	0
{(A and B) vs E}	93%	92%	91%	0.21
{(C and D) vs E}	92%	93%	92.5%	0.14
{(A, C and D) vs E}	90%	90%	91%	0
{(A, B, C and D) vs E}	89.5%	88%	90%	0.031

4.3. Performance evaluation based on different classifiers and 6-fold cross validation

Another three machine learning techniques were also applied to classify network's features. The *k*-means and *k*-nearest were chosen as supervised approaches while Naïve Bayes was used as unsupervised classification methods. The performances of those classifiers were assessed against the LS-SVM. The parameters of those classifiers were chosen empirically during the training session. Table 4 reports the parameters of the four classifiers. Tables 5–7 present the classification results in terms of sensitivity, accuracy, specificity and the FPR for all the EEG cases by those classifiers. The results showed the consistency in the performance of the proposed method across all the classifiers.

The results showed the consistency in the performance of the proposed method across all the classifiers. From the obtained results in Table 5, the *k*-means achieved an average accuracy of 96% across all the cases, and it is very close to those obtained by the LS-SVM in Table 3. It was the second highest classification accuracy for {A vs E}, {(A, C and D) vs E} and {(A, B, C and D) vs E}.

Table 6 reports the obtained results by the NB classifier. Based on the results the NB classifier obtained an average of accuracy, sensitivity, specificity and the FPR of 94%, 93% 93% and 0.047, respectively. They were slightly lower than those obtained by the *k*-means and the LS-SVM. The results demonstrated that the proposed method gained very similar results for all the cases with the *k*-means and the NB. When the *K*-nearest classier was used, different values of *k* were tested in the experiments and the results demonstrated that when $k = 7$ gave a high classification accuracy compared with $k = 3$ and $k = 6$. The results in Table 7 were relatively similar to those by the *k*-means, the NB and the LS-SVM although the results of {(A, C and D) vs E} and {(A, B, C and D) vs E} were lower than those by the *k*-means, the NB and the LS-SVM. By comparing the classification results in Tables 3, 5, 6 and 7 among the four classifiers, we c found that the LS-SVM was the best classifier to categorize the features of the networks. It performed better than other classifiers.

6-fold cross validation was also used to assess the performance of the proposed method in this paper. Figs. 13a–d show that there are slight fluctuations in the obtained classification results among the four classifiers. That shows the stability of the performance of the proposed method to classify all the EEG cases in each fold. We can notice that the obtained classification accuracies of the four classifiers are very similar and there is a little variation in the obtained results.

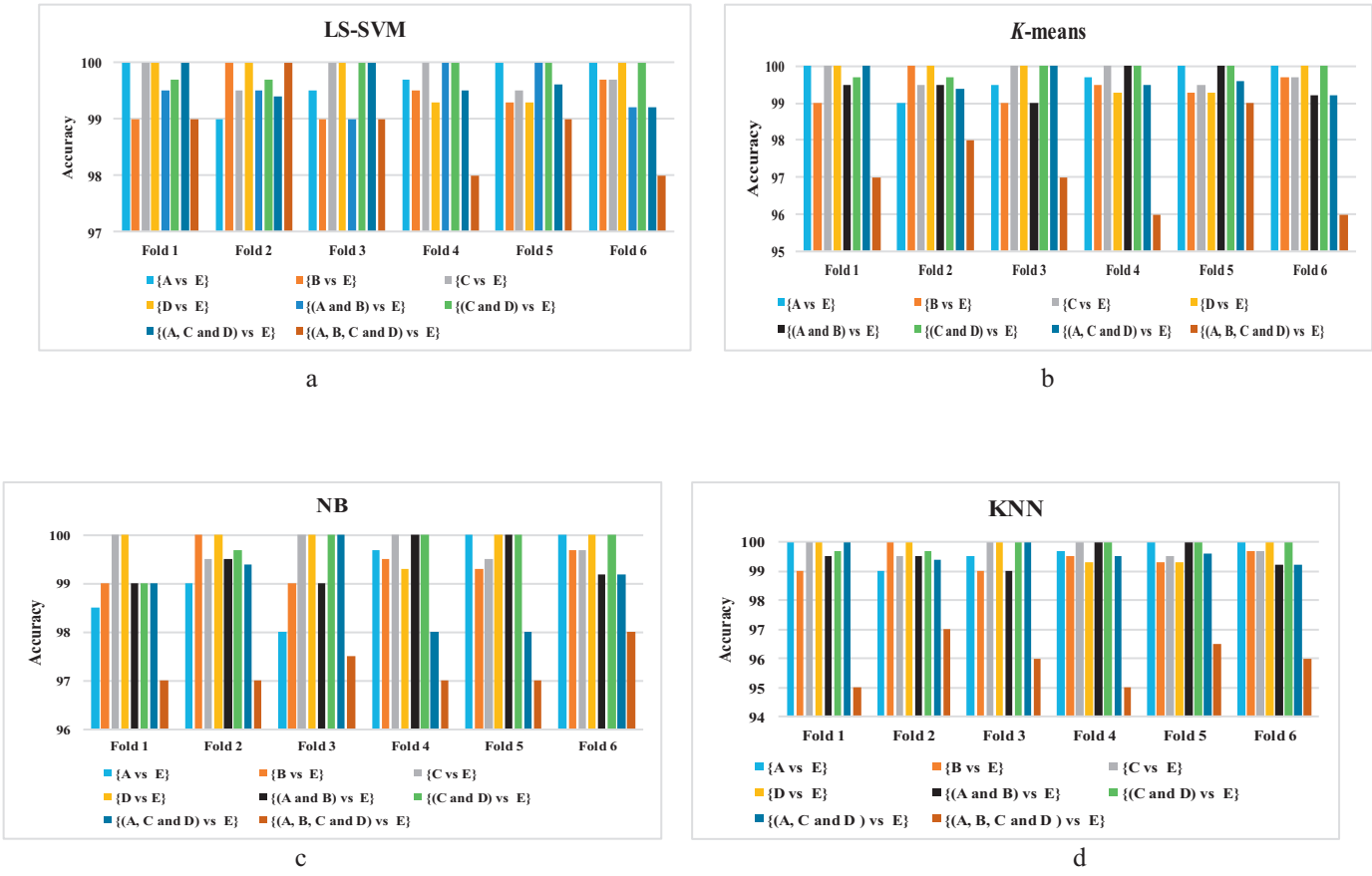


Fig. 13. Classification accuracy based on 6-fold cross validation.

5. Discussions

In this study, complex networks approach was used to identify epileptic seizure segments from non-seizures in EEG signals. The statistical features were utilized to reduce the dimensionality of the EEG signals, and to eliminate the irrelevant data. In this paper, a weighted undirected network was adopted and the threshold of the networks was empirically determined to identify the connections among nodes in the networks. Based on our findings, the following points are highlighted.

1. The modularity of the networks has high potentials to detect epileptic seizure segments from non-epileptic seizures segments. It was noticed that it reflected the sudden fluctuations in EEG signals during epileptic seizure activity and exhibited different values of modularity at each EEG group.
2. To assess the ability of the complex networks in epileptic seizures detection, the extracted statistical features in Section 3.1 were forwarded at the same time to the proposed method and directly to the LS-SVM. The obtained results demonstrated that there were big differences in the detection results when the complex networks were used to detect epileptic seizures. Fig. 14 reports the classification results for the eight groups, including {{A vs E}, {B vs E}, {C vs E}, {D vs E}, {(A, B) Vs E}, {(C, D) Vs E}, {(A, C and D) vs E}, {(A, B, C and D) vs E}} with and without using complex networks to classify the statistical features.
3. The 12 statistical features of {*median, maximum, minimum, mean, mode, range, first quartile, second quartile, standard deviation, variation, skewness, kurtosis*} were extracted and selected to represent EEG data in this study. They include linear and non-linear features. Kurtosis and skewness are non-linear features, and are often considered as high order statistics features while the other features are linear features. The features selection and ranking were based on our previous studies in Diykh et al. (2016) and Diykh and Li (2016). After the extraction, the features were used, separately, to evaluate its classification accuracy. The proposed method was repeated for 12 times and the importance of each feature was decided based on their classification accuracy results. It was found that the 12 chosen features could reflect the main characteristics of EEG signals. More details are available from our previous work in Diykh et al. (2016) and Diykh and Li (2016).
4. In general, the method has a high noise tolerance without the need of pre-processing the original EEG data. However, it would take a little longer execution time because of the mapping of the features into a complex network for each EEG segment.
5. Our finding showed that mapping an entire EEG data as complex networks without reducing the dimensionality of EEG signals based on statistical features increased the error rate and escalated the complexity of the proposed method.
6. The local network and global network features, such as clustering coefficients, average degree, and closeness centrality, were also analysed. They showed a high capacity to identify different EEG categories.
7. It showed that using a combination of the networks features gives more accurate results, and they can be used as efficient features to distinguish and analyse multi-channel EEG signals.
8. The proposed method showed its consistency in classification results when it was combined with different classifiers including supervised and non-supervised techniques.

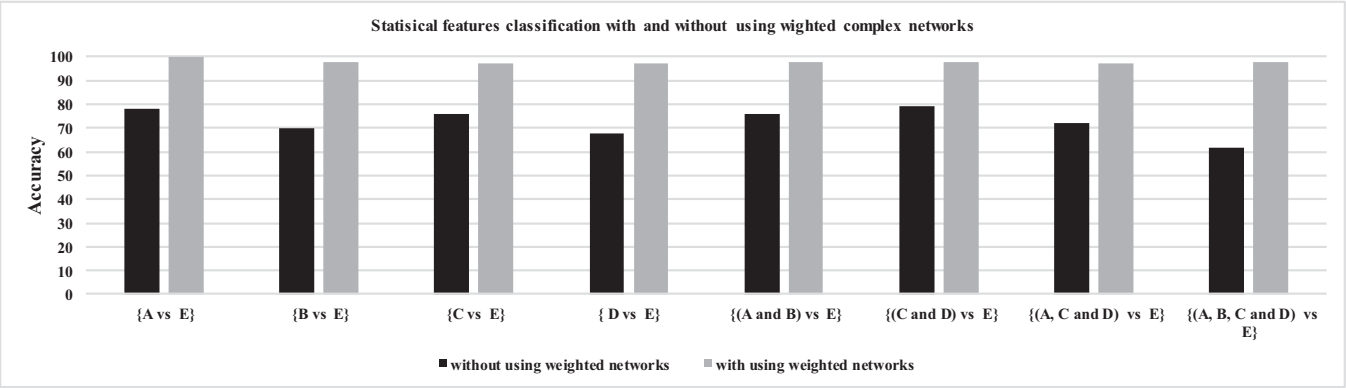


Fig. 14. Statistical features classification with and without using weighted complex networks.

9. To make the experimental data reflect the actual situation in which most of EEG data are non-ictal, new experiments were made to assess the proposed method. In the experiments, the EEG data were divided into two sets. The first set included the whole ictal EEG data while the second set was non-ictal EEG data that consisted of the first 25% of the four non-ictal sets A–D. The experiment was repeated by taking the fourth 25% of the non-ictal sets of A–D. Based on the obtained results, the proposed method showed a satisfactory performance in both experiments. The average of accuracy, sensitivity, specificity and the FPR was 97%, 96%, 97% and 0.012, respectively, by the proposed method.
10. The false positive rate (FPR) in our research was calculated over 39.3 s of EEG data. The FPR can be expended to an hourly rate of 0.0035/h (per hour). However, there were no studies that used the same datasets reported any FPRs so far. In the literature, there were several studies that used different EEG datasets reported FPRs. Aarabi, Wallois, and Grebe (2006) achieved an average FPR of 1.17/h. Zheng et al. (2012) obtained a FPR ranging from 1.6/h to 10.9/h. Williamson, Bliss, Browne, and Narayanan (2012) reported an average FPR of 0.094/h.
11. The proposed algorithm is fast and does not need the preprocessing of EEG signals. It can be used as a fully automatic seizure detection system/device in a practical clinical setting. The technique can also be adapted for predicting seizures and issuing an alert to the patient and healthcare personnel to prevent dangerous activities.

The performance of the proposed method was also compared with other existing methods of epileptic seizures detection. Based on the literature, it was found that there were no methods that used the time domain features and the complex networks characteristics in epileptic seizures detection. Furthermore, most of the existing methods of epileptic detection were tested with only five EEG categories, while in this paper eight pairs of {A vs E}, {B vs E}, {C vs E}, {D vs E}, {(A, B) vs E}, {(C, D) vs E}, {(A, C and D) vs E}, {(A, B, C and D) vs E} were discussed. Most of the developed methods of multichannel EEG signals were dependent on analysing EEG signals in time and frequency domains.

Based on the comparisons in Table 8, Siuly et al. (2011) developed an epileptic detection approach based on a sampling technique and a LS-SVM. The authors conducted their method with the same database used in this paper. In that study, an average of specificity, sensitivity and accuracy of 100%, 88% and 94% was obtained, respectively. However, our proposed method achieved a higher classification accuracy than those by Siuly et al. (2011), and a combination of eight cases was tested in this paper. Nicolaou and Georgiou (2012) employed a wavelet transform and a permutation entropy to detect epileptic seizures. For overall cases, they achieved an average of classification specificity, sensitivity and accuracy of 88.5%, 94% and 92%, respectively. That study was also conducted with the same datasets in this paper. Based on the obtained results, the proposed method outperformed Nicolaou and Georgiou (2012). Another study was made by Alam and Bhuiyan (2013) in which an empirical mode decomposition combined with a neural network was used. The authors reported only classification accuracy. The obtained results in our method were higher than those by Alam and Bhuiyan (2013).

Kumar et al. (2014) also used the same datasets to detect epileptic seizures. An approximation entropy was used as a feature to classify EEG signals. They applied a wavelet transform to analyse EEG signals. An average of specificity, sensitivity and accuracy of 92%, 94% and 93%, respectively, was obtained from classifying group D vs group E. Their results were lower than the proposed method. Tawfik, Youssef, and Kholief (2015) classified group A vs group E using a weighted permutation entropy combined with a SVM. In that study, only classification accuracy was reported. Upadhyay et al. (2016) utilized different wavelet functions to find out the appropriate wavelet function. In that study, different features were extracted and ranked based on a features selection technique. The obtained results in that study was also lower than our results. Bhattacharyya et al. (2017) applied tunable-Q wavelet to analyse EEG signals.

Multivariate fuzzy entropy was computed to identify different EEG cases. Comparing with our results, the proposed method yielded a high detection rate than Bhattacharyya et al. (2017). Patidar and Panigrahi (2017) detected epileptic seizures in EEG signals using Kraskov entropy. Their accuracy and specificity rates were also lower than that of the proposed method although our method was tested with eight combinations of A–E. Pippa et al. (2016) classified EEG signals into epileptic and non-epileptic segments based on a combination of time and frequency features. In that study, different machine learning methods were investigated. Their results were lower than our results.

The detection performance of the proposed method was also higher than those by Murugavel and Ramakrishnan (2016) and Kabir and Zhang (2016). From the results listed in Table 8, it was noticed that the proposed method achieved the highest accuracy compared with the existing methods.

6. Conclusions

In this paper, the networks characteristics and time domain features were integrated to classify ictal-EEG signals from non-ictal EEG signals. The networks attributes were investigated to identify the ability of each network feature to reflect the changes in EEG signals during epileptic seizures and seizure-free events. In this

Table 8
Comparisons among the proposed method against different epileptic seizures detection approaches with the same datasets.

Authors	Approach	Classes	Average accuracy	Average sensitivity	Average specificity
Nicolaou and Georgiou (2012)	Permutation Entropy with a SVM	A vs E, B vs E, C vs E, D vs E	88.5%	94%	93
Alam et al. (2013)	Empirical mode decomposition combined with a neural network	A vs B, (C, D) vs E	80%	–	–
		A vs E	100%	–	–
		A vs E, D vs E	100%	–	–
Siuly et al. (2011)	Simple random sampling technique combined with a least square support vector machine	D vs E	95.58%	88%	100%
Tawfik et al. (2015)	Weighted permutation entropy blended with a SVM	(A, B, C, D) vs E	93.75%	–	–
Upadhyay et al. (2016)	Wavelet transformation techniques based on features selection techniques	A vs E, B vs E, C vs E, D vs E	100%	95%	90%
Kumar et al. (2014)	Approximation entropy based wavelet transform (WT)	B vs E, C vs E, D vs E, (B,C, D) vs E, (A, B, C, D) vs E	95%	94%	92%
Patidar and Panigrahi (2017)	Entropy based Tunable-Q wavelet	A vs E	97.75%	97%	99%
Kabir and Zhang (2016)	Optimum allocation technique with logistic model tree	A vs E	95%	93%	97%
Murugavel and Ramakrishnan (2016)	Hierarchical multi-class SVM with WT	A vs E	93.63%	94%	98%
Bhattacharyya et al. (2017)	Tunable wavelet based fuzzy entropy	Focal vs non-focal	84.67%	83.86%	85.46%
Pippa et al. (2016)	Time domain and frequency domain features	A vs E	95%	94%	98%
Proposed method	weighted complex network combined with time domain features	Eight different combinations	**98%**	**97%**	**99%**

paper, the optimal networks features were identified after ranking them based on their discriminating ability. The modularity of the network was the best to reflect the sudden changes that were associated with epileptic seizures, compared with other network features. The experimental results also showed that using a combination of network features provided more accurate classification results. In the future, we will apply the proposed method to classify multi-channel sleep EEG signals. To reduce the computation time of the proposed method, less statistical features will be investigated. The proposed method can be used for developing a seizure warning system and can be adapted for assisting doctors and neurologists for better diagnosis and treatment of neurological disorders.

References

Aarabi, A., Wallois, F., & Grebe, R. (2006). Automated neonatal seizure detection: A multistage classification system through feature selection based on relevance and redundancy analysis. *Clinical Neurophysiology, 117*, 328–340.

Acharya, U. R., Yanti, R., Zheng, J. W., Krishnan, M. M. R., TAN, J. H., Martis, R. J., et al. (2013). Automated diagnosis of epilepsy using CWT, HOS and texture parameters. *International Journal of Neural Systems, 23*, 1350009.

Adeli, H., Zhou, Z., & Dadmehr, N. (2003). Analysis of EEG records in an epileptic patient using wavelet transform. *Journal of Neuroscience Methods, 123*, 69–87.

Al Ghaya, H. R., Li, Y., Abdulla, S., Diykh, M., & Wan, X. (2016). Classification of epileptic EEG signals based on simple random sampling and sequential feature selection. *Brain Informatics, 3*, 85–91.

Alam, S. S., & Bhuiyan, M. I. H. (2013). Detection of seizure and epilepsy using higher order statistics in the EMD domain. *IEEE Journal of Biomedical and Health Informatics, 17*, 312–318.

Andrzejak, R. G., Lehnertz, K., Mormann, F., Rieke, C., David, P., & Elger, C. E. (2001). 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, 64*, 061907.

Antoniou, I., & Tsompa, E. (2008). *Statistical analysis of weighted networks*. Discrete Dynamics in Nature and Society, 2008.

Bajaj, V., & Pachori, R. B. (2012). Classification of seizure and nonseizure EEG signals using empirical mode decomposition. *IEEE Transactions on Information Technology in Biomedicine, 16*, 1135–1142.

Barrat, A., Barthelemy, M., Pastor-Satorras, R., & Vespignani, A. (2004). The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America, 101*, 3747–3752.

Bashan, A., Bartsch, R. P., Kantelhardt, J. W., Havlin, S., & Ivanov, P. C. (2012). Network physiology reveals relations between network topology and physiological function. *Nature Communications, 3*, 702.

Bhardwaj, A., Tiwari, A., Krishna, R., & Varma, V. (2016). A novel genetic programming approach for epileptic seizure detection. *Computer Methods and Programs in Biomedicine, 124*, 2–18.

Bhattacharyya, A., Pachori, R. B., & Acharya, U. R. (2017). Tunable-Q wavelet transform based multivariate sub-band fuzzy entropy with application to focal EEG signal analysis. *Entropy, 19*, 99.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment, 2008*, P10008.

Bullmore, E., & Sporns, O. (2009). Complex brain networks: Graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience, 10*, 186–198.

Chandran, V., Acharya, R., & Lim, C. (2007). Higher order spectral (HOS) analysis of epileptic EEG signals. In *Engineering in medicine and biology society, 2007. EMBS 2007. 29th annual international conference of the IEEE* (pp. 6495–6498). IEEE.

Clauset, A., Newman, M. E., & Moore, C. (2004). Finding community structure in very large networks. *Physical Review E, 70*, 066111.

Das, A. B., Bhuiyan, M. I. H., & Alam, S. S. (2016). Classification of EEG signals using normal inverse Gaussian parameters in the dual-tree complex wavelet transform domain for seizure detection. *Signal, Image and Video Processing, 10*, 259–266.

Delanty, N. (2014). Complex detection, complex decisions: More detail on subclinical seizures in the acutely sick brain. *Epilepsy Currents, 14*, 129–130.

Diessen, E., Diederen, S. J., Braun, K. P., Jansen, F. E., & Stam, C. J. (2013). Functional and structural brain networks in epilepsy: What have we learned? *Epilepsia, 54*, 1855–1865.

Diykh, M., & Li, Y. (2016). Complex networks approach for EEG signal sleep stages classification. *Expert Systems with Applications, 63*, 241–248.

Diykh, M., Li, Y., & Wen, P. (2016). EEG sleep stages classification based on time domain features and structural graph similarity. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, 24*, 1159–1168.

Du, X., Dua, S., Acharya, R. U., & Chua, C. K. (2012). Classification of epilepsy using high-order spectra features and principle component analysis. *Journal of medical systems, 36*, 1731–1743.

Eberhart, R. C., Dobbins, R. W., & Webber, W. R. S. (1989). CASENET: A neural network tool for EEG waveform classification. In *Computer-Based Medical Systems, 1989. Proceedings., Second Annual IEEE Symposium on* (pp. 60–68). IEEE.

Fu, K., Qu, J., Chai, Y., & Dong, Y. (2014). Classification of seizure based on the time-frequency image of EEG signals using HHT and SVM. *Biomedical Signal Processing and Control, 13*, 15–22.

Gabor, A. J., & Seyal, M. (1992). Automated interictal EEG spike detection using artificial neural networks. *Electroencephalography and Clinical Neurophysiology, 83*, 271–280.

Gajic, D., Djurovic, Z., Gligorijevic, J., Di Gennaro, S., & Savic-Gajic, I. (2015). Detection of epileptiform activity in EEG signals based on time-frequency and non–linear analysis. *Frontiers in Computational Neuroscience, 9*, 38.

Gotman, J., Flanagan, D., Zhang, J., & Rosenblatt, B. (1997). Automatic seizure detection in the newborn: Methods and initial evaluation. *Electroencephalography and Clinical Neurophysiology, 103*, 356–362.

Guerrero-Mosquera, C., Trigueros, A. M., Franco, J. I., & Navia-Vázquez, Á. (2010). New feature extraction approach for epileptic EEG signal detection using time-frequency distributions. *Medical & Biological Engineering & Computing, 48*, 321–330.

Guler, I., & Ubeyli, E. D. (2007). Multiclass support vector machines for EEG-signals classification. *IEEE Transactions on Information Technology in Biomedicine, 11*, 117–126.

Guo, L., Rivero, D., & Pazos, A. (2010). Epileptic seizure detection using multiwavelet transform based approximate entropy and artificial neural networks. *Journal of Neuroscience Methods, 193*, 156–163.

Hassan, A. R., Siuly, S., & Zhang, Y. (2016). Epileptic seizure detection in EEG signals using tunable-Q factor wavelet transform and bootstrap aggregating. *Computer Methods and Programs in Biomedicine, 137*, 247–259.

John, G. H., & Langley, P. (1995). Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the eleventh conference on uncertainty in artificial intelligence* (pp. 338–345). Morgan Kaufmann Publishers Inc.

Kabir, E., & Zhang, Y. (2016). Epileptic seizure detection from EEG signals using logistic model trees. *Brain Informatics, 3*, 93–100.

Kelly, K., Shiau, D., Kern, R., Chien, J., Yang, M., Yandora, K., et al. (2010). Assessment of a scalp EEG-based automated seizure detection system. *Clinical Neurophysiology, 121*, 1832–1843.

Khan, Y., & Gotman, J. (2003). Wavelet based automatic seizure detection in intracerebral electroencephalogram. *Clinical Neurophysiology, 114*, 898–908.

Klatchko, A., Raviv, G., Webber, W., & Lesser, R. P. (1998). Enhancing the detection of seizures with a clustering algorithm. *Electroencephalography and Clinical Neurophysiology, 106*, 52–63.

Kumar, Y., Dewal, M., & Anand, R. (2014). Epileptic seizures detection in EEG using DWT-based ApEn and artificial neural network. *Signal, Image and Video Processing, 8*, 1323–1334.

Murali, L., Chitra, D., Manigandan, T., & Sharanya, B. (2016). An efficient adaptive filter architecture for improving the seizure detection in EEG signal. *Circuits, Systems, and Signal Processing, 35*, 2914–2931.

Murugavel, A. M., & Ramakrishnan, S. (2016). Hierarchical multi-class SVM with ELM kernel for epileptic EEG signal classification. *Medical & Biological Engineering & Computing, 54*, 149–161.

Newman, M. E. (2003). The structure and function of complex networks. *SIAM Review, 45*, 167–256.

Newman, M. E. (2004). Analysis of weighted networks. *Physical Review E, 70*, 056131.

Nguyen-Ky, Tai, Wen, P., & Li, Y. (2009). Theoretical basis for identification of different anesthetic states based on routinely recorded EEG during operation. *Computers in Biology and Medicine, 39*, 40–45.

Nicolaou, N., & Georgiou, J. (2012). Detection of epileptic electroencephalogram based on permutation entropy and support vector machines. *Expert Systems with Applications, 39*, 202–209.

Ocak, H. (2009). Automatic detection of epileptic seizures in EEG using discrete wavelet transform and approximate entropy. *Expert Systems with Applications, 36*, 2027–2036.

Opsahl, T., & Panzarasa, P. (2009). Clustering in weighted networks. *Social Networks, 31*, 155–163.

Orosco, L., Correa, A. G., Diez, P., & Laciar, E. (2016). Patient non-specific algorithm for seizures detection in scalp EEG. *Computers in Biology and Medicine, 71*, 128–134.

Özdaa, Ö., Zhu, G., Yaylali, I., & Jayakar, P. (1992). Real-time detection of EEG spikes using neural networks. In *Engineering in medicine and biology society, 1992 14th annual international conference of the IEEE: 3* (pp. 1022–1023). IEEE.

Patidar, S., & Panigrahi, T. (2017). Detection of epileptic seizure using Kraskov entropy applied on tunable-Q wavelet transform of EEG signals. *Biomedical Signal Processing and Control, 34*, 74–80.

Pippa, E., Zacharaki, E. I., Mporas, I., Tsirka, V., Richardson, M. P., Koutroumanidis, M., et al. (2016). Improving classification of epileptic and non-epileptic EEG events by feature selection. *Neurocomputing, 171*, 576–585.

Sahi, A., Lai, D., Li, Y., & Diykh, M. (2017). An efficient DDoS TCP flood attack detection and prevention system in a cloud environment. *IEEE Access, 5*, 6036–6048.

Samiee, K., Kovács, P., & Gabbouj, M. (2015). Epileptic seizure classification of eeg time-series using rational discrete short-time fourier transform. *IEEE Transactions on Biomedical Engineering, 62*, 541–552.

Sharma, M., Pachori, R. B., & Acharya, U. R. (2017). A new approach to characterize epileptic seizures using analytic time-frequency flexible wavelet transform and fractal dimension. *Pattern Recognition Letters, 94*, 172–179.

Siuly, Li, Y., & Wen, P. (2009). Classification of EEG signals using sampling techniques and least square support vector machines. In *The fourth international conference on rough sets and knowledge technology (RSKT2009), Gold Coast, Australia: 5589* (pp. 375–382).

Siuly, Li, Y., & Wen, P. (2010). Analysis and classification of EEG signals using a hybrid clustering technique. In *The proceedings of 2010 IEEE/ICME international conference on complex medical engineering* (pp. 34–39).

Siuly, Li, Y., & Wen, P. (2011). EEG signal classification based on simple random sampling technique with least square support vector machine. *International Journal of Biomedical Engineering and Technology, 7*, 390–409.

Siuly, & Li, Y. (2014). A novel statistical algorithm for multiclass EEG signal classification. *Engineering Applications of Artificial Intelligence, 34*(30 September), 154–167.

Siuly, S., & Li, Y. (2015). Designing a robust feature extraction method based on optimum allocation and principal component analysis for epileptic EEG signal classification. *Computer Methods and Programs in Biomedicine, 119*, 29–42.

Song, C., Havlin, S., & Makse, H. A. (2005). Self-similarity of complex networks. *Nature, 433*, 392–395.

Song, Y., Crowcroft, J., & Zhang, J. (2012). Epileptic EEG signal analysis and identification based on nonlinear features. In *Bioinformatics and biomedicine (BIBM), 2012 IEEE international conference on* (pp. 1–6). IEEE.

Swami, P., Gandhi, T. K., Panigrahi, B. K., Tripathi, M., & Anand, S. (2016). A novel robust diagnostic model to detect seizures in electroencephalography. *Expert Systems with Applications, 56*, 116–130.

Tang, X., Xia, L., Liao, Y., Liu, W., Peng, Y., Gao, T. , & Zeng, Y. et al. (2013). New approach to epileptic diagnosis using visibility graph of high-frequency signal. Clinical EEG and neuroscience, 1550059412464449.

Tawfik, N. S., Youssef, S. M., & Kholief, M. (2015). A hybrid automated detection of epileptic seizures in EEG records. *Computers & Electrical Engineering*.

Torres, M. E., Colominas, M. A., Schlotthauer, G., & Flandrin, P. (2011). A complete ensemble empirical mode decomposition with adaptive noise. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 4144–4147). IEEE.

Tzallas, A. T., Tsipouras, M. G., & Fotiadis, D. I. (2009). Epileptic seizure detection in EEGs using time–frequency analysis. *IEEE Transactions on Information Technology in Biomedicine, 13*, 703–710.

Upadhyay, R., Padhy, P., & Kankar, P. (2016). A comparative study of feature ranking techniques for epileptic seizure detection using wavelet transform. *Computers & Electrical Engineering, 53*, 163–176.

Wang, S., Li, Y., Wen, P., & Lai, D. (2016). Data selection in EEG signals classification. *Australasian Physical & Engineering Sciences in Medicine, 39*, 157.

Webber, W., Lesser, R. P., Richardson, R. T., & Wilson, K. (1996). An approach to seizure detection using an artificial neural network (ANN). *Electroencephalography and Clinical Neurophysiology, 98*, 250–272.

Williamson, J. R., Bliss, D. W., Browne, D. W., & Narayanan, J. T. (2012). Seizure prediction using EEG spatiotemporal correlation structure. *Epilepsy & Behavior, 25*, 230–238.

Wilson, D. R., & Martinez, T. R. (2000). Reduction techniques for instance-based learning algorithms. *Machine Learning, 38*, 257–286.

Xie, S., & Krishnan, S. (2013). Wavelet-based sparse functional linear model with applications to EEGs seizure detection and epilepsy diagnosis. *Medical & Biological Engineering & Computing, 51*, 49–60.

Zamir, Z. R. (2016). Detection of epileptic seizure in EEG signals using linear least squares preprocessing. *Computer Methods and Programs in Biomedicine, 133*, 95–109.

Zarjam, P., Mesbah, M., & Boashash, B. (2003). Detection of newborn EEG seizure using optimal features based on discrete wavelet transform. In *Acoustics, speech, and signal processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE international conference on*. IEEE Vol. 2, pp. II-265-268 vol. 262.

Zhang, J., & Small, M. (2006). Complex network from pseudoperiodic time series: Topology versus dynamics. *Physical Review Letters, 96*, 238701.

Zheng, G., Yu, L., Feng, Y., Han, Z., Chen, L., Zhang, S., et al. (2012). Seizure prediction model based on method of common spatial patterns and support vector machine. In *Information Science and Technology (ICIST), 2012 International Conference on* (pp. 29–34). IEEE.

Zhu, G., Li, Y., & Wen, P. P. (2014). Epileptic seizure detection in EEGs signals using a fast weighted horizontal visibility algorithm. *Computer Methods and Programs in Biomedicine, 115*, 64–75.

4.2 Summary of Results

Diykh et al. (2017) investigated the modularity in complex networks to detect abnormalities in EEG signals. They noticed that the modularity reflected the sudden fluctuations in EEG signals during epileptic seizure activity. Different network features were tested and evaluated. The main purpose of selecting different network characteristics was to examine the relationships between complex network features and the changes in EEG signals during epileptic seizures and non-seizure events. The proposed technique was compared with the previous work in which different transformation techniques were used. The results show that the complex networks approach outperforms other techniques in epileptic seizure detection.

Diykh et al. (2017) clearly demonstrated that using weighted complex networks based on machine learning techniques has a high potential to improve the performance of identifying abnormal behaviours in EEG signals.

5

CHAPTER 5

COMPLEX NETWORKS APPROACH FOR DEPTH OF ANAESTHESIA ASSSESSMENT

Depth of anaesthesia (DoA) monitoring during surgical operations is a very challenging task. An accurate assessment of DoA helps correctly deliver anaesthesia agents to patients and prevent unintended intraoperative awareness. After applying an anaesthesia agent to a patient, the drug takes affect in the central nervous system (CNS) where EEG signals are generated. The amplitude and frequency of EEGs change rapidly and are reflected the changes of the anaesthesia depth. This is the reason that EEG signals have been widely used as a powerful tool to capture information about anaesthesia depth.

Most of the methods developed to monitor DoA or diagnose different health issues, have been based on the analysis of EEG signals in the frequency domain, time domain or time and frequency domain. EEG signals analysis based on graph has also attracted a great deal of attention as different modern studies have shown the strength of using the networks concept in data analysis and classification. Diykh et al. (2016a, 2016b, and 2017) used complex networks to identify sleep stages and detect epileptic seizures. From the promising results of applying complex networks, and statistical approaches to analyse the stationary and non-stationary behaviors of EEG signals, Diykh et al. (2018) sought to estimate DoA based on the graph wavelet transform and the statistical model using EEG signals. Weighted complex networks were employed and spectral graph wavelet transform was performed on network nodes.

Diykh et al. (2018) show that the energies of the wavelet coefficients vary consistently

with the BIS values. As a result, a new function of DoA was designed. The proposed index was assessed using anaesthesia EEG recordings and the BIS values from 22 subjects.

5.1 Diykh et al., (2018) “COMPLEX NETWORKS APPROACH FOR DEPTH OF ANAESTHESIA ASSESSMENT”

The paper published by Diykh et al., (2016b), complex networks approach for depth of anaesthesia assessment.

Measurement 119 (2018) 178–189

Contents lists available at ScienceDirect

Measurement

journal homepage: www.elsevier.com/locate/measurement

Complex networks approach for depth of anesthesia assessment

Mohammed Diykh[a,b,*], Yan Li[a,c], Peng Wen[a], Tianning Li[a]

[a] School of Agricultural, Computational and Environmental Sciences, University of Southern Queensland, Australia
[b] Thi-Qar University, College of Education for Pure Science, Iraq
[c] School of Electrical and Electronic Engineering, Hubei University of Technology, China

ARTICLE INFO

Keywords:
Depth of anesthesia
Statistical model
Spectral graph wavelet transform
Poor signal quality

ABSTRACT

Despite numerous attempts to develop a reliable depth of anesthesia (DoA) index to avoid patients' intraoperative awareness during surgery, designing an accurate DoA index is a grand challenge in anesthesia research. In this paper, an attempt is made to design a new DoA index. We applied a statistical model and spectral graph wavelet transform (SGWT) to monitor the DoA. The de-noised electroencephalography (EEG) signals are partitioned into segments using a window technique. The window size is determined empirically, then each EEG segment is divided into sub-blocks to make the signal quasi stationary. 10 statistical characteristics are extracted from each sub-block. As a result, a vector of statistical characteristics is pulled out from each segment. Each vector of the features is then mapped as a weighted graph and spectral graph wavelet transform is performed. The total energy of wavelet coefficients at different scales is tested. The energy of wavelet coefficients at scale 3 is selected to form a $\mathbf{SGWT_{DoA}}$ function. The $\mathbf{SGWT_{DoA}}$ is evaluated using an anesthesia EEG recordings and the bispectral (BIS) from 22 subjects. The Bland-Altman, regression, Q-Q plot and Pearson correlation are used to verify the agreement between the $\mathbf{SGWT_{DoA}}$ and the BIS. The experimental results demonstrate that the $\mathbf{SGWT_{DoA}}$ has the ability to estimate the DoA accurately. The $\mathbf{SGWT_{DoA}}$ is also compared and tested with the BIS in the case of poor signal quality. Our findings show that, the $\mathbf{SGWT_{DoA}}$ can reflect the transition from unconsciousness to consciousness efficiently even for a poor signal while the BIS fails to display the DoA values on the monitor.

1. Introduction

Awareness during general anesthesia is an ongoing challenge as its consequences on patients and some juristically issues for anesthesiologists are critical [7,20,22]. Awareness happens when a patient is supposed to be anesthetized under anesthesia medications but his or her brain is active. As a result, after surgery the patient could endure severe psychological problems, such as nightmares, anxiety and depression [17,16,23].

Delivering a sufficient amount of anesthesia agents to patients helps anesthesiologists to avoid awareness during surgery, reduce costs associated with anesthesia medications consumption, maintain hemodynamic constancy and keep the recovery period short. The majority of the methods and devices that assess the DoA are based on clinical signs, such as heart rate, blood pressure and sweating could not estimate the DoA precisely, and are not reliable [9,17,26,28,29,30,31]. Using some types of anesthetic agents, such as a muscle relaxant can make the interpretation of those signs difficult [13,14]. However, clinical research and individual studies showed that there were no abnormalities in those signs for some patients who suffered awareness during general anesthesia [2,7,9,21].

Different human and animal research demonstrated that the electrical brain activities significantly correlated with the DoA during surgery. Consequently, most of the recent research have been turned their attention to developing and finding noninvasive ways of monitoring the DoA based on EEG signals [6,12,39,43,45].

Up to now, several clinical systems for monitoring the DoA were reported and developed in which the EEG signals were mainly used as an input for designing a DoA index [26,39]. One of the popular DoA index widely used for monitoring the DoA is the bispectral index (BIS). The BIS processes EEG data based on the frequency components of EEG signals [38]. The values of the BIS index are between 0 and 100, and are displayed on the BIS monitor. According to several clinical studies, up to now the BIS index reflects the patient's anesthetic states during surgery [38], which can help anesthesiologists to deliver appropriate amounts of anesthesia medications to patients [35]. However, it has received some criticisms, such as being delayed, not robust with different anesthesia medications [29], and not accurate across patients.

* Corresponding author at: School of Agricultural, Computational and Environmental Sciences, University of Southern Queensland, Australia.
E-mail addresses: Mohammed.Diykh@usq.edu.au (M. Diykh), Yan.Li@usq.edu.au (Y. Li), Peng.Wen@usq.edu.au (P. Wen), Tianning.Li@usq.edu.au (T. Li).

https://doi.org/10.1016/j.measurement.2018.01.024
Received 16 December 2016; Received in revised form 26 October 2017; Accepted 17 January 2018
Available online 02 February 2018
0263-2241/

Some alternatives to the BIS index, however, are being contemplated.

Most of the developed methods of monitoring the DoA or diagnosing different health issues were based on the analysis of EEG signals in frequency domain [14,16,17], time domain [48], or time and frequency domain [48]. EEG signals analysis based on graph domain (both terms of graphs and networks will be used interchangeably in this paper) has also been attracting a great deal of attention as different modern studies have shown the strength of using networks concept in the data analysis and classification [3,40,46,47].

Recently, a new structure for analysing signal/data through graphs, which is known as graph signal processing (GSP), has been developed [1,9,10,32,39,41,47]. The GSP is a powerful tool to analyse data in machine learning [1,6]. The most popular transformation technique in GSP is spectral graph wavelet (SGWT) which was introduced by Hammond et al. [10]. It allows the extension of the classical multiscale transform to an irregular domain [19,10,1,15]. The SGWT was used by Pham et al. [30] to recognize texture features of satellite images, by Drew et al. [5] to study the air traffic behaviour, by Smalter et al. [39] to design a chemical predicative model, and by Malek et al. [19] to classify the colour images.

We were motivated by the promising results from applying spectral graph wavelet and statistical approaches to analyse stationary and non-stationary behaviors of signals. In this paper, we sought to estimate the DoA based on the SGWT and a statistical model using EEG signals. A window technique is employed to divide de-noised EEG signals into segments. Each EEG segment is partitioned into k sub-blocks. Ten statistical characteristics are then extracted from each sub-block. A vector of the statistical features is pulled out and mapped into a weighted graph. An average energy of the wavelet coefficients and scale coefficients is obtained from each graph and used as the key characteristics for estimating the DoA. Our findings show that the energies of the wavelet coefficients vary consistently with the BIS values. As a result, a new function of the DoA ($SGWT_{DoA}$) is then designed. The proposed index is assessed using anesthetic EEG recordings and the BIS values from 22 subjects. The Bland-Altman, regression, and Pearson correlation are employed to verify the agreement between the $SGWT_{DoA}$ and the BIS. The obtained results demonstrate the ability of the new index to estimate the DoA state. The $SGWT_{DoA}$ is also compared and tested with the BIS in the case of poor signal quality. The findings from this research show that the $SGWT_{DoA}$ can reflect the transition from unconsciousness to consciousness accurately in the case of a poor signal while the BIS fails to show the value on the BIS monitor.

The remainder of the paper is arranged as follows. Section 2 presents a brief review about the current techniques of the DoA estimating. Section 3 depicts the datasets used in this paper. Section 4 reviews the key concept of spectral graph wavelet. Section 5 describes the proposed method. Section 6 presents the experimental results and simulations. Section 7 shows a case study of evaluating a poor signal quality signal case by the proposed method. Finally the conclusions of the study are drawn in Section 8.

2. Related work

Various techniques have been reported to estimate the DoA based on EEG recordings. The leading task of those methods was to design an accurate index to monitor the DoA. In this section, we review some of recent methods in which DoA indexes were developed.

Nguyen-Ky et al. [24] proposed a method to assess the DoA based on a Bayesian method. The maximum posterior probability (MPP) was used to examine the distribution of the EEG signal. A DoA function was designed using the MPP. That study was reported that the values of the MPP changed correspondingly with the change of the anesthetic depth. Tupaika et al. [44] applied a symbolic dynamics analysis method to monitor the DoA. An EEG signal was filtered and then separated into segments. Each segment was transferred into a sequence of three symbols with an overlapping of two symbols. As a result 64 different patterns of symbols were obtained. The Shannon entropy and Rényi entropy of those patterns were calculated and used to identify the different anesthetic states from EEG signals.

Jospin et al. [12] utilized detrended fluctuation analysis (DFA) to examine the EEG fluctuations for monitoring the anesthetic depth. 17 subjects were involved in that study. The behaviors of four slopes of the DFA were used as the indexes, and tested in two different EEG segment lengths: 30 s and 15 s. Palendeng et al. [28] suggested a DoA index based on a stationary wavelet transform (SWT). The EEG signals were analysed second by second using a window technique. The amplitude and instantaneous phases were obtained from every EEG segment and recursively computed through the whole EEG signal after applying the SWT.

Mousavi et al. [20] utilized a complex wavelet transformation to estimate the DoA. The behaviors of EEG signals in Alpha and Beta bands were studied. Each of those bands was also decomposed into 5 sub-bands and the modulation of signal (MS) was then calculated separately. The entropy of the MS was obtained and used as the key value to derive a DoA index. Zoughi et al. [49] considered the entropy of wavelet coefficients as a feature to estimate the DoA. An EEG signal was decomposed into different levels using discrete wavelet transform. The entropy at each scale was calculated and used to design a DoA index. 22 subjects were involved in that study. Liu et al. [18] traced the changes of the DoA using a multiscale entropy (MSE). 25 subjects were involved in that study. The MSE was used to analyse the complexity of EEG signals. The MSE and independent entropy combined with an artificial neural network were used together in designing a DoA index.

Kalinichenko et al. [13] made an attempt to monitor the DoA based on frequency domain, and a single channel EEG signal was used. A combination of several indexes was designed and tested. The approximate entropy, power spectrum density and signal randomness analysis were utilized in the DoA derivation and a statistical approach was employed to compare those methods. Shalbaf et al. [35] applied a permutation entropy and a frequency measure called Delta-Index for estimating the DoA. Those features were pulled out from each EEG segment to formulate a DoA index. The designed index was tested using a combination of statistical measures.

Nguyen-Ky et al. [25] also estimated the DoA based on a wavelet transform. The EEG signal was decomposed into different levels to extract the desired frequencies. As a result, six bands: θ-band, α-band, β-band, δ-band, γ-band and Electromyography (EMG), were extracted, and then an eigenvector of wavelet coefficients was considered. An index was designed based on the statistical characteristics of the extracted features and was then compared with the BIS index. Although, the existing studies achieved promising results compared with the BIS monitor, there are a surge of needs to improve the existing methods, especially in terms of accuracy and complexity time. This paper is to assess the depth of anesthesia (DoA) using graph domain and statistical features. More details about the proposed method is presented in Section 5.

3. Experimental data

The data used in this paper were collected from 37 adult subjects. The ethics approval was obtained from the University of Southern Queensland Human Research Ethics Committee (No: H09REA029) and the Toowoomba and Darling Downs Health Service District Human Research Ethics Committee (No: TDDHSD HREC 2009/016).

The demographics information of all the participants who involved in this study are explained in Table 1. Four adhesive forehead Quatro electrodes were applied to each subject. The EEG data were recorded through those electrodes. For the off line analysis, the captured data were transferred to a personal computer. The exported EEG data file contained the real time log, EEG data, the BIS index and the monitor error logs (critical events and any monitor errors). The EEG data were sampled at frequency of 128 Hz. The data were converted from ASCII

Table 1
Demographic information of Participants.

Age (yrs.)	Weight (kg)	Height (cm)	Gender (M/F)	Midazolam (mg)	Alfentanil (mg)	Propofo l (mg)	Parecoxib (mg)	Fentany1 (μg)
22–35	55–130	154–194	15/22	2–5	500, 750, 1000	90–200	40	100, 150

format to a signed number form using Matlab code. The single EEG channel (channel 1) data was used in this study.

3.1. Sample selection and signal de-noising

EEG signals can be contaminated by different types of noise, such as environmental and physiological noise. The environmental noise is produced by power line interference from the devices used in a surgical room while the physiological noise is generated by various noise sources, for example cardiac signals and movement artefacts. To eliminate those noises, a nonlocal mean method was utilized in this paper to de-noise the EEG signals. The details of de-noising EEG signals are in Li et al. [16]. For an accurate assessment of the proposed method, 22 subjects were used from the data collected. Four of those subjects that its signal quality indicator was lower than 15 were used to assess the ability of the proposed DoA index to estimate the DoA in the case of poor signal quality. The ID Nos. of the subjects are 14, 12, 17 and 32. While the ID Nos. of the remaining 18 subjects that whose signal quality was > 15 are 2–8, 11, 13, 18, 19, 20, 21, 22, 24, 25, 29 and 30. Fig. 1a and 1b show the example from subject 7 with high signal quality and subject 14 with low signal quality.

4. Review of wavelet transform and spectral graph transform

4.1. Classical wavelet transform

Wavelet transforms have been extensively utilized to tackle different problems in biomedical signals processing due to their ability to analyse the non-stationary behaviour of signals [6,15,27,24,36]. A wavelet function ψ (mother wavelet) at scale t and location k with continuous wavelet transform (CWT), can be designed as

$$\psi_{t,k}(x) = \frac{1}{t}\psi\left(\frac{x-k}{t}\right) \tag{1}$$

The wavelet coefficient at scale t and location k for a signal x is defined as

$$W_f(t,k) = \int_{-\infty}^{\infty} \psi^*_{t,k}(x) f(x) dx \tag{2}$$

According to Hammond et al. [11], Eq. (1) cannot be defined directly on graph nodes. This obstacle can be solved by performing a classical wavelet transform, $(x) = \frac{1}{t}\psi\left(\frac{x-k}{t}\right)$. It can be defined in the Fourier domain as

$$\psi_{t,k}(x) = \frac{1}{2\pi}\sum_{-\infty}^{\infty} \hat{\psi}(sw) e^{-jwa} e^{jwx} dw \tag{3}$$

where $\wedge$ symbol indicates Fourier domain.

The complex exponential defined in Eq. (3), e^{jwx}, is the eigenfunction of the one dimension Laplacian i.e. $\frac{d^2}{dx^2}e^{jxw} = -w^2 e^{jwx}$ and shifting the wavelet to location a, equivalent to a multiplication by e^{-jwa}.

4.2. Weighted graph and spectral graph concept

Consider a weighted graph $\mathbf{G} = \{\mathbf{E}, \mathbf{V}\}$ that contains of a set of nodes $\mathbf{V}$, a set of edges $\mathbf{E}$, and a weight function w: $\mathbf{E} \rightarrow \mathbb{R}^+$ which gives a positive weight to each edge. In this paper we considered only an undirected graph $w(i,j) = w(j,i)$ and $w(i,i) = 0$, and a finite graph where, $[\mathbf{V}] = N < \infty$, where N equals to the number of nodes. The adjacency matrix (**A**) of a weighted graph **G** is an $N \times N$ matrix with node $a_{i,j}$. It is defined as:

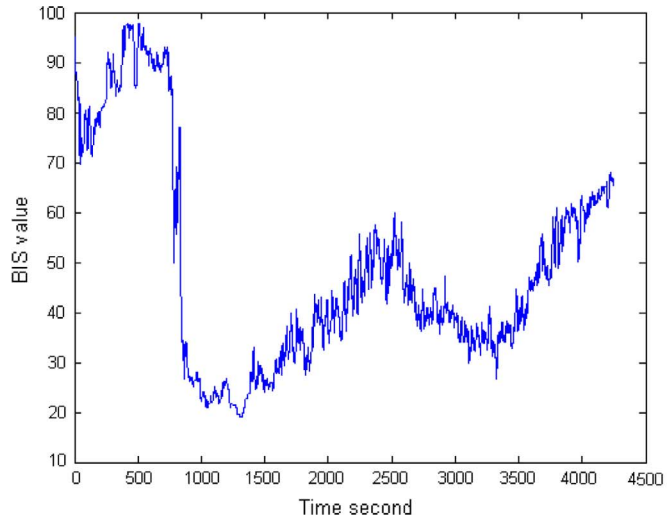
$$a_{i,j} = \begin{cases} w(e) & if\ e \in E \\ 0 & otherwise \end{cases} \tag{4}$$

The degree of node i in a weighted graph is defined as a sum of the weights of all the edges connected to it. It is defined as:

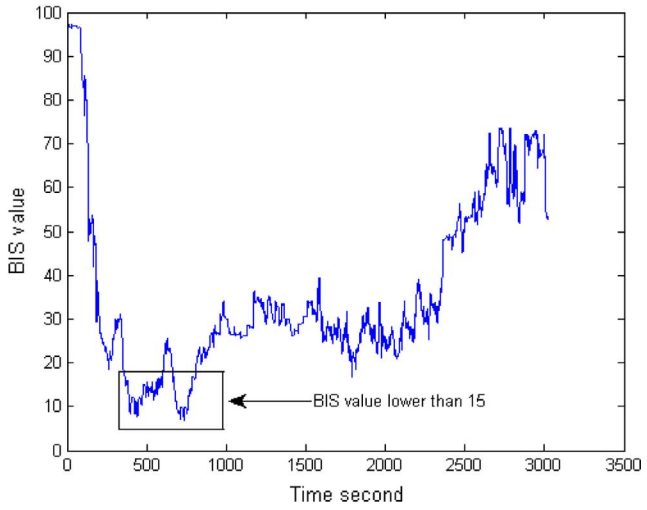
$$d(i) = \sum_j w_{i,j} \tag{5}$$

where $d(i)$ is the degree of node i, and $w_{i,j}$ is the weight of edge between nodes i,j.

The degree of all the nodes in graph **G** is represented by matrix **D**, which includes diagonal elements equalling to the node degree, and



a. Example of good signal quality indicator (Subject 7)



b. Example of poor signal quality (Subject 14)

Fig. 1. Example of different cases of signal quality indicators.

zeros elsewhere. Graph Laplacian L is:

$$L = \mathbf{D} - \mathbf{A} \tag{6}$$

where **D** is a diagonal matrix and **A** is an adjacency matrix of graph **G**. The normalized Laplacian $\mathscr{L}$ matrix is defined as

$$\begin{aligned}\mathscr{L} &= D^{-\frac{1}{2}} L D^{-\frac{1}{2}} \\ &= I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}\end{aligned} \tag{7}$$

The elements of $\mathscr{L}$ can be written as

$$\mathscr{L}_{ij} = \begin{cases} 1 & i = j \\ -\frac{1}{\sqrt{\delta(v_i)\delta(v_j)}} & i \neq j \end{cases} \tag{8}$$

Both versions of $\mathscr{L}$ and L Laplacians are symmetric and positive. They can be decomposed into non-negative eigenvalues and eigenvectors. The spectral graph transform is performed based on eigen-decomposition of L *or* $\mathscr{L}$. For L, the decomposition is:

$$L = X \wedge X^T \tag{9}$$

where $X = |\chi_1|\chi_2\}...|\chi_n|$ is an orthonormal matrix that includes N eigenvectors and $\wedge$ is a diagonal matrix of eigenvalue $|\lambda_1 = 0 \leqslant \lambda_2 \leqslant \lambda_3 ... \leqslant \lambda_N|$.

4.3. Spectral graph wavelet transform

As mentioned before, many interesting complicated data sets are located in domains in which classical wavelets are not suitable. Some transformation techniques that work on the nodes of weighted graphs, such as spectral graph Fourier transform (SGFT) and spectral graph wavelet transform (SGWT) have been developed [4,33,37]. The SGWT was developed to tackle the problem in the SGFT [10]. The SGWT was developed to define a scale wavelet basis $\psi_{t,k}$ on a graph. The spectral graph wavelet $\psi_{t,n}(m)$ at scale t and node n is defined as:

$$\psi_{t,n}(m) = \sum_{i=1}^{N-1} g(t_j\lambda_i) x_i^*(n) x_i(m) \tag{10}$$

where g is a kernel function and should behave as a bandpass filter $(0) = 0, \lim x \to \infty g(o) = 0$, and λ_n is an eigenvector of a Laplacian matrix. The spectral graph wavelet coefficients of a signal $f \in \mathbb{R}^N$ which represents the projection of f onto the orthonormal Laplacian eigenvector space is defined as:

$$W_f(t,n) = < \psi_{t,n} f \geqslant \sum_{m=0}^{N-1} \psi_{t,n}^*(m) f(m) \tag{11}$$

To compare Eq. (3) with Eq. (10), we can notice that

- e^{jwx} in (3) has been replaced by eigenvector $x_i^*(m)$ of Laplacian matrix.
- The role of frequency ω is played by eigenvalue λ_i
- Shifting the wavelet to node m is similar to a multiplication by $x_n^*(k)$, replacing e^{-iwa}.

Similar to the low-pass function in the classical wavelet transform, a low pass function h is defined to capture those residual low pass components of graph **G,** which is close to zero. Function h should satisfy the following conditions: $h(O) > 0$, and $h(x) \to as x \to \infty$

The scaling function coefficients of signal f should satisfy some conditions in [5]. Corresponding to the classical wavelet transform, the SGWT scaling function coefficients of signal $f \in \mathbb{R}^N$, at graph node n, location k and time scale t, is defined as follows:

$$S_f(n) = < \phi_n f \geqslant \sum_{m=0}^{N} \phi_n^*(m) f(m) \tag{12}$$

where the scaling function $\phi_n(m)$ was defined as:

$$\phi_n(m) = \sum_{i=0}^{N-1} h(t_j\lambda_i) x_i^*(n) x_i(m) \tag{13}$$

where signal $f \in \mathbb{R}^N$ and t belongs to a set of scales $\{t_i\}_{i=1,...n}$. The SGWT results in a set of graph scaling function coefficients and a set of graph wavelet coefficients. In this work, graph scaling function coefficients and graph wavelet coefficients are used as the key features to assess the DoA.

Applying the SGWT with large graphs may not be effective due to requiring a complex computation for the entire set of eigenvectors and eigenvalues. Taking into account the huge number of nodes in every graph, it could be impossible to apply such transform to analysis signals such as EEG and EOG, image processing [4]. Hammed et al. [10] tackled this issue by developing a fast transform (Chebyshev polynomoial approximation algorithm) to avoid the need for computing complete spectrums of $\mathscr{L}$ or L. In this paper we choose the cubic kernel to simplify the calculations, and the SGWT was implemented by using fast Polynomial approximation suggested in [10].

5. Methodology

The proposed method aims at estimating the DoA based on a statistical model and in spectral graph domain. Fig. 2 describes the proposed method. Each part of the block diagram is explained in detail in the next section. The original EEG signals were de-noised using a nonlocal means method. All the details of the de-noising method is available in [16].

The de-noised EEG signal is partitioned into small segments using a window segmentation technique. The window size in this paper was 56 s with an overlapping 55 s. Each EEG segment was divided into a number of blocks. The number of blocks (k) was empirically determined. Ten statistical features were extracted from each block to reduce the dimensionality of EEG data. As a result, a set of statistical characteristics was pulled out from each EEG segment and then transferred into a weighted graph. The SGWT was applied to the weighted graph and the wavelet coefficients and scaling function coefficients are extracted at each scale. The energy of the wavelet coefficients at scale 3 was employed to trace the changes of the DoA. An index of the DoA was

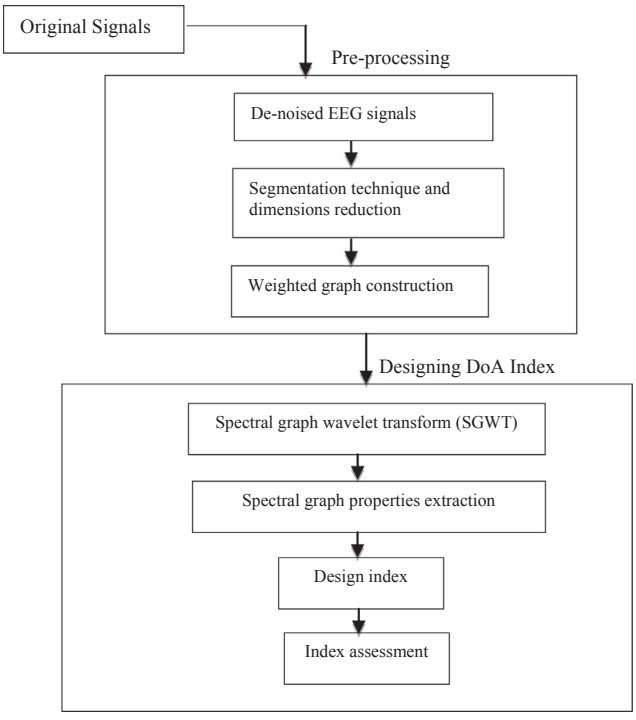


Fig. 2. Block diagram of the proposed method to estimate the DoA.

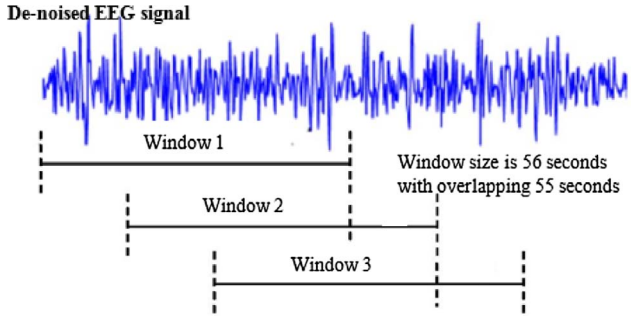


Fig. 3. Example of partitioning an EEG signal into segments using a window segmentation technique.

designed to identity the different anesthetic states from EEG signals. Different statistical methods are utilized to evaluate and assess the proposed index against the BIS index.

5.1. EEG dimensions reduction and stratification

In signal processing, most of the developed methods were designed for analysing stationary signals while the EEG signals are nonstationary [42]. That means the statistical properties of EEG signals change with time. One efficient approach to analyse the non-stationary time series such as EEG signals is to view it as consisting of many small segments that are themselves stationary. In this paper the EEG signals were divided into segments based on a window technique. The size of the window was 56 s with an overlapping 55 s. Based on our previous studies, it was found that using a sliding window to segment EEG signals with one second of new anesthetic EEG data points and an overlap of 55 s' previous data provided satisfactory DoA assessment results. The window segmenting with overlapping data points can make the DoA index more smooth and accurate over time. Fig. 3 shows an EEG signal being partitioned into segments. Then, each EEG segment was partitioned into k smaller groups called blocks. The number of blocks were determined in this work based on simulation and experimental results. Ten statistical characteristics were extracted from each block and put into one vector to represent the EEG data. As a result, a vector of $10 \times k$ statistical characteristics was pulled out from each EEG segment, where 10 represents the number of features and k refers to the number of blocks.

Fig. 4 shows an example of an EEG segment is partitioned into blocks and the statistical features are extracted. The ten features used in this work are *{median, maximum, minimum, mean, mode, range, first quartile, second quartile, standard deviation, variation}*. The purpose of using these ten statistical features was to reduce the dimensionality of the EEG data, and also it was found that some of EEG data are symmetric distribution and other skewed distribution. Some of these features were used as appropriate features for EEG data with symmetric distribution while others were used for skewed distribution.

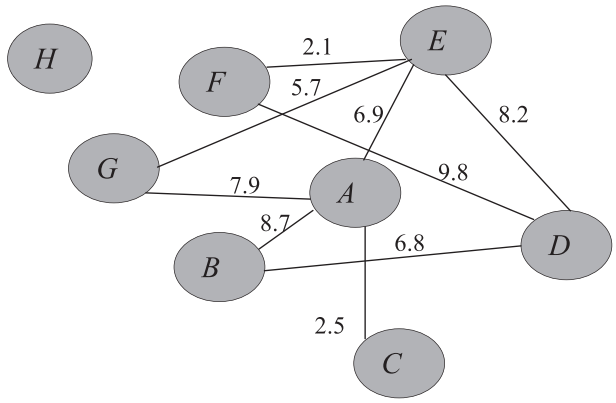


Fig. 5. Example of a time series converted to a weighted graph.

5.2. Transformation of statistical features to weighted graph

The obtained vector of the statistical features in Section 5.1 was mapped as a weighted graph to connect all the extracted features in one network. The graphs were structured according to our previous work [3], and Zhang and Small [46].

Each pair of nodes were connected and the weight of their edge was computed based on the nodes' similarity. Let {xij} = 1, 2, 3… N be a set of time series of N data points. Each data point of the series was assigned to be a node in a weighted graph. Two nodes, v_i and v_j, in a graph are connected if the distance between the two nodes is less than or equal to an adaptive threshold.

$$(v_i,v_j) \in E, \; if \;\; d(v_i,v_j) \leqslant \delta \tag{14}$$

w here δ is an adaptive threshold, Fig. 5 shows a time series: *{A = 21.3, B = 9.5, C = 8.7, D = 4.7, E = 10.2, F = 14.9, G = 7.9, H = 5.9}* being transferred into a weighted graph. The adjacency matrix, **A,** of graph **G** and the Laplacian matrix (Section 4.3) are calculated for all **V** to describe the connection of the graph nodes. The adjacent matrix and Laplacian matrix of an undirected graph are symmetric, i.e. A (v_i,v_j) = A (v_j,v_i). All the graphs constructed with a fixed number of nodes in this paper based on the assumption that the number of the nodes increases when there are more sub-segments in each EEG segment, and vice versa. From Fig. 5, we can notice that node H has no connections with other nodes in the graph. This means that node H is an isolated point in the graph.

In this work, the low-pass and band-pass filters mentioned in Section 4.3 were used to identify each eigenvector. The spectral graph scaling function coefficients and spectral graph wavelet coefficients were calculated with different scales and used as the key features to estimate the DoA. The Chebyshev polynomial approximation described in [10,42] was used in this work to avoid performing a full eigen-decomposition.

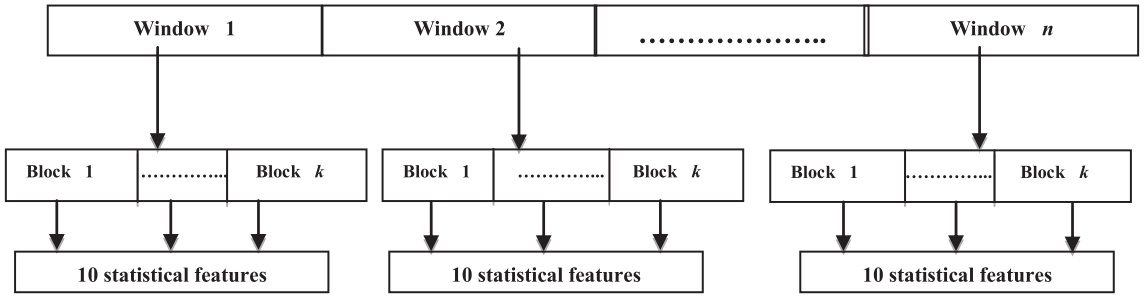


Fig. 4. An example of extracting statistical features from an EEG segment.

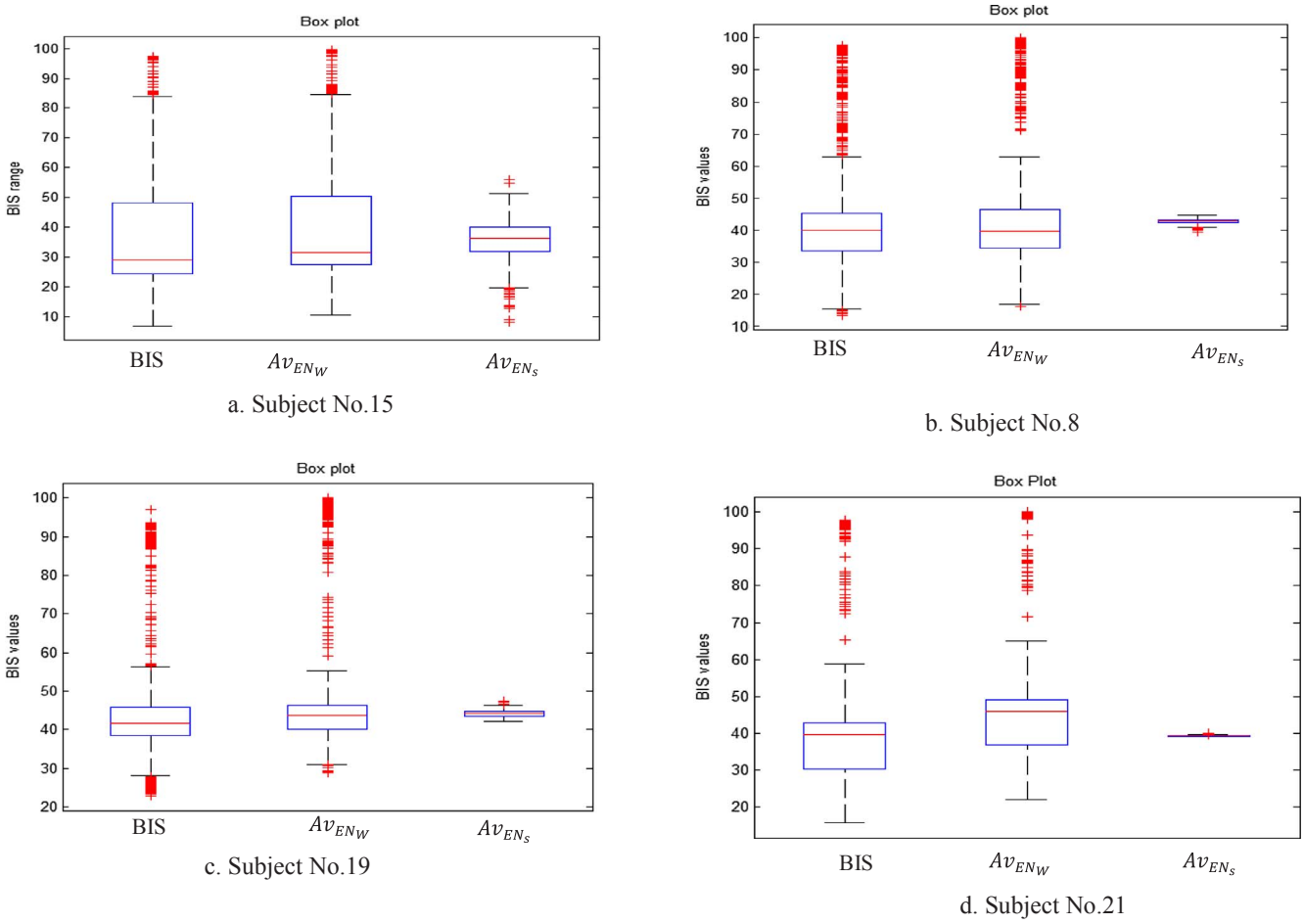


Fig. 6. Comparison among BIS, Av_{EN_s} and Av_{EN_W}.

5.3. Spectral graph characteristics extraction and designing a new DoA index

Applying the SGWT to EEG signals is effective to capture the desired frequencies by decomposing the signals into low-frequency (scaling coefficients) and high-frequency components (wavelet coefficients). The wavelet and scale function coefficients in Eqs. (11) and (12) at scale t were calculated. Then, the total energy of the wavelet and scaling function coefficients were chosen as the key characteristics for each graph. It was found that the energy of the wavelet coefficients across 22 subjects was increased during the awake state, when the whole brain is active, while it was degraded during the anesthetic states. Consequently, the total graph energy of the wavelet coefficients was chosen to trace the DoA in this research. During the experiments, it was noticed that the proposed DoA index matched the BIS index variations.

The number of the scale levels was chosen based on the spectral content of a signal. It was found that if the scale level was higher than five (roughly corresponding to the principal EEG rhythms), the wavelet coefficients did not provide significant information about the EEG signals [32,34]. As the result, the number of the scales from one to four was investigated to test the performance of the proposed method for estimating the DoA. It was found that $t = 3$ yielded better results than those of other numbers.

At each scale the energy of the wavelet coefficients and scale coefficients for each graph were calculated as follows:

- Calculate the energy of the wavelet coefficients and scale coefficients at each scale.

Let S_{it} and W_{it} be the ith wavelet coefficients and scale coefficients at scale t, respectively. The total energy of the wavelet coefficients (EN_g) and the scale coefficients (EN_s) at scale t were calculated by the following equations.

$$EN_w = \sum_{i=1}^{n} (W_{it})^2 \tag{15}$$

$$EN_s = \sum_{i=1}^{n} (S_{it})^2 \tag{16}$$

- Calculate the average energy of the wavelet and scale coefficients by dividing $EN_w and EN_s$ on the number of the wavelet and scale coefficients in the corresponding scale:

$$Av_{EN_W} = \frac{EN_w}{n} \tag{17}$$

$$Av_{EN_s} = \frac{EN_s}{n} \tag{18}$$

After calculating the total energy of the wavelet and scaling coefficients, those two features were tested against the BIS index. We observed that the average energy of the wavelet coefficients (Av_{EN_W}) showed a higher positive relationship with the BIS index values than the energy of the scaling coefficients (Av_{EN_s}). To show the distribution of those two features against the BIS index, the box plot was used to compare the distribution of the extracted features from 22 subjects against the BIS index values. Fig. 6 shows that the energy of the wavelet and scaling coefficients from four subjects against the BIS values. The IDs of the four subjects were 19, 15, 21, and 8. From Fig. 6, it was

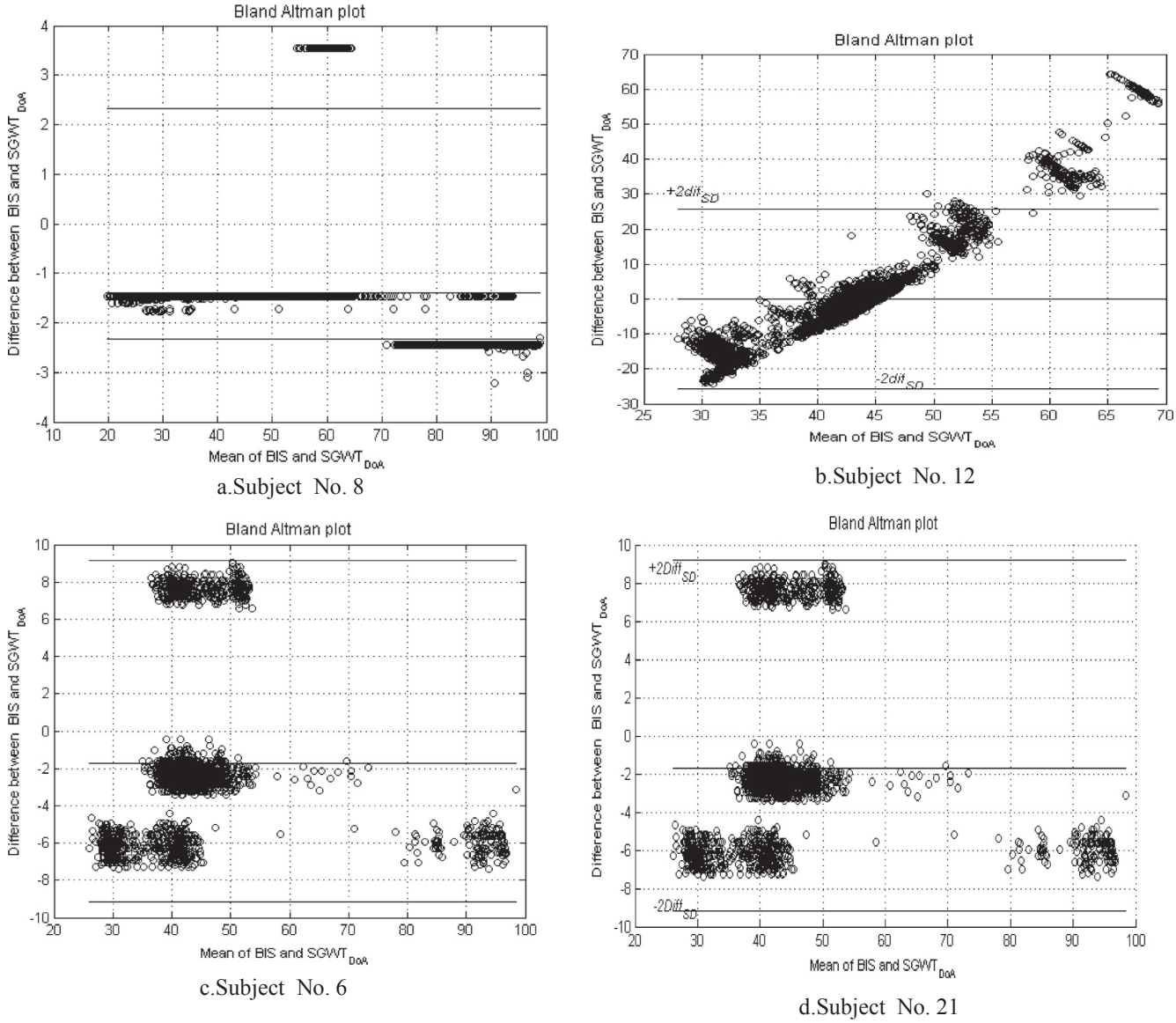


Fig. 7. Bland Altman.

noticed that Av_{EN_W} values changed in accordance with the BIS values across the four subjects. Based on the relationships between Av_{EN_W} and the anesthetic states, a new index of the DoA was designed as follows:

$$\mathrm{SGWT}_{DoA} = Av_{EN_W} + \rho \tag{19}$$

where ρ is an offset value, and Av_{EN_W} is the average energy of the wavelet coefficients at scale 3. For all the subjects, it was found that the SGWT_{DoA} values varied as the way the BIS changed.

The $SGWT_{DoA}$ index was normalized in order to keep the values of the $SGWT_{DoA}$ between 0 and 100. SGWT_{DoA} was normalized based on the following formula.

$$\begin{aligned} &if\ \mathrm{SGWT_{DoA}} > 100 \rightarrow \mathrm{SGWT_{DoA}} = 100 \\ &else\ if\ \mathrm{SGWT}_{DoA} < 0 \rightarrow \mathrm{SGWT_{DoA}} = 0 \end{aligned} \tag{20}$$

6. Experimental results

To evaluate the proposed DoA index, a set of experiments and simulations were designed using the data from 22 subjects whose demographics information are described in Section 3. As mentioned before, different statistical methods were used to evaluate the proposed index against the BIS index. The results were discussed in the next section. According to the experimental results, the SGWT_{DoA} can reflect the transition from unconsciousness to consciousness correctly.

The experiments were conducted using Matlab spectral graph wavelet transform toolbox[1] and Matlab tool for network analysis[2] on a computer with 3.40 GHz Intel(R) core(TM) i7 CPU processor machine, and 8.00 GB RAM.

6.1. The agreement of the $SWGT_{DoA}$ and the BIS index using Bland-Altman

Evaluating the agreement between two methods can be achieved using statistical methods to assess the degree of the similarity between the two approaches. In this work, the Bland-Altman method was used to test the agreement between the $\mathrm{SGWT_{DoA}}$ and the BIS index based on the difference between them [8] using the same EEG signals. The difference is defined as Dif = ($\mathrm{SGWT_{DoA}}$ -BIS). The mean difference (called estimate bias) and the standard deviation (which describes the fluctuation around the bias) of Dif are calculated, where Dif_{mean} = mean

[1] http://wiki.epfl.ch/sgwt.1.
[2] http://strategic.mit.edu/downloads.php?page = matlab_networks.

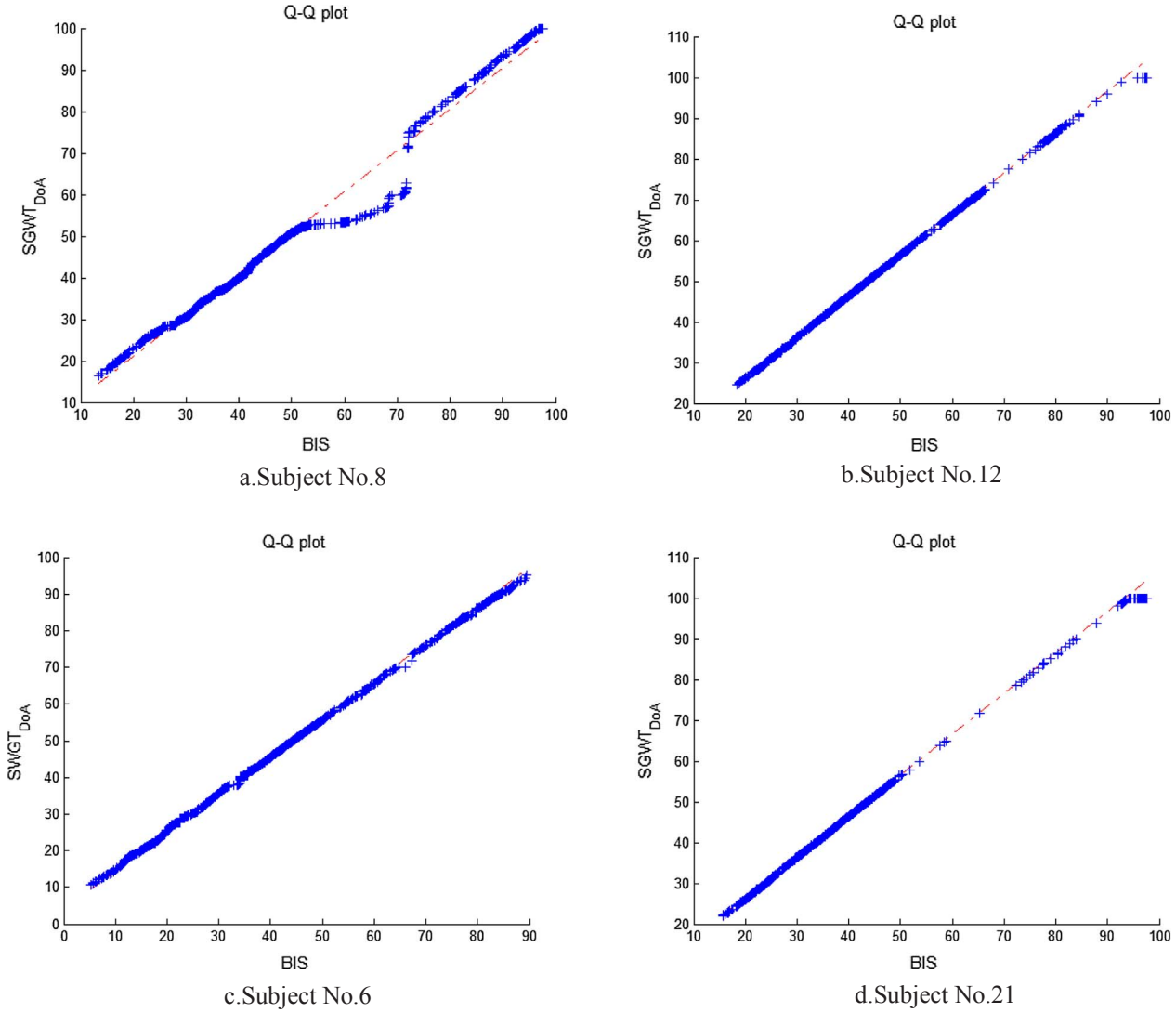


Fig. 8. Q-Q plot.

(Dif) and Dif_{SD} = std(Dif).

The Bland-Altman method recommended that 95% of the data points should lie in between ± $2Dif_{SD}$ of the main difference. Fig. 7 shows the mean of the differences between the $SGWT_{DoA}$ and the BIS for the four subjects (IDs of 8, 6, 12 and 21). It was noticed that most of the data points lied in between the upper limit (+ $2Dif_{SD}$) and lower limit ($-2Dif_{SD}$). The obtained results showed the good agreement of the two indexes.

6.2. Quantile-Quantile plot

The quantile–quantile plot (Q-Q) is a graphical method that is used to determine the validity of two methods depending on whether they have a common distribution, similar distribution shape and similar tail behaviour.

For further testing, the Q-Q plot was used to plot the quantiles of the $SGWT_{DoA}$ against the quantiles of the BIS index. The quantiles mean that the fraction (or percent) of points lie below a given value. The Q-Q plot suggests that 0.3 of the points which represent 30% of the data should fall below and the rest points should lie above that value. A 45-degree reference line is also plotted. If the $SGWT_{DoA}$ and the BIS index have the same distribution, the points should fall approximately along this reference line. Fig. 8 shows the Q-Q plot of the $SGWT_{DoA}$ and the BIS index for the four subjects.

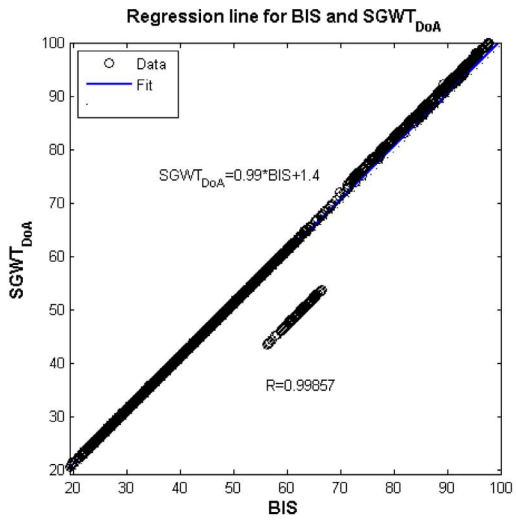
From the obtained results in Fig. 8, we can notice that the $SGWT_{DoA}$ and the BIS have a similar distribution and the behaviour of the $SGWT_{DoA}$ is close to the BIS index.

6.3. Regression technique

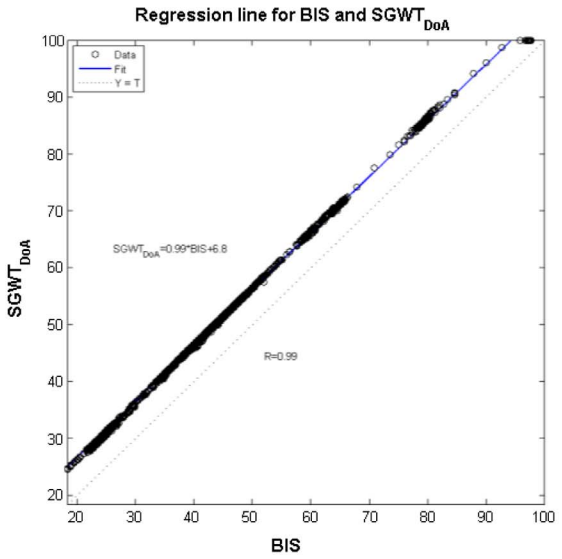
Regression is a statistic measure which determines the degree of a relationship between an independent variable and a dependent variable. It is also called coefficient of determination. It ranges from 0 to 1. If the coefficient of determination value is close to 1, that means that there is a strong relationship between the $SGWT_{DoA}$ and the BIS index, and vice versa.

In this work, the regression method was also used to evaluate the degree of similarity between the BIS and the $SGWT_{DoA}$. Here, the regression was run for all the 22 subjects and the coefficient determination values were calculated.

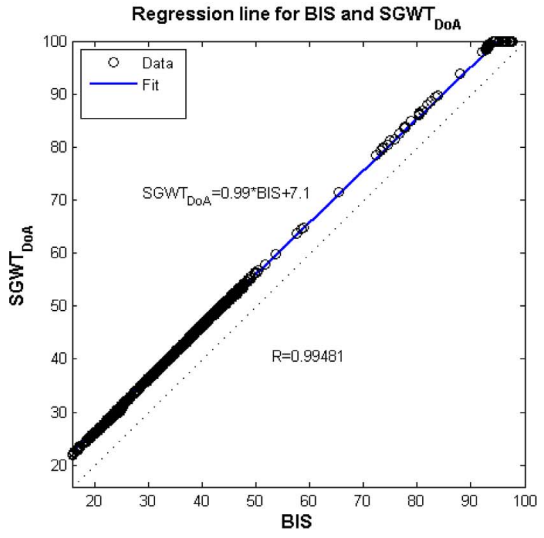
Fig. 9 shows the regression line of the BIS and the $SGWT_{DoA}$ for subjects of Nos: 8, 6, 12 and 21. From the obtained results, it is clear that the $SGWT_{DoA}$ and the BIS index are associated with a high agreement. The average of the coefficient determination for the 22 subjects was 0.95681. Fig. 10 shows the coefficient determination values. From the obtained results in Figs. 9 and 10, we can notice that the $SGWT_{DoA}$ function and the BIS have a similar distribution and the behaviour of the $SGWT_{DoA}$ is very close to the BIS index.



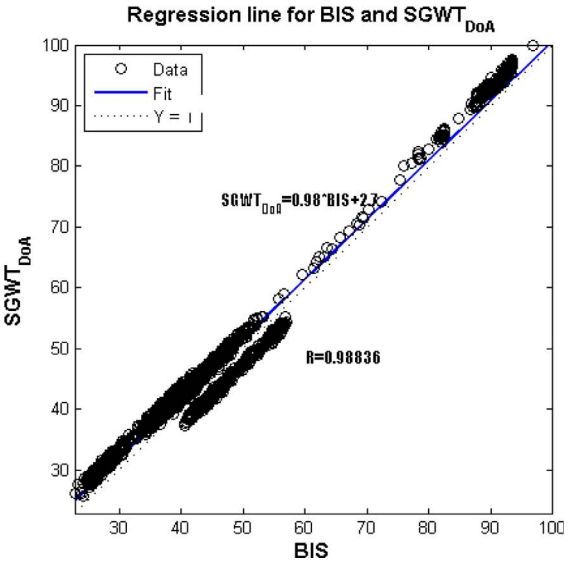
a.Subject No.8



b.Subject No.12



c.Subject No.6



d.Subject No.21

Fig. 9. Regression line for the BIS index and SGWT$_{DoA}$.

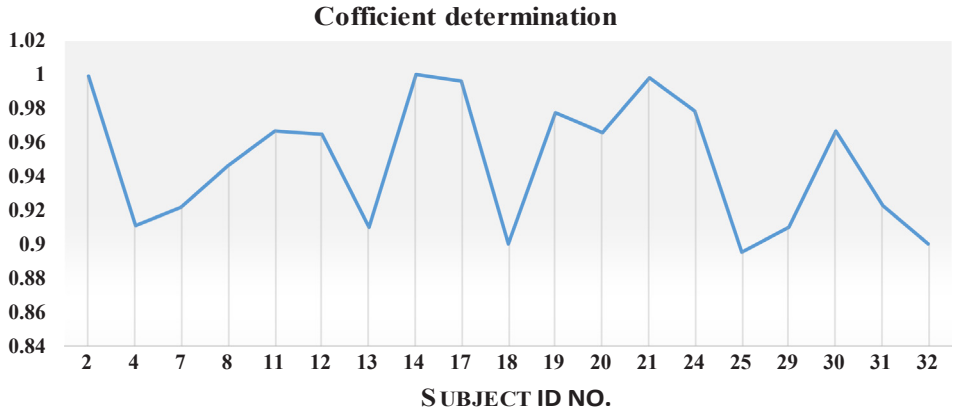


Fig. 10. Coefficient determination values across all the 22 subjects.

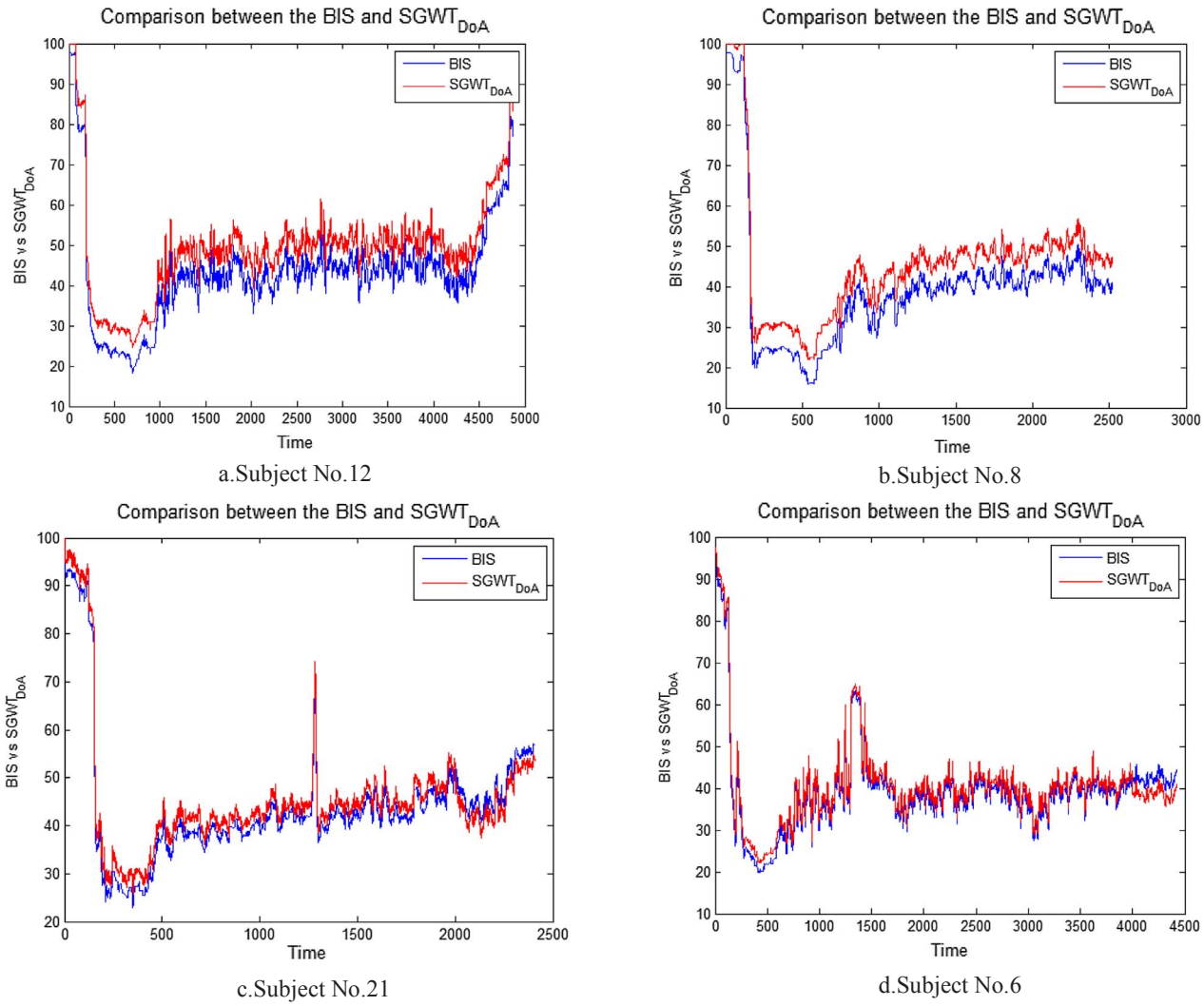


Fig. 11. Comparison between the BIS index and SGWT$_{DoA}$.

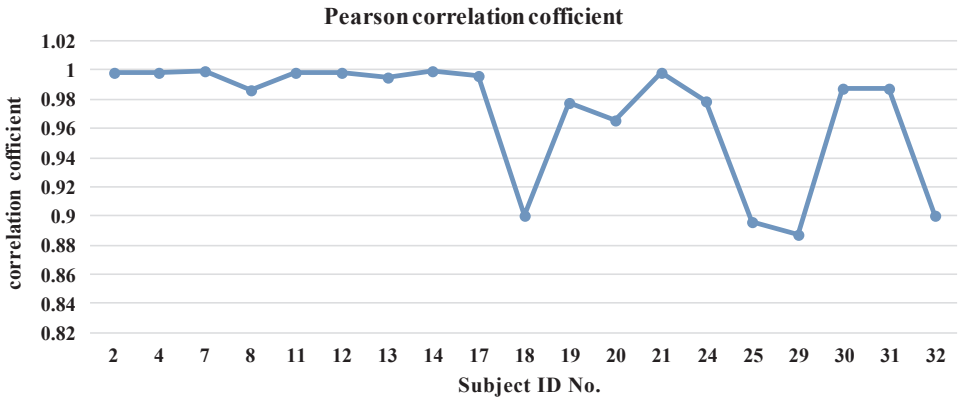


Fig. 12. Correlation coefficient value for all the subjects.

6.4. Pearson correlation

To measure the strength of a linear relationship between the BIS and the SGWT$_{DoA}$, Pearson correlation was used to verify the nature of the linear association between those indexes. It is also called correlation coefficient (r) and is defined as

$$r = \frac{\sum_i (m-\overline{m})(n-\overline{n})}{\sqrt{\sum_i (m-\overline{m})^2 (n-\overline{n})^2}}$$

where *mandn* represent the BIS and the SGWT$_{DoA}$, respectively.

The value of the correlation coefficient is between ($-1 \leqslant r \leqslant 1$). A positive value refers to a high correlation between the SGWT$_{DoA}$ and the BIS, while a negative value indicates there is a negative relationship between the indexes. The zero value means that there is no a linear

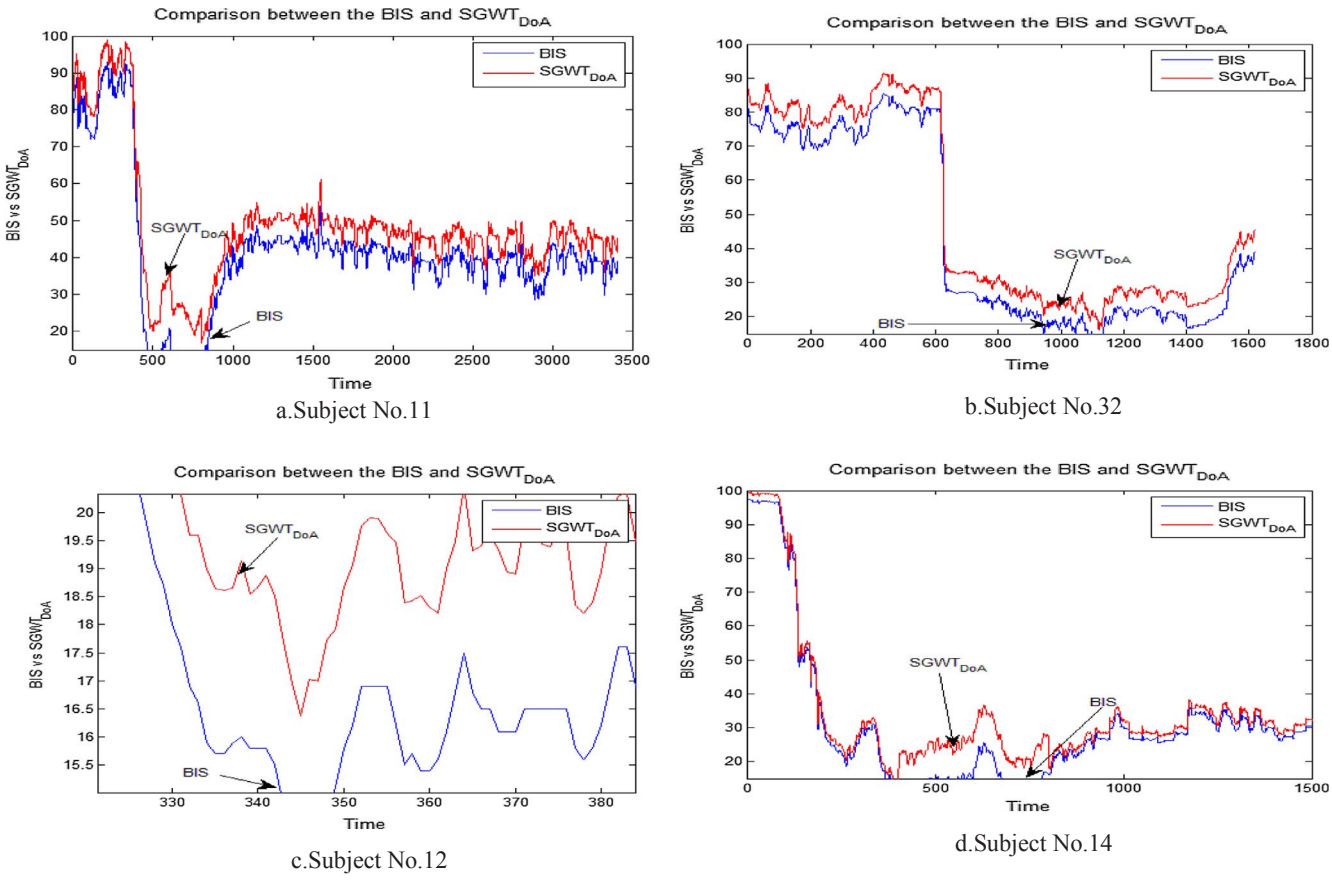


Fig. 13. Comparison between the BIS index and SGWT$_{DoA}$.

correlation between the SGWT$_{DoA}$ and the BIS. In this work, the correlation coefficient was used to evaluate the relationship between the BIS and SGWT$_{DoA}$. Fig. 11a–d show a comparison between the BIS and the SGWT$_{DoA}$ for subjects of ID Nos: 6, 8, 12 and 21. It was noticed that the behaviour of the SGWT$_{DoA}$ was similar to the BIS index and there was a high correlation between them.

Fig. 12 presents the values of the correlation coefficients for the 22 subjects. The average of the correlation coefficients was 0.99. From Figs. 11 and 12, it was observed that the proposed index yielded a high performance across all the subjects.

7. Patient state in the case of poor signal quality

The BIS monitor shows a signal quality index as well as a real time EEG signal, the BIS values, EMG and, burst and suppression ratio. The BIS index is considered an efficient method to trace the depth of anesthesia. However, the major problem with using the BIS index is that the BIS index can fail to display the values on the screen and does not fully reflect the anesthetic states when the signal quality is lower than 15. The signal quality of an EEG channel is measured by using a signal quality indicator (SQI) and it is calculated based on different variables, such as artefacts, impedance data etc. A higher SQI number (SQI $>$ 15) refers that the BIS values are reliable and more accurate while a lower SQI number (SQI $<$ 15) indicates that the BIS values can't be displayed on the screen along with other variables and parameters.

In this paper, the performance of the SGWT$_{DoA}$ was evaluated in the case of poor signal quality. The recordings were used from four subjects for which their SQIs were lower than 15 and the BIS did not display the values on the screen. Fig. 13a presents an example of the poor signal quality from subject 11. The demographic information of subject 11 were: 53 yrs. old, 129 kg, 172 height and gender/female. The surgery was started at 10:11:12 am and ended at 11:24:15 am. The medication administration contained: Medazolam 4 mg at 10:11:12 am, Alfentanil 100 μg at 10:11:26 am, Propofol 160 mg at 10:12:20 am, Morphine 5 mg, at 10:25:30 am and tramadol 100 mg at 11:23:00 am. From Fig. 13a we can see that from 500 to 900 s the BIS index failed to display its values, while at the same time the SGWT$_{DoA}$ assessed the DoA.

The performances of the SGWT$_{DoA}$ and the BIS were also evaluated using another case of poor signal quality from subject 32. The demographic information were 63 years, gender/female, 72 kg, and 154 height. She was underwent perineal surgery. The surgery was started at 10:31:35 am and ended at 10:55:26 am. The medication administration included Medazolam 4 mg at 10:31:35 am, Alfentanil 1000 μg at 10:31:55 am, Parecoxib 40 mg, at 10:32:35 am, Propofol 150 mg at 10:33:30 am and Fentanyl 100 mg at 10:40:15 am. Fig. 13b presents a case of poor signal quality from subject 32. We can observe that the BIS index dropped from 800 to 1100 s and it was unable to show the DoA values on the screen. However, the SGWT$_{DoA}$ monitored the DoA well. The obtained results in Fig. 13a and b demonstrate that the performance of the SGWT$_{DoA}$ outperforms the BIS index when the SQI was lower than 15.

Another two cases of poor signal quality were also studied in which the BIS index and SGWT$_{DoA}$ were tested. Fig. 13c and 13d present the two cases of poor signal quality. The BIS index from the two cases was dropped and did not assess the DoA. In the first case for subject 12 (Fig. 13c), the BIS index did not show values on the screen from 3450 to 3500 s while the SGWT$_{DoA}$ assessed the DoA during that time. The same situation happened with subject 14 during 400–900 s. From Fig. 13d we can see that the BIS also did not display the DoA on the monitor. However, the SGWT$_{DoA}$ could assess the DoA values.

8. Conclusion

This paper presented a novel technique to assess the DoA based on

the spectral graph wavelet transform and statistical properties. A new index of the DoA ($SGWT_{DoA}$) was designed and evaluated using the EEG recordings from 22 subjects. A set of statistical features were extracted from each EEG segment and then was mapped as a weighted graph. The graph wavelet coefficients were used to design the depth of anesthesia index. It was found that the energy of the wavelet coefficients at scale 3 changed along with the anesthetic states. The $SGWT_{DoA}$ was evaluated against the BIS index using the Q-Q plot, regression technique, Bland-Altman and Pearson correlation. The simulation results showed that the $SGWT_{DoA}$ performed better than the BIS index across all the 22 subjects and it can reflect the DoA in general anesthesia. In the case of poor signal quality, the BIS index and $SGWT_{DoA}$ were tested, and the obtained results demonstrated that the $SGWT_{DoA}$ could monitor the DoA values while the BIS failed. The proposed method can be used to help neurologists and clinicians monitor the DoA accurately during surgery. In the future we will employ big data technologies to implement the proposed method to assess the DoA, and to analyse different EEG data, such as sleep stages classification and epileptic seizures detection.

References

[1] S. Chen, R. Varma, A. Sandryhaila, J. Kovačević, Discrete signal processing on graphs: sampling theory, IEEE Trans. Signal Process. 63 (2015) 6510–6523.
[2] I. Constant, N. Sabourdin, Monitoring depth of anesthesia: from consciousness to nociception. A window on subcortical brain activity, Pediatric Anesthesia 25 (2015) 73–82.
[3] M. Diykh, Y. Li, Complex networks approach for EEG signal sleep stages classification, Expert Syst. Appl. 63 (2016) 241–248.
[4] M. Drew, K. Sheth, A Wavelet Analysis Approach for Categorizing Air Traffic Behavior, in: 15th AIAA Aviation Technology, Integration, and Operations Conference, 2015, p. 2731.
[5] O.L.M. Dusan, D.A.B. Rosas, M. Cagy, R.D.H. Idarraga, Nonlinear analysis of the electroencephalogram in depth of anesthesia, Revista Facultad de Ingeniería (2015) 45–56.
[6] A. Gadde, A. Anis, A. Ortega, Active semi-supervised learning using sampling theory for graph signals, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2014, pp. 492–501.
[7] D. Galante, D. Fortarezza, M. Caggiano, G. de Francisci, D. Pedrotti, M. Caruselli, Correlation of bispectral index (BIS) monitoring and end-tidal sevoflurane concentration in a patient with lobar holoprosencephaly, Brazil. J. Anesthesiol. (English Edition) 65 (2015) 379–383.
[8] D. Giavarina, Understanding Bland Altman analysis, Biochemia medica 25 (2015) 141–151.
[9] D. Hadzidiakos, A. Nowak, N. Laudahn, J. Baars, K. Herold, B. Rehberg, Subjective assessment of depth of anaesthesia by experienced and inexperienced anaesthetists, Eur. J. Anaesthesiol. 23 (2006) 292–299.
[10] D.K. Hammond, P. Vandergheynst, R. Gribonval, Wavelets on graphs via spectral graph theory, Appl. Comput. Harmon. Anal. 30 (2011) 129–150.
[11] L. Hasak, M. Wujtewicz, R. Owczuk, Assessment of the depth of anaesthesia during inhalational and intravenous induction of general anaesthesia, Anaesthesiol. Intensive Therapy 46 (2014) 274–279.
[12] M. Jospin, P. Caminal, E.W. Jensen, H. Litvan, M. Vallverdú, M.M. Struys, H.E. Vereecke, D.T. Kaplan, Detrended fluctuation analysis of EEG as a measure of depth of anesthesia, IEEE Trans. Biomed. Eng. 54 (2007) 840–846.
[13] A. Kalinichenko, L. Manilo, A. Nemirko, Analysis of anesthesia stages based on the EEG entropy estimation, Pattern Recognit Image Anal. 25 (2015) 632–641.
[14] H. Kang, H. Na, The assessment of effect of neuromuscular blocking agent on depth of anesthesia using the BIS and AEPindex: A-83, Eur. J. Anaesthesiol. (EJA) 23 (2006) 22.
[15] N. Leonardi, D. Van De Ville, Tight wavelet frames on multislice graphs, IEEE Trans. Signal Process. 61 (2013) 3357–3367.
[16] T. Li, P. Wen, S. Jayamaha, Anaesthetic EEG signal denoise using improved non-local mean methods, Australas. Phys. Eng. Sci. Med. 37 (2014) 431–437.
[17] T.-N. Li, Y. Li, Depth of anaesthesia monitors and the latest algorithms, Asian Pacific J. Tropic. Med. 7 (2014) 429–437.
[18] Q. Liu, Y.-F. Chen, S.-Z. Fan, M.F. Abbod, J.-S. Shieh, EEG signals analysis using multiscale entropy for depth of anesthesia monitoring during surgery through artificial neural networks, Comput. Math. Meth. Med. 2015 (2015).
[19] M. Malek, D. Helbert, P. Carré, Color graph based wavelet transform with perceptual information, J. Electron. Imaging 24 (2015) 053004-053004.
[20] S.M. Mousavi, A. Adamoğlu, T. Demiralp, M.G. Shayesteh, A wavelet transform based method to determine depth of anesthesia to prevent awareness during general anesthesia, Comput. Math. Meth. Med. 2014 (2014).
[21] I. Mporas, V. Tsirka, E.I. Zacharaki, M. Koutroumanidis, M. Richardson, V. Megalooikonomou, Seizure detection using EEG and ECG signals for computer-based monitoring, analysis and management of epileptic patients, Expert Syst. Appl. 42 (2015) 3227–3233.
[22] T. Musialowicz, P. Lahtinen, Current status of EEG-based depth-of-consciousness monitoring during general anesthesia, Curr. Anesthesiol. Reports 4 (2014) 251–260.
[23] T. Nguyen-Ky, P. Wen, Y. Li, Modified detrended fluctuation analysis method in depth of anesthesia assessment application, in: Proceedings of the International Conference on Bioinformatics and Computational Biology (BIOCOMP 2008), CSREA Press, 2008, pp. 279–284.
[24] T. Nguyen-Ky, P. Wen, Y. Li, Monitoring the depth of anesthesia using discrete wavelet transform and power spectral density, International Conference on Rough Sets and Knowledge Technology, Springer, 2009, pp. 350–357.
[25] T. Nguyen-Ky, P. Wen, Y. Li, Consciousness and depth of anesthesia assessment based on bayesian analysis of EEG signals, IEEE Trans. Biomed. Eng. 60 (2013) 1488–1498.
[26] N. Nicolaou, J. Georgiou, Neural network-based classification of anesthesia/awareness using granger causality features, Clin. EEG Neurosci. 45 (2014) 77–88.
[27] M. Özdemir, Comparison of statistical methods and wavelet energy coefficients for determining two common PQ disturbances: sag and swell, in International Conference on Electrical and Electronics Engineering, 2009. ELECO 2009. IEEE, 2009, pp. I-80-I-84.
[28] M.E. Palendeng, P. Wen, Y. Li, Real-time depth of anaesthesia assessment using strong analytical signal transform technique, Australas. Phys. Eng. Sci. Med. 37 (2014) 723–730.
[29] A. Petsiti, V. Tassoudis, G. Vretzakis, D. Zacharoulis, K. Tepetes, G. Ganeli, M. Karanikolas, Depth of anesthesia as a risk factor for perioperative morbidity, Anesthesiol. Res. Practice 2015 (2015).
[30] M.-T. Pham, G. Mercier, J. Michel, Pointwise graph-based local texture characterization for very high resolution multispectral image classification, IEEE J. Selected Topics Appl. Earth Observ. Remote Sens. 8 (2015) 1962–1973.
[31] R. Rahul, M. Sowmya, S. Rangalakshmi, B. Roshan Kumar, G. Karthik, Monitoring depth of anaesthesia using PRST score and bispectral index.
[32] B. Ricaud, D.I. Shuman, P. Vandergheynst, On the sparsity of wavelet coefficients for signals on graphs, SPIE Optical Engineering+ Applications, International Society for Optics and Photonics, 2013, pp. 88581L-88581L-88587.
[33] A. Sandryhaila, J.M. Moura, Discrete signal processing on graphs, IEEE Trans. Signal Process. 61 (2013) 1644–1656.
[34] A. Sandryhaila, J.M. Moura, Big data analysis with signal processing on graphs: representation and processing of massive data sets with irregular structure, IEEE Signal Process Mag. 31 (2014) 80–90.
[35] R. Shalbaf, A. Mehrnam, H. Behnam, Depth of anesthesia indicator using combination of complexity and frequency measures, in: 2014 21th Iranian Conference on Biomedical Engineering (ICBME), IEEE, 2014, pp. 156–160.
[36] D. Shuman, M. Faraji, P. Vandergheynst, A multiscale pyramid transform for graph signals, 2013.
[37] D.I. Shuman, C. Wiesmeyr, N. Holighaus, P. Vandergheynst, Spectrum-adapted tight graph wavelet and vertex-frequency frames, IEEE Trans. Signal Process. 63 (2015) 4223–4235.
[38] J. Smajic, M. Hodzic, S. Hodzic, A. Srabovic-Okanovic, N. Smajic, Z. Djonlagic, Assessment of depth of anesthesia: PRST score versus bispectral index, Med. Arch. 65 (2011) 216.
[39] A. Smalter, J. Huan, G. Lushington, Graph wavelet alignment kernels for drug virtual screening, J. Bioinform. Comput. Biol. 7 (2009) 473–497.
[40] C. Stam, W. De Haan, A. Daffertshofer, B. Jones, I. Manshanden, A.V.C. Van Walsum, T. Montez, J. Verbunt, J. De Munck, B. Van Dijk, Graph theoretical analysis of magnetoencephalographic functional connectivity in Alzheimer's disease, Brain 132 (2009) 213–224.
[41] M. Tan, A. Qiu, Spectral Laplace-Beltrami Wavelets With Applications in Medical Images, IEEE Trans. Med. Imag. 34 (2015) 1005–1017.
[42] N. Tremblay, P. Borgnat, Graph wavelets for multiscale community mining, IEEE Trans. Signal Process. 62 (2014) 5227–5239.
[43] J. Tünsmeyer, K. Hopster, S.B. Kästner, Clinical use of a multivariate electroencephalogram (narcotrend) for assessment of anesthetic depth in horses during Isoflurane-Xylazine anesthesia, Front. Veterinary Sci. 3 (2016).
[44] N. Tupaika, M. Vallverdu, M. Jospin, E.W. Jensen, M.M. Struys, H.E. Vereecke, A. Voss, P. Caminal, Assessment of the depth of anesthesia based on symbolic dynamics of the EEG, in: Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE, IEEE, 2010, pp. 5971–5974.
[45] Y. Wang, Z. Liang, L.J. Voss, J.W. Sleigh, X. Li, Multi-scale sample entropy of electroencephalography during sevoflurane anesthesia, J. Clin. Monitor. Comput. 28 (2014) 409–417.
[46] J. Zhang, M. Small, Complex network from pseudoperiodic time series: topology versus dynamics, Phys. Rev. Lett. 96 (2006) 238701.
[47] T. Zhong, Q. Guo, Y. Pang, L. Peng, C. Li, Comparative evaluation of the cerebral state index and the bispectral index during target-controlled infusion of propofol, British J. Anaesthesia 95 (2005) 798–802.
[48] G. Zhu, Y. Li, P.P. Wen, Analysis and classification of sleep stages based on difference visibility graphs from a single-channel EEG signal, IEEE J. Biomed. Health Inform. 18 (2014) 1813–1821.
[49] T. Zoughi, R. Boostani, M. Deypir, A wavelet-based estimating depth of anesthesia, Eng. Appl. Artif. Intell. 25 (2012) 1710–1722.

5.2 Summary of Results

Diykh et al. (2017) developed a new index for DoA assessment using spectral graph wavelet transform. The energy of the spectral graph wavelet coefficients were investigated. Based on the simulation, it was found that if the scale level was higher than five, the wavelet coefficients did not provide significant information about the EEG signals. As the result, scales one to four was investigated to test the performance of the proposed method for estimating DoA. It was found that $t = 3$ yielded better results than those of the other numbers.

The new index was evaluated and tested using different statistical metrics. The simulation results showed that the designed index has the potential to trace DoA during surgery.

Diykh et al. (2017), clearly demonstrates that the use of graph wavelet transform improves existing techniques of DoA assessment.

6

CHAPTER 6

Discussion and Conclusions

EEGs signals are an important artefact of electrical activity generated by a brain. They are generally recorded using electrodes placed on the scalp using a conductive gel. The human brain contains millions of neurons, each one generating small electrical fields. The aggregate of these electrical potentials forms an electrical reading on the scalp which can be detected and recorded.

A variety of techniques have been developed to investigate the composition of EEG signals. Fast Fourier and wavelet transforms are the most common techniques used to study the characteristics of EEG signals. They are widely used to analyse different types of EEG signals such as sleep stages, epileptic and anaesthesia EEG signals. However, as each signal type has unique patterns and characteristics, previous studies have been unable to develop a robust signal analysis approach.

This thesis presents the design and development of robust techniques that identify and analyse abnormality in different EEG signals. In this thesis three techniques, considered to be its main objectives, have been developed:

1. Development of a robust technique to score EEG sleep stages, thus improving classification accuracy.
2. Introduction of a new method to detect epileptic seizures in EEG signals.
3. Design of a new index to assess DoA accurately.

To achieve these objectives, three methods based on complex networks, a statistical model and spectral graph wavelet were developed. A summary of the developed method is provided in the following sections.

6.1 EEG sleep analysis

To identify EEG sleep stages, we introduced simple method to classify EEG sleep stages based on structural and topological complex network attributes coupled with statistical features (see Chapters 2 and 3). In Chapter 2 and Chapter 3, we applied the concept of complex networks combined with a statistical model to classify a single EEG channel signal into six sleep stages In these studies, each EEG segment was divided into a number of sub-segments and the dimensionality of each sub-segment was reduced by extracting statistical features. The statistical features of each EEG segment were mapped into a complex network.

In Chapter 2, we studied two networks attributes: average degree and Jaccard coefficients. Based on simulation results, these two attributes can identify EEG sleep stages when they were combined. However, to improve the performance of the proposed method, in Chapter 3, we made some improvements on the method developed in in Chapter 2 by investigating other networks characteristics. Degree distribution and clustering coefficients were investigated in that study. Compared with the previous study in Chapter 2, we , a 3.0% improvement in accuracy was obtained. This advance could create significant improvements in medical diagnostics.

Different machine learning techniques, including supervised and unsupervised algorithms, were used to classify the networks features. K-means was chosen as the unsupervised machine learning approach while a support vector machine (SVM) was selected as a supervised algorithm for comparisons. To assess the performance of the proposed sleep stage classification methods, data from three different sleep stage databases were used. These databases were acquired from different EEG channels and scored by either the R&K or AASM guidelines. The obtained results showed that there was a consistency in classification accuracy even though the proposed methods were applied to different EEG channels.

In Chapter 2 and 3, we clearly demonstrated that using structural and topological graphs' features could improve the classification accuracy of sleep stages. The results

showed that the networks had a higher local efficiency (clustering coefficients) during the deep sleep stages compared with the AWA stage, and that local clustering decreased significantly in SWS stages compared to the REM, S1, S2 and AWA.

To check whether the developed approaches have advantages or not, the proposed methods were compared with previous methods in which different transformation techniques were used. The results of comparisons showed the effectiveness of using complex networks to classify EEG sleep signals.

6.2 Epileptic seizure detection

EEG signals exhibit nonlinear behaviours and one of the effective non-linear methods is complex networks The excellent results achieved using complex networks to classify EEG sleep signals in Chapter 2 and 3, motivated an exploration of complex networks for epileptic seizure detection. As a result, In Chapter 4, we developed a new method to detect the abnormalities in EEG signals caused by epileptic seizures using weighted complex networks. Based on the obtained results, it was found that weighted complex networks were capable of identifying the abnormalities in epileptic EEG signals.

Each single EEG channel was divided into four segments, with each segment further divided into small blocks. A set of statistical features was extracted from each block. As a result, each EEG single channel was represented by a vector of statistical features. The extracted features were then mapped into an undirected weighted complex network.

The modularity of the networks was used to analyse epileptic EEG signals. In Chapter 3, we showed that modularity had high potential to differentiate epileptic seizure events from non-epileptic seizures events. The simulation results showed that the modularity highly reflected the sudden fluctuations in EEG signals during epileptic seizure activity and exhibited different values of modularity for each EEG group.

In Chapter 3, we also studied other global and local network features: clustering coefficients, average degree and closeness, to detect epileptic seizure events in EEG signals. The proposed method showed that the use of weighted networks reveals hidden patterns in EEG signals during epileptic seizures; patterns that are difficult to identify using unweighted networks. It was also found that the abnormalities in EEG

signals were better analysed by combining networks attributes with time domain features.

Four well known classifiers, a least support vector machine, K-means, Naïve Bayes and K-nearest, were used to classify networks attributes. The performance of the proposed method was compared with other existing methods of epileptic seizure detection. A review of the literature found that there were no methods using time domain features and complex network characteristics in epileptic seizure detection. Furthermore, most of the existing methods of epileptic detection were tested with only five EEG categories

The obtained results showed that the proposed method can be used to develop a seizure warning system, and that it can be adapted to assist doctors and neurologists in better diagnosing and treating neurological disorders.

6.3 Depth of anaesthesia assessment based on graph wavelet transform

Awareness during surgery is a very serious problem for patients as well as for anesthesiologists. Such incidents motivate most of the legal claims made by patients against anesthesiologists. Awareness happens when a patient is supposed to be anesthetized but his or her brain remains active. As a result, after surgery the patient can recall intraoperative events and suffer severe psychological problems such as nightmares, anxiety and depression.

In Chapter 4, we developed a new technique to track anaesthesia states using spectral graph wavelets. The main object of the developed method was to design a new index for delivering a sufficient amount of anaesthesia agent to patients, thus helping them avoid awareness during surgery.

The proposed method includes three parts. First, the EEG signals were de-noised using a nonlocal method. Second, a segmentation method based on a sliding window technique was used to partition EEG signals into overlapping segments. Third, each segment was mapped into an undirected weighted complex network and spectral graph wavelet transform was performed.

The cubic kernel and polynomial approximation were used to implement the spectral graph wavelet transform and to accelerate the performance of the proposed method.

The wavelet and scale coefficients were studied and the total energy of the wavelet and scaling coefficients were tested and investigated in designing the DoA index at each scale. The scale, numbering one to four, was tested. Scale number three was found to reveal the characteristics of spectral content of EEG signals.

Based on simulations, it was observed that the average energy of the wavelet coefficients showed a higher positive relationship with BIS index values than the energy of the scaling coefficients. As a result, the average energy of the wavelet coefficients was adopted to design a new index of DoA.

A set of experiments and simulations were designed using EEG data acquired from 22 subjects. The simulation results showed that the new index accurately reflects DoA

6.4 Future work

We believe that the methods developed in this thesis has high potential in EEG signals' classification and processing. Extensive future work will be carried out to apply these methods to other EEG applications.

Further work on complex networks in EEG signal analysis can be done to improve the method through a decrease in execution time. Processing real time EEG signal data in biomedical applications requires high speed techniques. A planned future work aims to reduce computational time by using fewer statistical features and applying parallel processing techniques.

In regards to dimensionality reduction, twelve and ten statistical features have been used in this thesis. To reduce the dimensionality of EEG signals, one of our future investigations is to decrease the number of features by eliminating those features that have the same behaviour and reflect the same characteristics of EEG signals. This step would decrease the processing time and reduce the memory required to process the EEG signals.

In addition, EEG signals are sometimes contaminated with different types of noise. Noise artefacts can be the result of environmental and physiological factors. In this thesis, the proposed techniques did not remove or deal with noise artefacts. Further investigation and study is required to improve the developed techniques to deal with

noisy EEG signals. This would be a significant improvement in signals processing and signals classification.

However, to increase the accuracy of the proposed methods, we will study another network feature such as the fractal dimensions of networks and spectral network attributes. Analysing fractal and multifractal characteristics of networks could help to reveal some of the hidden patterns of EEG signals that cannot be detected using topological and structural network attributes. In addition, using spectral network attributes to analyse EEG signals could exhibit abnormal behaviours in EEG signals that can be difficult to detect using topological and structural network attributes.

In addition, we will classify the extracted features using a combination of machine learning techniques instead of using a single classifier. Our studies have shown that selecting machine learning algorithms to classify the extracted features is an extremely challenging task in EEG signal classification as the quality of the classification results depends on how classification algorithms are chosen accurately. Classification algorithms are mostly evaluated in terms of multiple criteria such as accuracy, sensitivity and other metrics. The evaluation process can be carried out with a single evaluation model using multi-criteria decision making (MCDM). We will use the MCDM to choose the classification algorithms to form an ensemble classifier. Using the ensemble classifier could improve the classification accuracy and efficiency of the designed models compared with using a single classifier to classify the extracted features of EEG signals.

References

Acharya, UR, Chua, EC-P, Chua, KC, Min, LC & Tamura, T 2010, 'Analysis and automatic identification of sleep stages using higher order spectra', International journal of neural systems, vol. 20, no. 06, pp. 509-21.

Acharya, UR, Sree, SV, Swapna, G, Martis, RJ & Suri, JS 2013, 'Automated EEG anlysis of epilepsy: a review', Knowledge-Based Systems, vol. 45, pp. 147-65.

Al Ghayab, HR, Li, Y, Abdulla, S, Diykh, M & Wan, X 2016, 'Classification of epileptic EEG signals based on simple random sampling and sequential feature selection', Brain informatics, vol. 3, no. 2, pp. 85-91.

Al-Qazzaz, NK, Hamid Bin Mohd Ali, S, Ahmad, SA, Islam, MS & Escudero, J 2015, 'Selection of mother wavelet functions for multi-channel eeg signal analysis during a working memory task', Sensors, vol. 15, no. 11, pp. 29015-35.

Al-salman, W, Li, Y, Wen, P & Diykh, M 2018, 'An efficient approach for EEG sleep spindles detection based on fractal dimension coupled with time frequency image', Biomedical Signal Processing and Control, vol. 41, pp. 210-21.

Amin, HU, Malik, AS, Kamel, N & Hussain, M 2016, 'A novel approach based on data redundancy for feature extraction of EEG signals', Brain topography, vol. 29, no. 2, pp. 207-17.

Bankman, I & Gath, I 1987, 'Feature extraction and clustering of EEG during anaesthesia', Medical and Biological Engineering and Computing, vol. 25, no. 4, pp. 474-7.

Berger, H 1929, 'Über das elektrenkephalogramm des menschen', Archiv für psychiatrie und nervenkrankheiten, vol. 87, no. 1, pp. 527-70.

Berry, RB, Brooks, R, Gamaldo, CE, Harding, SM, Marcus, C & Vaughn, B 2012, 'The AASM manual for the scoring of sleep and associated events', Rules, Terminology and Technical Specifications, Darien, Illinois, American Academy of Sleep Medicine.

Carlson, NR 2002, 'Foundations of physiological psychology,' 5th ed., Boston,

Collura, TF 1993, 'History and evolution of electroencephalographic instruments and techniques', Journal of clinical neurophysiology, vol. 10, no. 4, pp. 476-504.

Diykh, M & Li, Y 2016, 'Complex networks approach for EEG signal sleep stages classification', Expert Systems with Applications, vol. 63, pp. 241-8.

Diykh, M, Li, Y & Wen, P 2016, 'EEG sleep stages classification based on time domain features and structural graph similarity', IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 24, no. 11, pp. 1159-68.

Diykh, M, Li, Y & Wen, P 2017, 'Classify epileptic EEG signals using weighted complex networks based community structure detection', Expert Systems with Applications, vol. 90, pp. 87-100.

Diykh, M, Li, Y, Wen, P & Li, T 2018, 'Complex Networks Approach for Depth of Anaesthesia Assessment', Measurement.

Donald H Edwards, Rb 2006, 'Neuroscience. Third Edition. Edited by Dale Purves, George J Augustine, David Fitzpatrick, William C Hall, Anthony-Samuel LaMantia, James O McNamara , and S Mark Williams', The Quarterly Review of Biology, vol. 81, no. 1, pp. 86-7.

Felton, EA, Wilson, JA, Williams, JC & Garell, PC 2007, 'Electrocorticographically controlled brain–computer interfaces using motor and sensory imagery in patients with temporary subdural electrode implants: report of four cases', Journal of neurosurgery, vol. 106, no. 3, pp. 495-500.

Fonseca, C, Cunha, JS, Martins, R, Ferreira, V, De Sa, JM, Barbosa, M & da Silva, AM 2007, 'A novel dry active electrode for EEG recording', IEEE Transactions on Biomedical Engineering, vol. 54, no. 1, pp. 162-5.

Fraiwan, L, Lweesy, K, Khasawneh, N, Fraiwan, M, Wenz, H & Dickhaus, H 2010, 'Classification of sleep stages using multi-wavelet time frequency entropy and LDA', Methods of information in Medicine, vol. 49, no. 3, p. 230.

Gao, V, Turek, F & Vitaterna, M 2016, 'Multiple classifier systems for automatic sleep scoring in mice', Journal of neuroscience methods, vol. 264, pp. 33-9.

Geddes, L & Roeder, R 2003, 'Criteria for the selection of materials for implanted electrodes', Annals of biomedical engineering, vol. 31, no. 7, pp. 879-90.

Giri, EP, Arymurthy, AM, Fanany, MI & Wijaya, SK 2015, 'Sleep stages classification using shallow classifiers', in Advanced Computer Science and Information Systems (ICACSIS), 2015 International Conference on, pp. 297-301.

Gloor, P 1994, 'Berger lecture. Is Berger's dream coming true', Electroencephalography and clinical neurophysiology, vol. 90, no. 4, pp. 253-66.

Harvard health online 2018, https://www.health.harvard.edu.

Henao-Idarraga, RD 2016, 'Nonlinear analysis of the electroencephalogram in depth of anaesthesia ', Revista de la Facultad de Ingeniería, vol. 30, no. 2.

Herrera, LJ, Fernandes, CM, Mora, AM, Migotina, D, Largo, R, Guillén, A & Rosa, AC 2013, 'Combination of heterogeneous EEG feature extraction methods and stacked sequential learning for sleep stage classification', International journal of neural systems, vol. 23, no. 03, p. 1350012.

Holly Nelson, 2015, https://www.quora.com.

Holmes, GL & Khazipov, R 2007, 'Basic neurophysiology and the cortical basis of EEG', in The clinical neurophysiology primer, Springer, pp. 19-33.

Kauhanen, L, Nykopp, T, Lehtonen, J, Jylanki, P, Heikkonen, J, Rantanen, P, Alaranta, H & Sams, M 2006, 'EEG and MEG brain-computer interface for tetraplegic patients', IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 14, no. 2, pp. 190-3.

Kayikcioglu, T, Maleki, M & Eroglu, K 2015, 'Fast and accurate PLS-based classification of EEG sleep using single channel data', Expert Systems with Applications, vol. 42, no. 21, pp. 7825-30.

Kemp, B, Zwinderman, AH, Tuk, B, Kamphuisen, HA & Oberye, JJ 2000, 'Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG', IEEE Transactions on Biomedical Engineering, vol. 47, no. 9, pp. 1185-94.

Kiernan, J 1886, 'Psychiatry: A Clinical Treatise on Diseases of the Fore-Brain, based upon a study of its structure, functions, and nutrition', The Journal of Nervous and Mental Disease, vol. 13, no. 2, pp. 115-7.

Klem, GH, LuÈders, HO, Jasper, H & Elger, C 1999, 'The ten-twenty electrode system of the International Federation', Electroencephalogr Clin Neurophysiol, vol. 52, no. 3, pp. 3-6.

Klimesch, W 1997, 'EEG-alpha rhythms and memory processes', International Journal of psychophysiology, vol. 26, no. 1-3, pp. 319-40.

Lajnef, T, Chaibi, S, Ruby, P, Aguera, P-E, Eichenlaub, J-B, Samet, M, Kachouri, A & Jerbi, K 2015, 'Learning machines and sleeping brains: automatic sleep stage classification using decision-tree multi-class support vector machines', Journal of neuroscience methods, vol. 250, pp. 94-105.

Lee, J-M, Kim, D-J, Kim, I-Y, Park, KS & Kim, SI 2004, 'Nonlinear-analysis of human sleep EEG using detrended fluctuation analysis', Medical Engineering and Physics, vol. 26, no. 9, pp. 773-6.

Lee, J-M, Kim, D-J, Kim, I-Y, Park, KS & Kim, SI 2004, 'Nonlinear-analysis of human sleep EEG using detrended fluctuation analysis', Medical Engineering and Physics, vol. 26, no. 9, pp. 773-6.

Lindsley, DB 1936, 'Brain potentials in children and adults', Science.

Liu, H, Wang, J, Zheng, C & He, P 2006, 'Study on the effect of different frequency bands of EEG signals on mental tasks classification', in Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the, pp. 5369-72.

Loomis, AL, Harvey, EN & Hobart, G 1935, 'Potential rhythms of the cerebral cortex during sleep', Science.

Lotte, F 2008, 'Study of electroencephalographic signal processing and classification techniques towards the use of brain-computer interfaces in virtual reality applications', INSA de Rennes.

Mason, SG & Birch, GE 2003, 'A general framework for brain-computer interface design', IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 11, no. 1, pp. 70-85.

Mellinger, J, Schalk, G, Braun, C, Preissl, H, Rosenstiel, W, Birbaumer, N & Kübler, A 2007, 'An MEG-based brain–computer interface (BCI)', Neuroimage, vol. 36, no. 3, pp. 581-93.

Millett, D 2001, 'Hans Berger: From psychic energy to the EEG', Perspectives in biology and medicine, vol. 44, no. 4, pp. 522-42.

Morshed, BI & Khan, A 2014, 'A brief review of brain signal monitoring technologies for BCI applications: challenges and prospects', Journal of Bioengineering & Biomedical Sciences, vol. 4, no. 1, p. 1.

Nguyen-Ky, T, Wen, PP & Li, Y 2013, 'Consciousness and depth of anaesthesia assessment based on bayesian analysis of EEG signals', IEEE Transactions on Biomedical Engineering, vol. 60, no. 6, pp. 1488-98.

Nicolas-Alonso, LF & Gomez-Gil, J 2012, 'Brain computer interfaces, a review', Sensors, vol. 12, no. 2, pp. 1211-79.

Petsiti, A, Tassoudis, V, Vretzakis, G, Zacharoulis, D, Tepetes, K, Ganeli, G & Karanikolas, M 2015, 'Depth of anaesthesia as a risk factor for perioperative morbidity', Anesthesiology research and practice, vol. 2015.

Ramadan, RA, Refat, S, Elshahed, MA & Ali, RA 2015, 'Basics of brain computer interface', in Brain-Computer Interfaces, Springer, pp. 31-50.

Rao, TK, Lakshmi, MR & Prasad, T 2012, 'An exploration on brain computer interface and its recent trends', arXiv preprint arXiv:1211.2737.

Rechtschaffen, A 1968, 'A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects', Public health service.

Sahi, A, Lai, D, Li, Y & Diykh, M 2017, 'An efficient DDoS TCP flood attack detection and prevention system in a cloud environment', IEEE Access, vol. 5, pp. 6036-48.

Salvo, P, Raedt, R, Carrette, E, Schaubroeck, D, Vanfleteren, J & Cardon, L 2012, 'A 3D printed dry electrode for ECG/EEG recording', Sensors and Actuators A: Physical, vol. 174, pp. 96-102.

Sanei, S & Chambers, JA 2013, EEG signal processing, John Wiley & Sons.

Schnitzler, A & Gross, J 2005, 'Normal and pathological oscillatory communication in the brain', Nature reviews neuroscience, vol. 6, no. 4, p. 285.

Schwary, G, Litscher, G, Wang, L, Schoepfer, A & Roetzer, I 2005, 'The effect of acupressure on the bispectral index and entropy parameters in mentally handicapped humans: a pilot study', Intern J Neuromonit, vol. 4, no. 1.

Şen, B, Peker, M, Çavuşoğlu, A & Çelebi, FV 2014, 'A comparative study on classification of sleep stage based on EEG signals using feature selection and classification algorithms', Journal of medical systems, vol. 38, no. 3, pp. 1-21.

Tallgren, P, Vanhatalo, S, Kaila, K & Voipio, J 2005, 'Evaluation of commercially available electrodes and gels for recording of slow EEG potentials', Clinical Neurophysiology, vol. 116, no. 4, pp. 799-806.

Teplan, M 2002, 'Fundamentals of EEG measurement', Measurement science review, vol. 2, no. 2, pp. 1-11.

Thut, G, Miniussi, C & Gross, J 2012, 'The functional importance of rhythmic activity in the brain', Current Biology, vol. 22, no. 16, pp. R658-R63.

Uğuz, H 2012, 'A hybrid system based on information gain and principal component analysis for the classification of transcranial Doppler signals', Computer methods and programs in biomedicine, vol. 107, no. 3, pp. 598-609.

van Erp, J, Lotte, F & Tangermann, M 2012, 'Brain-computer interfaces: beyond medical applications', Computer, vol. 45, no. 4, pp. 26-34.

Vaughan, TM, Wolpaw, JR & Donchin, E 1996, 'EEG-based communication: prospects and problems', IEEE transactions on rehabilitation engineering, vol. 4, no. 4, pp. 425-30.

Wang, X-J 2010, 'Neurophysiological and computational principles of cortical rhythms in cognition', Physiological reviews, vol. 90, no. 3, pp. 1195-268.

Wolpaw, JR, McFarland, DJ & Vaughan, TM 2000, 'Brain-computer interface research at the Wadsworth Center', IEEE transactions on rehabilitation engineering, vol. 8, no. 2, pp. 222-6.

Yeung, J 2010, 'Sedation', Cambridge:Cambridge: Cambridge University Press, pp. 77-84.

Zaehle, T, Rach, S & Herrmann, CS 2010, 'Transcranial alternating current stimulation enhances individual alpha activity in human EEG', PloS one, vol. 5, no. 11, p. e13766.

Zhang, J, Wu, Y, Bai, J & Chen, F 2016, 'Automatic sleep stage classification based on sparse deep belief net and combination of multiple classifiers', Transactions of the Institute of Measurement and Control, vol. 38, no. 4, pp. 435-51.

Zhu, G, Li, Y & Wen, PP 2014, 'Analysis and classification of sleep stages based on difference visibility graphs from a single-channel EEG signal', Biomedical and Health Informatics, IEEE Journal of, vol. 18, no. 6, pp. 1813-21.

Appendix A

An Efficient Approach for EEG Sleep Spindles Detection Based on Fractal Dimension Coupled with Time Frequency Image

A contribution was made in this paper. Analysing some of the obtained results, and editing.

Biomedical Signal Processing and Control 41 (2018) 210–221

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc

An efficient approach for EEG sleep spindles detection based on fractal dimension coupled with time frequency image

Wessam Al-salman*, Yan Li, Peng Wen, Mohammed Diykh

School of Agricultural, Computational and Environmental Sciences, University of Southern Queensland, Australia

ARTICLE INFO

Article history:
Received 25 April 2017
Received in revised form 9 October 2017
Accepted 26 November 2017
Available online 13 December 2017

Keywords:
Sleep spindles
Time frequency image
Fractal dimension
Box counting
EEG signals

ABSTRACT

Detection of the characteristics of the sleep stages, such as sleep spindles and K-complexes in EEG signals, is a challenging task in sleep research as visually detecting them requires high skills and efforts from sleep experts. In this paper, we propose a robust method based on time frequency image (TFI) and fractal dimension (FD) to detect sleep spindles in EEG signals. The EEG signals are divided into segments using a sliding window technique. The window size is set to 0.5 s with an overlapping of 0.4 s. A short time Fourier transform (STFT) is applied to obtain a TFI from each EEG segment. Each TFI is converted into an 8-bit binary image. Then, a box counting method is applied to estimate and discover the FDs of EEG signals. Different sets of features are extracted from each TFI after applying a statistical model to the FD of each TFI. The extracted statistical features are fed to a least square support vector machine (LS_SVM) to figure out the best combination of the features. As a result, the proposed method is found to have a high classification rate with the eight features sets. To verify the effectiveness of the proposed method, different classifiers, including a K-means, Naive Bayes and a neural network, are also employed. In this paper, the proposed method is evaluated using two publically available datasets: Dream sleep spindles and Montreal archive of sleep studies. The proposed method is compared with the current existing methods, and the results revealed that the proposed method outperformed the others. An average accuracy of 98.6% and 97.1% is obtained by the proposed method for the two datasets, respectively.

1. Introduction

Sleep scoring is a challenging task in sleep classification research due to the characteristics of the sleep stages vary [12,31,39]. According to the Rechtsaffen and Kales (R&K) guidelines [49], a human sleep cycle is divided into two main parts: the non-rapid eyes movements sleep (NREM) and rapid eyes movements sleep (REM), where the NREM includes four stages namely: Stage 1 (S1), Stage 2 (S2), Stage 3 (S3) and Stage 4 (S4).

The guidelines of the R&K have been modified by the American Academy of Sleep Medicine (AASM) in 2002. The AASM presented a different version of sleep scoring [28] by which the NREM is reduced to three stages, with S3 and S4 are combined into one stage as slow wave stage (SWS). Much clinical research have revealed that individual sleep stages exhibit unique electroencephalogram (EEG) patterns and characteristics that reflect human states whether he/she is awake or asleep. Those characteristics of sleep stages reflect the changes in brain neurons and muscles at each sleep stages [11]. Analyzing those brain waveforms is an important task for neurologists to score and analyse EEG sleep signals [17,29].

Two of the important transiting bio-signal waveforms in sleep stages are sleep spindles and k-complexes that are often used to score sleep stages [28]. Sleep spindles are the most important transient events to detect sleep stage 2 in EEG signals. They are defined as a series of distinct waves which are within a frequency range of 11–16 Hz with a minimum duration of 0.5 s (s) [60,28]. Some studies reported that, the minimum and maximum durations of sleep spindles are 0.5 s and 3s, respectively [30,60,13], with an amplitude from 5 μV to 25 μV [34]. The presence or absence of sleep spindles in EEG sleep signals has a high impact on the memory consolidation of humans [35,42]. From EEG recordings, it is observed that any change in the density of sleep spindles can result in some sleep disorders, such as insomnia and schizophrenia and autism [20,59]. Consequently, automatically detecting and analyzing sleep spindles can help experts in diagnosing sleep disorders.

* Corresponding author.
E-mail addresses: WessamAbbasHamed.Al-Salman@usq.edu.au, wessam.abbas1980@yahoo.com (W. Al-salman), Yan.Li@usq.edu.au (Y. Li), Peng.Wen@usq.edu.au (P. Wen), Mohammed.Diykh@usq.edu.au (M. Diykh).

https://doi.org/10.1016/j.bspc.2017.11.019
1746-8094/

Traditionally, the detection of sleep spindles mainly depends on visual inspection that is carried out based on the knowledge of clinicians or sleep expert. The accuracy and reliability of the manual scoring are based on the experiences of experts. Visual scoring of sleep spindles is very time consuming, subjective and prone to errors due to there are typically thousands of sleep spindles occurred in each EEG recording [1]. Identifying sleep spindles in EEG signals visually requires high skills from experts. However, developing an automatic approach to identify those marked occurrences in the sleep stages is an ongoing challenge.

Various attempts were made in identifying sleep spindles based on Fourier, wavelet and hybrid transforms [16,27,54,26]. Machine learning methods, such as support vector machines, neural networks, and genetic algorithms, were also employed to classify the extracted features by those transformation techniques [2,3,38]. Yücelbaş et al. [61] used a short time Fourier transform (STFT) combined with an artificial neural network to detect sleep spindles in EEG signals. The STFT was also used as a feature extractor by da Costa et al. [14]. The extracted features were fed to a K-means to recognize the segments of sleep spindles from non-sleep spindles segments. Estévez et al. [18] propounded a merge neural gas model with the STFT to analyse EEG signals. A maximum sensitivity of sleep spindles detection was 62.9%. Güneş et al. [24] utilized the STFT to decompose an EEG signal. The most discriminating features were extracted from the frequencies of interest. The extracted features were forwarded to two machine learning methods: a support vector machine and a multilayer perceptron to detect sleep spindles.

Recently, many researchers reported the detection of sleep spindles based on a matching pursuit and filtering techniques. Ventouras et al. [58] utilized a bandpass filter with an artificial neural network to detect sleep spindles. The obtained results in terms of sensitivity and accuracy were reported. In that study, an average of 87.5% accuracy was achieved. Żygierewicz et al. [66] presented a matching pursuit method to detect sleep spindles. The maximum sensitivity reported in that study was 90%. Schönwald et al. [52] also employed the matching pursuit to detect sleep spindles based on the amplitude, frequency, and duration characteristics of the signals. An average of sensitivity and specificity of 80.6% and 81.2% were achieved, respectively.

According to the literature, we found that the fractal dimension has been proved to be an efficient approach to explore the hidden patterns in digital images and signals. It has been used to analyse EEG signals to trace the changes in EEG signals during different sleep stages, and also was employed to recognize different digital images patterns. Yang et al., [63] and Sourina et al. [56] applied a fractal dimension technique to analyse sleep stages in EEG signals. Ali et al. [7] also utilized a fractal dimension technique for voice recognition. Furthermore, a time frequency image (TFI) has been used to analyse different types of EEG signals, such as EEG sleep stages signals. Bajaj and Pachori [9] identified EEG sleep stages based on time frequency images. Fu et al., in [22] used a time frequency image as a features extractor for epileptic seizures classification. Bajaj et al., [8] also classified alcoholic EEGs based on time frequency images.

Although the existing methods have achieved some good results in sleep spindles detection, a considerable amount of further improvement on the existing methods are still in demand. In this paper, the fractal dimension combined with time frequency images is used to detect sleep spindles in EEG signals. Firstly, each EEG signal is partitioned into segments of 0.5s. Then, each segment is transformed into a time frequency image using a short time Fourier transform (STFT). Each TFI is converted into a binary image. The box counting technique is applied to each TFI and the statistical features are extracted from the FD. Different set of statistical features are extracted and tested from the FDs to figure out the best combination of features for detecting sleep spindles. Different classifiers are also used to validate the proposed method. The obtained results showed that the proposed method achieved a high accuracy for detecting sleep spindles in EEG signals.

The rest of this paper is organized as follows: Section 2 describes the EEG datasets used. Section 3 presents the methodology of the proposed method. The experimental results are explained in Section 4. Finally the discussions, conclusions and future work are provided in Section 5.

2. Experimental EEG data

In this study, two different datasets were used to evaluate the proposed method for detecting sleep spindles in EEG signals. Those databases that are publicly available are: the DREAMS datasets (Devusty) [15] and Montreal Archive Sleep Studies (MASS) (O'Reilly et al. [40]. The following section briefly explains the details of the two datasets.

2.1. The dream sleep spindles dataset (Datsaet-1)

The EEG data sets used in this paper were collected through the Dream Project at University of Mons-TCTS Laboratory (Devyust et al.). The sleep EEG data sets were recorded from eight subjects with various sleep diseases, such as dysomnia, restless legs syndrome, insomnia, and apnea/hypopnea syndrome. The subjects were aged between 30 and 55 years. The signals were recorded in 30 min intervals during a whole night. The recorded signals were scored, and the ending and starting time instances of the sleep spindles were marked. Six of the EEG recordings were sampled at 200 Hz, while the other two recordings were sampled at 100 Hz and 50 Hz. Each EEG recoding included with two EOG channels of P8-A1 and P18-A1, three EEG channels of CZ-A1 or C3-A1, FP1-A and O1-A1, and one EMG channel. The sleep spindles in the Dream database were detected manually by two experts. The first expert scored all the eight recordings, while the second expert annotated six recordings out of the eight EEG recordings. In this study, the CZ-A1 channel and the EEG recording sampled at 200 Hz were used. The subjects selected were subject IDs 2, 4, 5, 6, 7 and 8. Table 1 shows the number of the segments that were used in this research. The dataset along with additional information is publicly available from: http:\\www.tcts.fpms.ac.be/~devuyst/Database/DatabaseSpindles.

2.2. Montreal archive of sleep studies (Dataset-2)

The database was recorded from 19 subjects: 8 males and 11 females. The age of the subjects was between 30–55 years. The EEG signals were recorded in 20 min intervals during a whole night. The EEG signals were sampled at 256 Hz. Each EEG recording included 19 EEG channels, four Electrooculography (EOG), electromyography (EMG) and Electrocardiography (ECG) channels. In this database the visual scoring of sleep spindles were carried out aslo by two experts. The first expert annotated 19 recordings, including sleep spindles according to the AASM rules, while the second only annotated 15 out of 19 recordings, including sleep spindles according to the R&K criteria. In this study, the EEG scoring from six subjects were chosen randomly. The subjects selected were subject IDs 1, 2, 7, 9, 14 and 18. Table 2 shows the number of segments that were used in this research. The datasets can be accessed through *http://www.ceams-carsm.ca/en/MASS*. Tables 1 and 2 included five columns: namely, subject ID, the number of segments with sleep spindles, the number of all segments in all EEG signals, minimum and maximum sleep spindles. The experiments were conducted using Matlab software (Version: R2015) on a computer with the

Table 1
The number of segments for each subject (Dataset-1).

Subject ID.	No. of segments with sleep spindles	No. of all segments in EEG signals	Minimum Spindle Period (second)	Maximum Spindle Period (second)
ID2	60	3599	0.5s	1.1s
ID4	44	1799	0.5s	1.8s
ID5	56	1219	0.5s	1.2s
ID6	72	2342	0.5s	1.5s
ID7	18	1869	0.5s	1.3s
ID8	48	4589	0.5s	1.9s
Total	**298**	**15417**	–	–

Table 2
The number of segments for each subject (Dataset-2).

Subject ID.	No. of segments with sleep spindles	No. of all segments in EEG signals	Minimum Spindle Period (second)	Maximum Spindle Period (second)
ID1	1040	28958	0.5s	1.3s
ID2	1141	32360	0.5s	1.2s
ID7	905	20280	0.5s	1.1s
ID9	810	27600	0.5s	1.0s
ID14	708	30320	0.5s	1.6s
ID18	1156	28640	0.5s	1.2s
Total	5760	**168150**	–	–

following settings: 3.40 GHz Intel(R) core(TM) i7 CPU processor machine, and 8 GB RAM.

3. Methodology

In this study, an efficient technique to detect sleep spindles is presented based on a short time Fourier transform (STFT) and the original EEG signals are divided into segments by a sliding window technique. The size of the window is set to 0.5 s with an overlapping of 0.4s. Then, each EEG segment is passed through the STFT to obtain its time frequency image (TFI). The obtained TFI is transformed into an 8-bit binary image. Then, a box counting method is applied to each TFI to calculate the fractal dimension, as well as to extract the features of interest. Eight statistical features are extracted from each FD of the TFI. The extracted features are used as the input to different classifiers, including a LS_SVM, K-means, Naive Bayes and a neural network. For further investigation, different features sets, including two, four, six and eight features sets, from each TFI, are tested. Comparisons are then made with the previous studies. The obtained results showed that the proposed method provided better classification results than the other methods. Fig. 1 depicts the methodology of the proposed method.

3.1. Segmentation

In this paper, a sliding window is used to segment the EEG signals into small intervals. A window size of 0.5 s is empirically selected and used to separate EEG signals with an overlapping of 0.4s. Different window sizes are tested and applied in order to figure out the best window size. The obtained results from the proposed scheme revealed that a window of 0.5 s gives better results than other window sizes. Fig. 2 shows an EEG signal being divided into segments with an overlapping of 0.4s.

3.2. Spectrogram

The main formula of the STFT is defined as [8,9]:

$$X(n,\omega) = \sum_{m=-\infty}^{\infty} x[m]\, w[n-m]\, e^{-jwn} \tag{1}$$

where $x[m]\, w[n-m]$ is a short time of signal X at time n.

The discrete STFT can be formulated as

$$X(n,k) = X(n,\omega)|_{\omega} = \frac{2\pi k}{N} \tag{2}$$

where N refers to the number of discrete frequencies.

Before calculating the Fourier transform, the centered function $w = [m]$ at time n was multiplied with signal X. The Fourier transform is an estimate at time n, and the window function of signal X is considered close to time n. To obtain the STFT, a fixed positive function was used, which is denoted as$w[m]$. However, the spectrogram can be formulated as:

$$S(n,k) = |X(n,\omega)|^2 \tag{3}$$

An EEG signal is transformed into time frequency domain. Then, the spectrogram of the STFT is applied to obtain the time frequency images (TFIs) of the EEG signals. The STFT spectrogram is defined as the normalized and squared magnitude of the STFT coefficients.

The STFT coefficients are obtained using a sliding window in time domain in order to divide the signals into smaller blocks. Each block is then analyzed using Fourier transform to determine their frequencies. Thus a time varying spectrum can be obtained. Based on Eqs. (1) and (2), the spectrogram of the signal can be calculated from the square of the discrete STFT.

Based on the literature, it is found that the spectrogram is an effective approach to analyse non-stationary and periodic signals. In this paper, the spectrogram is applied to each EEG segment to obtain the TFIs.

3.3. Fractal dimension based on box counting method (BCM)

Fractal is a scale which is used to represent a geometric pattern that cannot be represented by a classical geometry. It allows to measure the degree of complexity of an object. Based on fractal concept, each figure is presented using a series of fragments that each one can be represented as a figure. Those fragmented parts can be used to reflect the original image. There are some criteria used to define the fractal:

(1) Fractal is a simple structure with small scales.

(2) Fractal cannot be described using the traditional Euclidean geometry.

One of the main fractal features is that it possesses scaling properties. By using a fractal, an one-dimension object can be segmented into n equal parts. Each part can be scaled down by a ratio *of* $\mathrm{r} = \frac{1}{\mathrm{n}}$.

Another example is for two-dimensional objects. For example, a square area in a plane, which can be separated into n self-similar

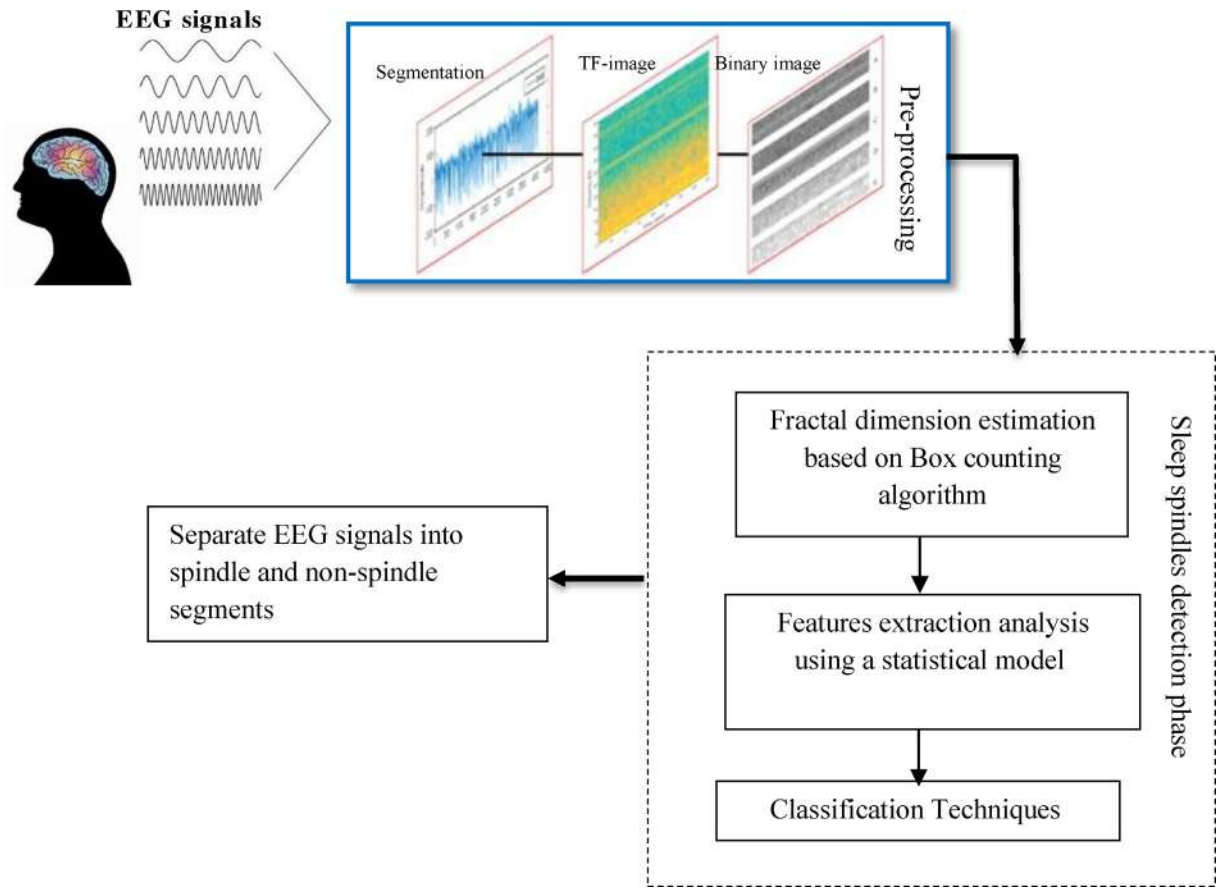


Fig. 1. The methodology of the proposed method for sleep spindles detection.

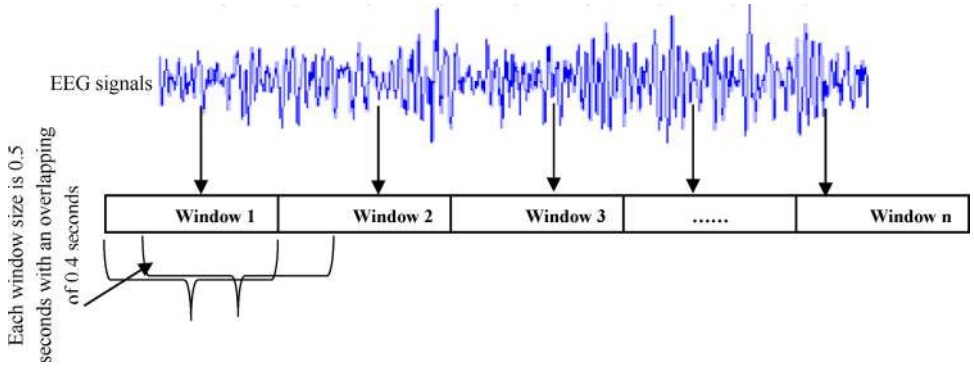


Fig. 2. An example of segmenting an EEG signal into windows.

parts, with each one scaled by a factor of $r = \frac{1}{\sqrt{n}}$. Further, a solid cube is an example of three-dimensional objects, which could also be partitioned into n little cubes with each one is scaled down by a ratio of $r = \frac{1}{\sqrt[3]{n}}$. D-dimensional self-similar objects are, therefore, scaled down by a factor of $r = \frac{1}{\sqrt[D]{N}}$. They can be partitioned into n smaller parts, with each one is scaled down by a factor of $r = \frac{1}{\sqrt[D]{N}}$.

As a result, a self-similar object of N parts can be scaled by a ratio r from the whole. Its fractal or similarity dimension is given by:

$$D = \frac{Log\,(N)}{Log\left(\frac{1}{r}\right)} \tag{4}$$

The fractal dimension is normally not an integer number. For example, von Koch curve is constructed from four sub-segments. Each one is scaled down by a factor of $\frac{1}{3}$. By applying the above equation, the fractal is equal to 1.26. The obtained results are often a non-integer value that is greater than one and less than two.

Extracting features from images is a common step that is used in various image applications, by which the important features, such as texture and color features can be pulled out. A fractal dimension (FD) technique is one of the powerful methods to extract the hidden patterns in images [45]. It is commonly used to explore the key patterns in biomedical signals and images [37]. It has been used to analyse and classify EEGs, EMG and ECG [62,21,32]. The term of fractal dimension refers to any fractal characteristics, such as information dimensions, capacity dimensions and correlation dimensions [47]. In this paper, the capacity dimensions are used.

A box counting algorithm is one of the fractal dimension methods, which is used to obtain the FD of an image or a signal [50,61]. In this paper, the box counting algorithm is used to estimate the FD of

Table 3
The numbers of non-empty grid (box size) in ten scale.

Box size δ	1	2	4	8	16	32	64	128	256	512	1024
No. of box $N(\delta)$	435823	110918	28205	7321	1973	571	166	42	12	4	1
Log$(1/\delta)$	0	0.30102	0.60205	0.90308	1.20411	1.50514	1.80617	2.10720	2.40823	2.70926	3.01029
logN (δ)	5.6393	5.04500	4.45032	3.8645	3.29512	2.75663	2.22201	1.62324	1.07918	0.60206	0

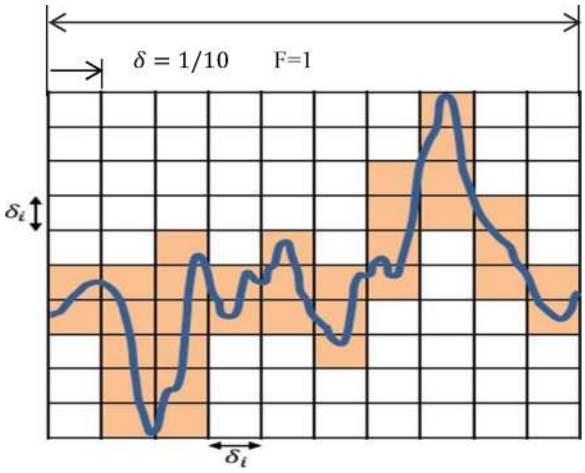


Fig. 3. An illustration of the box counting algorithm to create the size (δ) and the numbers of boxes N_δ.

a TFI to detect sleep spindles in EEG signals. Each TFI is converted into a gray scale image. Each gray scale image is then translated into a binary image before applying the box counting algorithm.

To convert a grayscale image into a binary image, a predefined threshold value (δ) is used based on the following equation.

$$Im\,(p(i)) > \delta \rightarrow 1; Im\,(p(i)) < \delta \rightarrow 0 \quad (5)$$

where Im is an image, $p\,(i)$ refers to the ith pixel, and δ is a predefined threshold. Each pixel value is set to 0 or 1, based on Eq. (5). If the pixel value is greater than or equal to the threshold, then the pixel is set to 1, otherwise 0.

The main concept of the box counting method can be described as follows: assume **X** is a TFI, and we need to determine the FD of **X**. The following equation is used [60,45].

$$D_\beta = \lim_{\delta \to 0} logN(\delta)/\log(1/\delta) \quad (6)$$

where D_β is a fractal dimension, $N(\delta)$ is the total number of boxes, and δ is the size of boxes that is required to cover image **X**. In order to cover the entire TFI in this paper, different sizes of boxes are tested and $N(\delta)$ and δ are determined. For example, to estimate the FD of a TFI, firstly, the TFI is normalized by rescaling it from the size of n x n to the size of m x m. We use an image scaling technique called nearest neighbour to rescale the TFIs. A TFI is converted from one resolution/dimension to another one without losing the visual content. Nearest neighbour is one of the fastest and simplest forms of rescaling techniques. During enlarging (upscaling), the empty spaces will be replaced with the nearest neighbouring pixels. For shrinking, the pixel sizes are reduced. One of the TFIs is used as a reference image or a base image to construct a new scaled image. This new rescaled image is used to rescale other IFIs. The number of the boxes that are required to cover a TFI and the size of each box are then investigated. That means at each iteration, a different number of boxes with different sizes of boxes are tested until the values of δ and N are decided. Fig. 3 shows an example of how to create the size and the number of boxes using the box counting algorithm. By using Eq. (6), the fractal of each TFI can be obtained from a slope of the least square fit of $\log N(\delta) = versus - \log(1/\delta)$.

Table 3 shows the number of the boxes that are required to cover the entire TFI by which the FD can be estimated.

If the box size $\doteq$ is 32 and the number of boxes that are required to cover the curve is 571, based on the equation , $\log N(\delta) = versus - \log(1/\delta)$, the fractal value for the sixth features (*FD6)* is equal to 1.788. The same procedure is applied to get all the features. The fractal dimension values are between 1 and 2 and all the FD values are non-integer.

3.4. Extracted features

As mentioned before in Section 3.3, the FD is calculated after transferring an EEG signal into a TFI using the STFT. The obtained result of the TFI is converted into an 8-bit binary image. To extract the features from each image, the box-counting algorithm is applied to each TFI and a set of statistical features are then extracted based on the fractal estimation values. Each element in the fractal dimension features is computed using Eq. (6) and the fractal is obtained based on the slope of least square best straight line. Ten fractal dimension features are extracted from each TFI, and they are denoted as FD = {*FD1, FD2, FD3 . . . FD10*}, where the number of the features corresponding to the number of iterations used to cover each TFI. Different numbers of boxes with different sizes of boxes are tested until an optimal number of boxes is obtained to cover the whole TFI. At each iteration, a fractal feature is extracted. The procedure is repeated 10 times in this paper when the maximum number of boxes to cover the whole TFI is reached. As a result, 10 features are extracted. A statistical model is then used to extract the statistical characteristics of the fractal dimension features and the number of the features is reduced to eight [63].

Different combinations of these statistical features including two, four, six and eight features sets, are tested in this paper to find out the best combination to represent each TFI. Table 4 presents the formulae of the eight statistical features.

- $F_{mean} = mean\,(FD)$
- $F_{max} = max\,(FD)$
- $F_{meadin} = median\,(FD)$
- $F_{SD} = standard\ deviation\,(FD)$
- $F_{min} = min\,(FD)$
- $F_{Sk} = skewness\,(FD)$
- $F_{rang} = Range\,(FD)$
- $F_{ku} = kurtosis\,(FD)$

Where N is the length of *FD*, m is the *mean* of the *FD* [16,34,53].

3.5. Classifiers

To evaluate the performance of the proposed method to detect sleep spindles in EEG signals, different classification methods are used and tested. The features extracted from each TFI are used as the input to the LS_SVM as well as to a K-means, Naïve Bayes and neural network classifiers. The used classifiers are briefly discussed in this section.

3.5.1. Least square support vector machine (LS_SVM)

The LS_SVM is a robust method for signals regression and classification. It is a popular classifier due to its high accuracy and with a

Table 4
Definitions of the statistical features.

No.	Features name	Formula	No.	Features name	Formula
1	*mean (FD)*	$F_{mean} = \frac{1}{N}\sum_{n=1}^{N} FD_n$	5	*standarddeviation (FD)*	$F_{SD} = \sqrt{\sum_{n=1}^{n}(FD_n - m)^{\frac{2}{n-1}}}$
2	*max (FD)*	$F_{max} = \max[FD_n]$	6	*Range (FD)*	$F_{rang} = F_{max-}F_{min}$
3	*min (FD)*	$F_{min} = min\,[FD_n]$	7	*skewness (FD)*	$F_{Sk=}\sum_{n=1}^{N}(FD_n - m)\frac{3}{(N-1)SD^3}$
4	*median (FD)*	$F_{meadin} = \left(\frac{N+1}{2}\right)^{th}$	8	*kurtosis (FD)*	$F_{ku} = \sum_{n=1}^{N}(FD_n - m)\frac{4}{(N-1)SD^4}$

minimum execution time. Many researchers have used the LS_SVM in EEG signals classification. It was used by Suily et al. [55] for the motor image classification, also by Al Ghayab et al. [6] for detecting the epileptic EEG signals.

The LS_SVM depends on two hyper parameters, γ and σ. The two parameters can positively or negatively affect the performance of the proposed method. It is necessary to choose those parameters carefully in order to obtain the desired classification results. In this study, the radial basis function (RBF) kernel was used, and the optimum values for γ and σ are set to $\gamma = 10$ and $\sigma = 0.5$, selected during the training session.

3.5.2. K-means

The K-means is widely used to classify data in various fields, such as biomedical signals, digital images and time series classification. It is generally known as a clustering algorithm [41,19]. The architecture of this classifier depends on dividing data into groups according to their similarities or differences among their elements. The K-means identifies the cluster center and other elements by reducing the squared errors based on an objective function. The main objective of using a clustering algorithm is, firstly, to identify the cluster center. Secondly, it associates each element which has the same characteristics with the nearest cluster center. In this paper, K-means is used to distinguish between sleep spindles and non spindles segments.

3.5.3. Neural network

The backpropagation algorithm of a neural network is a supervised learning algorithm. It is commonly used in classification research [10]. It was used by Bishop et al. [25] to classify k-complexes in sleep EEG signals. The connection weights in each iteration are updated. The architecture of a typical neural network consists of three layers, namely, an input layer, a hidden layer and the output layer. The input layer is fed with the input features. The second layer is a hidden layer. It has five neurons with an activation function of $y(x) = 1/(1 + e^{-\sigma x})$. The number of the hidden layer and neurons are determined empirically. The value of σ is set to 1. The number of iterations is set to 1000, the target error is set to 10e-5. The learning rate is set to 0.05.

3.5.4. Naïve bayes (NB)

Naïve Bayes is an efficient and effective technique for classification and it is commonly used in pattern recognition. It works based on the applications of Bayes' rules and posterior hypothesis. The Naïve Bayes assumes that each attribute influences differently on a given class. It has received a great attention from many researchers as it is simple and fast [4]. Puntumapon et al. [46] used a naïve classifier for classifying cellular phone mobility. Rakshit et al. [48] also employed this classifier to classify left and right movement patterns in EEG signals. In this paper, Naïve Bayes is also employed to detect sleep spindles.

3.6. Performance evaluation

The accuracy, sensitivity and specificity measurements are used to evaluate the performance of the proposed method to detect sleep spindles [64,51,65]. The main formulas of those statistical measurements are defined as.

Sensitivity (SEN) or true positive rate: It is used to estimate the performance of the classification method by measuring the proportion of the actual positive predication. It is defined as:

$$Sensitivity\ (SEN) = \frac{TP}{TP + FN} \tag{7}$$

where TP (true positive) means the actual sleep spindle waves that are correctly detected using the proposed method, FN (false negative) shows the actual sleep spindles that are incorrectly marked as non-sleep spindles.

Accuracy: it refers to the number of correctly classified cases. It is calculated by dividing the aggregating of classification results by the number of cases. The accuracy is defined as:

$$Accuracy\ (ACC) = \frac{TP + TN}{Total\ number\ of\ the\ cases} \tag{8}$$

where TN (true negative) is the actual non-sleep spindles that are correctly classified using the proposed method as non-sleep spindles

Specificity: it is used to calculate the proportion of the actual negative predication. It is defined as.

$$Specificity\ (SPE) = \frac{TN}{TN + FP} \tag{9}$$

where FP (false positive) refers to the number of sleep spindles that are incorrectly determined by the proposed method.

F-score*:* it is one of the most important measurements that are used to show the overlapping between the sets of true sleep spindles and the found sleep spindles by using the proposed method. F-score is defined as a harmonic mean of precision *(PPV)* and recall *(TPR):*

$$F - Score = 2x\frac{(PPV.TPR)}{PPV + TPR} \tag{10}$$

where PPV is precision or positive predictive value that is calculated as

$$Precision\ (PPV) = \frac{TP}{TP + FP} \tag{11}$$

Kappa coefficient: it measures the performance agreement between two models. It is defined as.

$$Cohen's\ Kappa\ Coefficient\ (k) = \frac{pr(a) - pr(e)}{1 - pr(e)} \tag{12}$$

where *pr(a)* and *pr(e)* represent the actual agreement and chance agreement respectively.

Table 5
The performance of the proposed method based on two features set.

Dataset-1				Dataset-2		
Fold	sensitivity%	specificity%	accuracy%	sensitivity%	specificity%	accuracy%
Fold1	73	75	78	39.2	87.6	81.9
Fold2	76	72	74	55	85	79
Fold3	74	76	76	77	81	77
Fold4	75.5	74	77	60	80	81
Fold5	72	73	75	76	83	80
Fold6	76	75	76.5	72	79	75
average	**74.41**	**74.16**	**76.03**	**63.2**	**82.6**	**78.9**

K-cross-validation: It is a popular measure to assess the classification accuracy. It is used to describe the performance of the proposed method. The dataset is divided into k equal subsets. One of them is used as the testing set, while the rest subsets are used as the training set. All the subsets are tested. The testing classification accuracy for all the subsets are calculated and recorded.

In this paper, $k = 6$ (6-cross-validation) is used. Therefore, the average accuracy is computed as below.

$$\text{Performance} = \frac{1}{6}\sum_{1}^{6} accuracy^{(R)} \tag{13}$$

where $accuracy^{(R)}$ is the accuracy for the 6 iterations.

Receiver Operating Characteristics (ROC): The ROC curve is a suitable metric in studying the dependency of sensitivity and specificity. The relationships among true positive rate, false negative rate, false positive rate and true negative were investigated in this paper using the ROC. The ROC curve represents by a graph in which the false positive rate is plotted on the x-axis while the true positive rate is plotted on the y-axis. The left lower point (0, 0) indicts the method dose not commit false positive errors and dose not obtain true positive rate, while the upper right point (1, 1) represents the opposite strategy. The perfect point in the ROC is represented by the point (0, 1).

4. Experimental results

In this study, the proposed method is developed to detect sleep spindles based on fractal dimension and time frequency image. All the experiments were conducted with the databases discussed in Section 2. The EEG signals were divided into segments using a sliding window technique. The size of the window was set to 0.5 s with an overlapping of 0.4s. Then, the EEG signal was, firstly, converted to a TFI using a STFT. Each TFI was converted into an 8-bit binary image. The fractal dimension based on the box-counting method was used to extract the desired features from each TFI. The obtained results showed that the extracted features using the box counting algorithm yielded accurate results. The experiments were conducted using Matlab software (Version: R2015) on a computer with the following settings: 3.40 GHz Intel(R) core(TM) i7 CPU processor machine, and 8 GB RAM.

The number of square boxes that were required to cover the entire curve is investigated in this paper. Eight statistical features were extracted after obtaining the FDs based on the box-counting method. Those features were considered as the key features, and were then forwarded to different classifiers of LS_SVM, K-means, Nave Bayes and a neural network. For further investigation and to evaluate the performance of the proposed method, different sets of features, including two, four, six and eight features, were used to detect sleep spindles. The results were discussed in the next section. According to the experimental results, the proposed method with eight features achieved high classification results, with an average accuracy of 98.6% for Dataset-1, and 97% for Dataset-2.

4.1. Two features set

Two features $\{F_{mean}, F_{max}\}$ were tested to detect sleep spindles in EEG signals. In this case four boxes of size 512 were considered in this experiment to extract the FDs. Each TFI was presented as a vector of two statistical features. According to the obtained results, two features set was not good enough to distinguish the sleep spindles with an acceptable accuracy. Table 5 reports the obtained results in term of accuracy, sensitivity and specificity in each fold based on the two features for the both datasets. The 6-cross validation was used in this paper. An average accuracy, sensitivity and specificity of 76%, 74.4% and 74.1% for Dataset-1, while the average accuracy, sensitivity and specificity of 78.9%, 63.2% and 82.6% for Dataset-2 were recorded. To obtain a higher accuracy, the number of features was increased to four features in the next experiment.

4.2. Four features set

Four features were also investigated. The four features of $\{F_{mean}, F_{max}, F_{meadin}, F_{SD}\}$ were extracted from the FDs of each TFI. In this experiment, 64 boxes of size 128 were considered to extract the estimates of the fractal dimensions for each TFI. The accuracy, sensitivity and specificity were increased by about 7% when the number of features was increased to four. Table 6 shows the obtained results for the both datasets.

We can notice that there are differences in the results when the number of features was increased. Because the box size and the number of boxes used to extract the FDs were covered most

Table 6
The performance of the proposed method based on four features set.

Dataset-1				Dataset-2		
Fold	sensitivity%	specificity%	accuracy%	sensitivity%	specificity%	accuracy%
Fold1	84	86	89	65.5	98.2	88
Fold2	81	84	88	75	90	86
Fold3	83	82.6	84.6	79	89	84
Fold4	80.5	84	85	81	91	87
Fold5	83	85.5	84	76	86	85
Fold6	82	87	86	72	84	84
average	**82.25**	**84.85**	**86.1**	**74.75**	**89.7**	**85.6**

Table 7
The performance of the proposed method based on six features set.

Dataset-1				Dataset-2		
Fold	sensitivity%	specificity%	accuracy%	sensitivity%	specificity%	accuracy%
Fold1	96	96	99	79.6	97.8	90.6
Fold2	95	98	95	74	92	91
Fold3	94.5	94	97	69	96	97
Fold4	97	95	96	80	97	93
Fold5	94	97	97	79	94	91
Fold6	93	98	99	78	96	90
average	**94.9**	**96.3**	**97**	**76.6**	**95.4**	**92.1**

Table 8
The performance of the proposed method based on eight features set.

Dataset-1				Dataset-2		
Fold	sensitivity%	specificity%	accuracy%	sensitivity%	specificity%	accuracy%
Fold1	97	97.6	99	93.7	99	98.1
Fold2	95	96	98	92	95	98
Fold3	96	97.5	99	96	97	97
Fold4	96	99	98.7	97	98	95
Fold5	98	98	99	99	99	99
Fold6	99	97	98	95	94	96
average	**96.8**	**97.5**	**98.6**	**95.4**	**97**	**97.1**

of the image region. Also, the proposed method provides a better performance by using the four features. The average accuracy, sensitivity and specificity of the proposed method with Dataset-1 is 86.1%, 82.25% and 84.85%, respectively. The average accuracy, sensitivity and specificity of 85.6%, 74.5% and 89.7% for Dataset-2 were recorded. Our finding showed that there were no big differences in results when the proposed method was evaluated with two different datasets. It is clear that, the proposed method achieved quite similar results using two different channels.

4.3. Six and eight features

In this case, a vector of six features including {F_{mean},F_{max}, F_{meadin}, F_{SD}, F_{min}, F_{Sk}} were extracted and used to detect sleep spindles. Table 6 shows the performance of the proposed method based on the six features with the both datasets. The experimental results showed that the classification performance by the six features set were better than that by the four features set with an increase of 8%. The proposed method was also tested with eight features of {F_{mean},F_{max}, F_{median}, F_{SD}, F_{min}, F_{Sk}, F_{rang}, F_{ku}}. It was noticed that the accuracy, sensitivity and specificity were slightly increased, but there were no big differences between using the six or eight features set. The performance results based on the eight features set by the proposed method with two datasets are presented in Table 8. Tables 7 and 8 show that the six and eight features sets using the two datasets yielded quite similar results. From the obtained results, it is clear that increasing the number of the features to eight can increase the performance of detecting sleep spindles. The obtained results demonstrated that the proposed method yielded the best performance with an average accuracy of 98.6% and 97.1% with Dataset-1 and Dataset-2, respectively. Figs. 4 and 5 demonstrate the classification accuracy against the number of the features. From Fig. 4 one can notice that the proposed scheme achieves better results with an average accuracy of 98% using six and eight features with Dataset-1. Fig. 5 shows the average accuracy of 97.1% using Dataset-2. The above results show that the proposed method has potentials to classify EEG signals for sleep spindles and non-spindles segments.

Fig. 6 presents the classification accuracy based on the number of the features for the both datasets. From Fig. 6, it was found that the eight features set yielded the best accuracy with the both

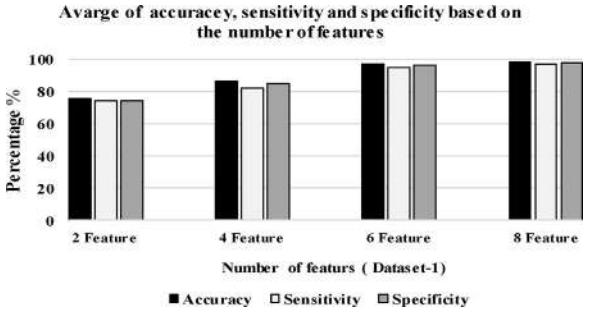


Fig. 4. The accuracy, sensitivity and specificity percentages with the number of the features for Dateset-1.

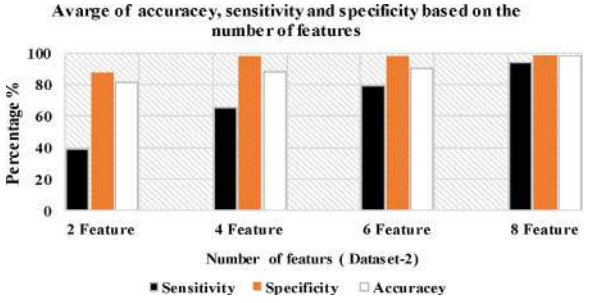


Fig. 5. The accuracy, sensitivity and specificity percentage with the number of the features for Dateset-2.

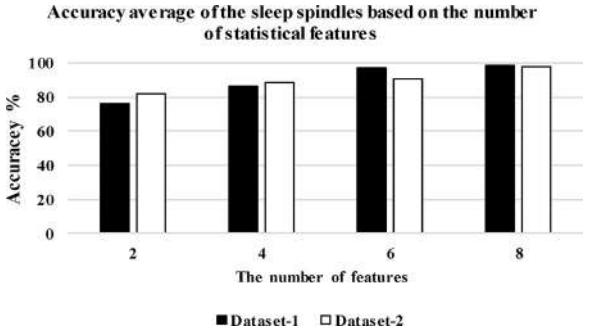


Fig. 6. Classification accuracy based on the number of the features.

Table 9
The performance of the proposed method based on F-score and Kappa coefficient.

Type of Database	Measurements	
	F-score	Kappa coefficient
Dataset-1	0.95	0.87
Dataset-2	0.89	0.83

databases when compared to the results with another features, including two, four, six and eight features sets. It was observed that there were relationships between the number of the features and the accuracy. Based on the above results, the proposed method obtained an average of 98.6% and 97.1% accuracy with eight features set for the two datasets, separately.

For further evaluation, the performance of the proposed scheme were also tested using different metrics, including F-score and kappa coefficient. Table 9 reports the average of measurements for the both datasets. The averages of F-score and kappa coefficient were 0.95% and 0.87% for Dataset-1, while the average of F-score and kappa coefficient were 0.89 and 0.83 for Dataset-2. All the results in Tables 5–9 were carried out using the LS_SVM classifier.

5. Comparison of study

To evaluate the proposed method, extensive experiments were conducted to detect sleep spindles in EEG signals. The extracted features were fed into the LS_SVM as well as to the neural network (NN), Naïve Bayes and K-means to evaluate the performance of the proposed method. We also compared the performance of the proposed method with other existing studies that used the same datasets as described in Section 2. Finally, the complexity time was computed to evaluate the speed of the proposed method in sleep spindles detection.

5.1. Comparison with different classifiers

In this section, the performance of the proposed method was compared using accuracy, sensitivity, specificity and F-score with different classifiers, including the LS_SVM, the NN, K-means and Naïve Bayes. Fig. 7 presents the results of the comparisons. Different numbers of segments were selected randomly from the two datasets. The eight statistical features set was considered in the comparisons.

Based on the obtained results in Fig. 7, the proposed method achieved better results with LS_SVM than the other classifiers. One can see that the best accuracy is 98.6% by the LS_SVM. Furthermore, the sensitivity and specificity with the same classifier are 96.8% and 97.5%, respectively. The second highest accuracy, sensitivity and specificity 94.6%, 93% and 94%, respectively, were recorded with K-means classifier. For further investigation in terms of the efficiency of the proposed method, F-score was assessed for all the classifiers. The proposed method with the LS_SVM classifier yielded the highest F-score value as 0.95.

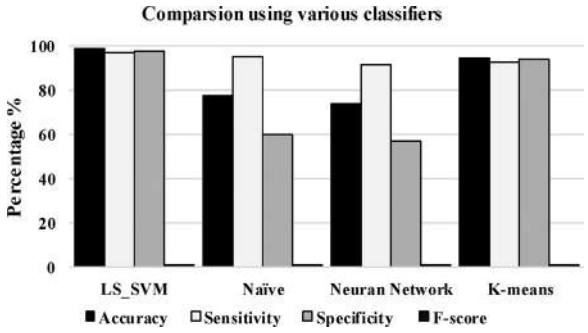


Fig. 7. The performance of the proposed method based on different classifiers.

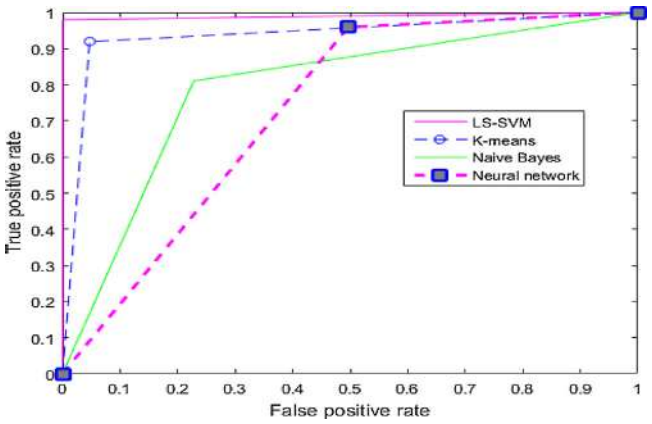


Fig. 8. The ROC curves for the four classifiers.

The evaluation results for all the classifiers were supported by using Receiver Operating Characteristics (ROC) curve. From Fig. 8, it is clear that the biggest area under the ROC curve was constructed by LS_SVM. The second best results was obtained with K-means classifier, and the area under the ROC curve for artificial neural network was generally lower than other classifiers. The results of the area under the curve of 0.98 reported were from LS_SVM classifier. From those results, it was evidence that the LS_SVM was the best classifier for detecting sleep spindles in EEG signals.

5.2. Performance evaluation based on time complexity

Comparisons in terms of the time complexity were made for the both datasets. Fig. 9 shows the results of the comparisons based on the number of segments and complexity time for each classifier. Different numbers of segments were used to compare our proposed method using those classifiers. All the segments were randomly selected from the both datasets. According to the results, the performance of the proposed method with the LS_SVM is better than those by other classifiers. The minimum time execution of the proposed methods with the LS_SVM was 0.41second(s) when dealing with 100 segments. In addition, one can see that the time slightly increased when the number of the segments was between 200 and 500. On the other hand, the maximum time of the proposed method increased to a record of 10.0 s with 1500 segments. From Fig. 9, it is clear that the longest time was consumed by the NN classifier.

5.3. Comparison with histogram representation

Form the literature, the previous studies that used a time frequency image to extract features from EEG signals were developed

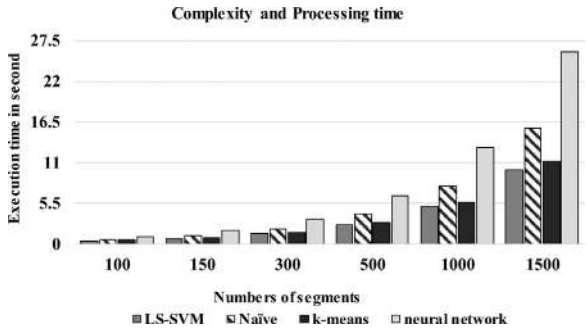


Fig. 9. Time complexity comparisons with different numbers of segments from the both datasets.

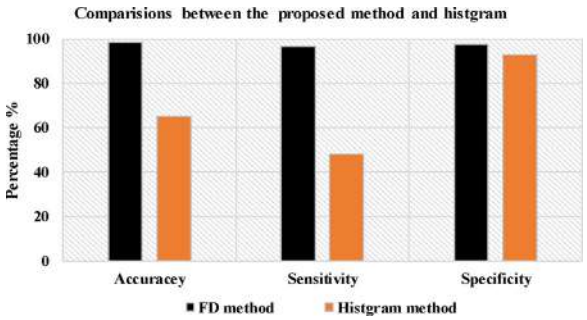


Fig. 10. Comparison of the performance of the proposed method based on histogram and fractal dimension features.

based on analyzing the characteristics of histograms. The characteristics of the TFI histograms were used by Bajaj and Pachori [9] to identify EEG sleep stages and to classify alcoholic EEG signals. They were also used by Fu et al. in [22] to detect epileptic seizures in EEG signals. In this paper, the characteristics of the TFI histograms were also investigated. The obtained results were compared with those by the fractal dimension technique. In this experiment, the histogram features were extracted from each TFI and the performance of the proposed method was evaluated in terms of accuracy, sensitivity and specificity. All experiments were made with the both datasets. Eight statistical features of $\{F_{mean}, F_{max}, F_{median}, F_{SD}, F_{min}, F_{Sk}, F_{rang}, F_{ku}\}$ were extracted from each TFI histogram. The extracted features were fed to the LS_SVM. From Fig. 10, we can notice that the maximum accuracy, sensitivity and specificity obtained by the histogram dimensional features were 65%, 48% and 93%. It is clear that the obtained results are lower than those obtained by fractal dimension. Based on the results, extracted features from TFIs using the box counting method achieved high classification rates than the histogram features.

5.4. Comparison with the existing methods

To evaluate the performance of the proposed methods, the comparisons with other existing algorithms were made. All the selected studies were conducted using the same databases as described in Section 2. Table 10 shows the comparisons among the proposed method and those from Nonclercq et al. [36], Imtiaz et al. [30], Patti et al. [43], Ahmed et al. [5], Gorur et al. [23] and Devuyst, S., et al. [15], Kuriakose, et al. [33], Zhuang and Peng [65], Patti et al. [44] and Yucelbas et al. [64]. Those studies were conducted using Dataset-1. The proposed method was also compared with some studies in which Dataset-2 was used. The comparison studies were made by Tsanas et al. [57], Saifutdinova et al. [51] and Patti et al. [44]. Based on the results in Table 10, the proposed method yields the best results comparing with the others.

Nonclercq et al. [36] used a 0.5 s window size with an overlapping 125 ms. The maximum sensitivity and specificity in that study were 75.1% and 94%, respectively. The obtained results using the proposed methods were better than those by Nonclercq et al. [36]. Another study was presented by Devuyst, et al. [15]. The used window in that study was 0 0.5 s with an overlapping 0.1 ms. The maximum sensitivity and specificity achieved were 70.20% and 98%, respectively. We can observe that the proposed method performed better than those by Devuyst, S., et al. [15].

Table 10
The performance comparisons of the proposed method with other existing methods.

Authors	Method	ACC (%)	SEN (%)	SPE (%)	F-score(%)	KA(%)	WS (s)	OVR (s)	DB
Nonclercq et al.	Sleep spindles detection using amplitude based features extraction	–	–	94	–	–	0.5	0.25	DB-1
Imtiaz et al.	Teager energy and spectral edge frequency	91	80	96	–	–	0.25	50	DB-1
Devuyst et al.	a systematic assessment method	–	80.3	97.6	–	–	0.5	–	DB-1
Imtiaz et al.	Line length	–	83.6	87.9	–	–	1.0	50	DB-1
Kuriakose, et al.	Transformation coefficients as known Karhunen-level transform	–	86.9	93.5	–	–	0.5	–	DB-1
Gorur et al.	a short time Fourier transform with a SVM and neural network		95.4 88.7	–	–	–	0.5	–	DB-1
Ahmed et al.	Wavelets packets Energy Ratio and Teager Energy Operator	93.7	–	–	–	–	1.28	–	DB-1
Tsanas et al.	Continuous wavelet transform with Morlet basis function		76	92	0.46	0.66	1.0	–	DB-2 DB-1
Patti et al.	Gaussian mixture model	–	74.9	–	–	–	1.5	–	DB-1
Zhuang and Peng	Utilize a sliding window- based probability estimation method		50	99	0.58		1.0	50	DB-1
Patti et al	Random Forest classifier		71.2	96.73	–	–	–	–	DB-2
Yucelbas et al.	Fast Fourier transform, autoregressive, multiple signal classification and Welch filter	84 without PCA							Private
Yucelbas et al.	Fast Fourier transform, autoregressive, multiple signal classification and Welch filter	94with PCA	–	–	–	–	–	–	Private
Parekh et al	Optimization algorithm for the detection k-complex and sleep spindles.	96	71	96	0.69	0.67	1.0	75	DB-1
Saifutdinova et al.	used an empirical mode decomposition to detect sleep spindles in EEG signals	–	–	–	0.40	–	–	–	DB-2
Proposed method	**Time frequency image based on fractal dimension**	**98.6**	**96.8**	**98.2**	**0.95**	**0.87**	**0.5**	**0.4**	DB-1
Proposed method	**Time frequency image based on fractal dimension**	**97.1**	**95.4**	**97**	**0.89**	**0.83**	**0.5**	**0.4**	DB-2

where ACC = accuracy, SEN = sensitivity, SPE = specificity, KA = kappa coefficient, WE = window size, OVR = an overlapping, s=second and DB = dataset.

Imtiaz et al. [30] reported a Teager energy and spectral edge frequency method to detect sleep spindles. In this study, a window size of 0.25 s with an overlapping of 50% was considered, and over 91% of sleep spindles were detected correctly. However, the proposed method archived a 98.6% accuracy, higher than the method by Imtiaz et al. [30].

Patti et al. [43] used a Gaussian mixture model to identify sleep spindles in EEG signals. A window size of 1.5 s without overlapping was employed. Four features were extracted and forwarded to a classifier to detect sleep spindles. An average sensitivity of 74.9% was reported. In comparison, the proposed method achieved more than 96.8% sensitivity with a 0.5 window size. Most recently, Ahmed et al. [5] introduced a wavelet packet transform and Teager energy operator algorithm for detecting sleep spindles. A window of 1.28 s without overlapping was considered. From the results in Table 9, we can see that the proposed method yielded a better classification accuracy, comparing to those by Patti et al. [43] and Ahmed et al., [5].

Gorur et al. [23] used a short time Fourier transform to distinguish EEG sleep spindles. They used the same window size of 0.5 s without overlapping. In that study, the maximum average of sensitivity using the SVM and the NN was 95.4% and 88.7%, respectively. According to the results, one can see that the sensitivity of 96.8% by the proposed method is higher than those by Gorur et al. [23]. Another study presented by Zhuang and Peng [64], in which the sleep spindles were detected based on a sliding window-based probability estimation method. An EEG signal was passed through a Mexican hat wavelet transform. A set of wavelet coefficients were employed. A window size of 1.0 s with an overlapping of 50% was used in that study. An average 50.98% sensitivity and 99% specificity were reported. Although the average specificity in that study was higher than our proposed method, but we achieved the highest sensitivity and accuracy of 96.5% and 97.9%, respectively, comparing with those by Zhuang and Peng [64].

Tsanas et al. [57] detected sleep spindles based on a continuous wavelet transform and local weighted smoothing. The paper reported a sensitivity and specificity of 76% and 92%, respectively. It is clear that the proposed method achieved better accuracy, sensitivity and specificity compared with the existing methods. Patti et al. [44] applied a Random Forest classifier to detect sleep spindles. Three channels in the central EEG signals, including CZ, C3 and C4, were utilized for detecting sleep spindles. A window size of 0.5 s without overlapping was used in that study. Three features of Alpha Ratio, Sigma index and spindle band ratio were employed for the detection. The maximum sensitivity and specificity of 71.2% and 96.73% were reported, respectively.

Another study presented by Saifutdinova et al. [51] used an empirical mode decomposition to detect sleep spindles in EEG signals. The average F-score in that study was 40.72%, and 48.59%. The proposed method obtained a high classification F-score compared with the results presented by Saifutdinova et al. [51]. The proposed method was also compared with other methods in which different datasets were used.

Yucelbas et al. [64] presented the sleep spindles detection results using a fast Fourier transform, autoregressive, multiple signal classification and Welch filter. The detection phase was carried out by a NN classifier. An average accuracy of 84.8% was reported. In that study, the results changed when a principle component analysis was used. The maximum accuracy was 94%. The proposed method performed much better than those by Yucelbas et al. [64]. In summary, the comparisons with the previous studies showed that using time frequency image based on fractal dimension is effective and suitable to detect sleep spindles in EEG signals.

6. Conclusion

In this paper, a new method to detect sleep spindles in EEG signals was presented. The proposed method applied time frequency image and fraction dimension techniques to detect sleep spindles with a high classification accuracy and low execution time.

A window size of 0.5 s with an overlapping of 0.4 s was adopted in this study. The EEG signals were converted into time frequency images by using spectrogram of a short time Fourier transform. A box counting algorithm was applied to calculate the fractal dimensions (FDs) from each TFI. Eight statistical features were extracted from each FD. Those features were passed to different classifiers, including the least square support vector machine, K-means, neural network, Naïve Bayes classifiers to figure out the best classification method to detect sleep spindles. The best results of 98.6% accuracy, 96.8% sensitivity and 97.5% specificity were achieved with Dateset-1. It was found that using the TFI with the fractional dimension can improve the detection of sleep spindles. The outcomes of this study can help sleep experts to efficiently analyse EEG signals. In the future work, we will apply the proposed method to detect K-complexes in EEG signals.

References

[1] N. Acır, C. Güzeliş, Automatic spike detection in EEG by a two-stage procedure based on support vector machines, Comput. Biol. Med. 34 (2004) 561–575.
[2] N. Acır, C. Güzeliş, Automatic recognition of sleep spindles in EEG by using artificial neural networks, Expert Syst. Appl. 27 (2004) 451–458.
[3] N. Acır, C. Güzeliş, Automatic recognition of sleep spindles in EEG via radial basis support vector machine based on a modified feature selection algorithm, Neural Comput. Appl. 14 (2005) 56–65.
[4] M. Ahangi, N. Karamnejad, R. Mohammadi, N. Ebrahimpour, Multiple classifier system for EEG signal classification with application to brain–computer interfaces, Neural Comput. Appl. 23 (2013) 1319–1327.
[5] B. Ahmed, A. Redissi, R. Tafreshi, An automatic sleep spindle detector based on wavelets and the Teager energy operator, in: 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE, 2009, pp. 2596–2599.
[6] H.R. Al Ghayab, Y. Li, S. Abdulla, M. Diykh, X. Wan, Classification of epileptic EEG signals based on simple random sampling and sequential feature selection, Brain Inform. 3 (2016) 85–91.
[7] Z. Ali, I. Elamvazuthi, M. Alsulaiman, G. Muhammad, Detection of voice pathology using fractal dimension in a multiresolution analysis of normal and disordered speech signals, J. Med. Syst. 40 (2016) 1–10.
[8] Y. Bajaj, A. Guo, S. Sengur, O.F. Siuly, A hybrid method based on time?frequency images for classification of alcohol and control EEG signals, Neural Comput. Appl. (2016) 1–7.
[9] V. Bajaj, R.B. Pachori, Automatic classification of sleep stages based on the time-frequency image of EEG signals, Comput. Methods Progr. Biomed. 112 (2013) 320–328.
[10] I.S. Baruch, V.A. Quintana, E.P. Reynaud, Complex-valued neural network topology and learning applied for identification and control of nonlinear systems, Neurocomputing (2016).
[11] T.A. Camilleri, K.P. Camilleri, S.G. Fabri, Automatic detection of spindles and K-complexes in sleep EEG using switching multiple models, Biomed. Signal Process. Control 10 (2014) 117–127.
[12] A. Castelnovo, A. D'Agostino, C. Casetta, S. Sarasso, F. Ferrarelli, Sleep spindle deficit in schizophrenia: contextualization of recent findings, Curr. Psychiatry Rep. 18 (2016) 1–10.
[13] D. Coppieters't Wallant, P. Maquet, C. Phillips, Sleep spindles as an electrographic element: description and automatic detection methods, Neural. Plasticity 2016 (2016) 6783812, http://dx.doi.org/10.1155/2016/6783812.
[14] J.C. da Costa, M.D. Ortigueira, A. Batista, K-means clustering for sleep spindles classification, Int. J. Inf. Technol. Comput. Sci. (2091–1610) 10 (2013) 77–85.
[15] S. Devuyst, T. Dutoit, P. Stenuit, M. Kerkhofs, Automatic sleep spindles detection—overview and development of a standard proposal assessment method, in: 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE, 2011, pp. 1713–1716.
[16] M. Diykh, Y. Li, P. Wen, EEG sleep stages classification based on time domain features and structural graph similarity, IEEE Trans. Neural Syst. Rehabil. Eng. 24 (2016) 1159–1168.
[17] F. Duman, A. Erdamar, O. Erogul, Z. Telatar, S. Yetkin, Efficient sleep spindle detection algorithm with decision tree, Expert Syst. Appl. 36 (2009) 9980–9985.
[18] P.A. Estévez, R. Zilleruelo-Ramos, R. Hernández, L. Causa, C.M. Held, Sleep spindle detection by using merge neural gas, in: 6th Int WSOM, Bielefeld, Germany, 2007.

[19] K. Faraoun, A. Boukelif, Neural networks learning improvement using the k-means clustering algorithm to detect network intrusions, World Acad. Sci. Eng. Technol. Int. J. Comput. Electr. Automa. Control. Inf. Eng. 1 (2007) 3138–3145.
[20] R. Ferrarelli, M.J. Huber, M. Peterson, M. Massimini, B.A. Murphy, A. Riedner, P. Watson, G. Bria, Reduced sleep spindle activity in schizophrenia patients, Am. J. Psychiatry 164 (3) (2007) 483–492.
[21] F. Finotello, F. Scarpa, M. Zanon, EEG signal features extraction based on fractal dimension, in: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC, IEEE, 2015, pp. 4154–4157.
[22] K. Fu, J. Qu, Y. Chai, Y. Dong, Classification of seizure based on the time-frequency image of EEG signals using HHT and SVM, Biomed. Signal Process. Control 13 (2014) 15–22.
[23] U. Gorur, H. Halici, G. Aydin, F. Ongun, K. Ozgen, Sleep spindles detection using short time Fourier transform and neural networks, Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on IEEE (2002) 1631–1636.
[24] S. Güneş, M. Dursun, K. Polat, Ş. Yosunkaya, Sleep spindles recognition system based on time and frequency domain features, Expert Syst. Appl. 38 (2011) 2455–2461.
[25] E. Hernández-Pereira, V. Bolón-Canedo, N. Sánchez-Maroño, D. Álvarez-Estévez, V. Moret-Bonillo, A. Alonso-Betanzos, A comparison of performance of K-complex classification methods using feature selection, Inform. Sci. 328 (2016) 1–14.
[26] E. Hernandez-Pereira, I. Fernandez-Varela, V. Moret-Bonillo, A comparison of performance of sleep spindle classification methods using wavelets, in: Innovation in Medicine and Healthcare 2016, Springer, 2016, pp. 61–70.
[27] E. Huupponen, G. Gómez-Herrero, A. Saastamoinen, A. Värri, J. Hasan, S.-L. Himanen, Development and comparison of four sleep spindle detection methods, Artif. Intell. Med. 40 (2007) 157–170.
[28] C. Iber, The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications, American Academy of Sleep Medicine, 2007.
[29] E. Imtiaz, Evaluating the use of line length for automatic sleep spindle detection, in: 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE, 2014, pp. 5024–5027.
[30] S.A. Imtiaz, S. Saremi-Yarahmadi, E. Rodriguez-Villegas, Automatic detection of sleep spindles using Teager energy and spectral edge frequency, in: 2013 IEEE Biomedical Circuits and Systems Conference (BioCAS), IEEE, 2013, pp. 262–265.
[31] M.M. Kabir, R. Tafreshi, D.B. Boivin, N. Haddad, Enhanced automated sleep spindle detection algorithm based on synchrosqueezing, Med. Biol. Eng. Comput. 53 (2015) 635–644.
[32] Y.W. Kim, K.K. Krieble, C.B. Kim, J. Reed, A.D. Rae-Grant, Differentiation of alpha coma from awake alpha by nonlinear dynamics of electroencephalography, Electroencephalogr. Clin. Neurophysiol. 98 (1996) 35–41.
[33] S. Kuriakose, G. Titus, Karhunen-loeve transform for sleep spindle detection, Devices, Circuits and Systems (ICDCS) 2016 3rd International Conference on, IEEE (2016) 249–253.
[34] Y. Li, P.P. Wen, Clustering technique-based least square support vector machine for EEG signal classification, Comput. Methods Progr. Biomed. 104 (2011) 358–372.
[35] M. Nishida, Y. Nakashima, T. Nishikawa, Slow sleep spindle and procedural memory consolidation in patients with major depressive disorder, Nat. Sci. Sleep 8 (2016) 63.
[36] A. Nonclercq, C. Urbain, D. Verheulpen, C. Decaestecker, P. Van Bogaert, P. Peigneux, Sleep spindle detection through amplitude?frequency normal modelling, J. Neurosci. Methods 214 (2013) 192–203.
[37] W. Nunsong, K. Woraratpanya, Modified differential box-counting method using weighted triangle-box partition, in: 2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE), IEEE, 2015, pp. 221–226.
[38] H. Ocak, Optimal classification of epileptic seizures in EEG using wavelet analysis and genetic algorithm, Signal Process. 88 (2008) 1858–1867.
[39] E. Olbrich, P. Achermann, Oscillatory events in the human sleep EEG—detection and properties, Neurocomputing 58 (2004) 129–135.
[40] C. O'reilly, N. Gosselin, J. Carrier, T. Nielsen, Montreal Archive of Sleep Studies: an open-access resource for instrument benchmarking and exploratory research, J. Sleep Res. 23 (2014) 628–635.
[41] U. Orhan, M. Hekim, M. Ozer, EEG signals classification using the K-means clustering and a multilayer perceptron neural network model, Expert Syst. Appl. 38 (2011) 13475–13481.
[42] A. Parekh, I.W. Selesnick, D.M. Rapoport, I. Ayappa, Detection of K-complexes and sleep spindles (DETOKS) using sparse optimization, J. Neurosci. Methods 251 (2015) 37–46.
[43] C.R. Patti, R. Chaparro-Vargas, D. Cvetkovic, Automated Sleep Spindle detection using novel EEG features and mixture models, in: 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE, 2014, pp. 2221–2224.
[44] C.R. Patti, S.S. Shahrbabaki, C. Dissanayaka, D. Cvetkovic, Application of random forest classifier for automatic sleep spindle detection, in: Biomedical Circuits and Systems Conference (BioCAS) 2015 IEEE, IEEE, 2015, pp. 1–4.
[45] M.D. Prieto, A.G. Espinosa, J.-R.R. Ruiz, J.C. Urresty, J.A. Ortega, Feature extraction of demagnetization faults in permanent-magnet synchronous motors based on box-counting fractal dimension, IEEE Trans. Ind. Electron. 58 (2011) 1594–1605.
[46] K. Puntumapon, W. Pattara-Atikom, Classification of cellular phone mobility using Naive Bayes model, in: Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE, IEEE, 2008, pp. 3021–3025.
[47] B. Raghavendra, N.D. Dutt, Computing fractal dimension of signals using multiresolution box-counting method, Int. J. Inf. Math. Sci. 6 (2010) 50–65.
[48] A. Rakshit, A Naïve Bayesian approach to lower limb classification from EEG signals, in: Control, Instrumentation, Energy & Communication (CIEC), 2016 2nd International Conference on, IEEE, 2016, pp. 140–144.
[49] A. Rechtschaffen, A. Kales, A Manual of Standardized Terminology, Techniques and Scoring System for Sleep Stages of Human Subjects, Government Printing Office, Washington, D.C, 1968.
[50] D. Ristanović, B.D. Stefanović, N. Puškaš, Fractal analysis of dendrite morphology using modified box-counting method, Neurosci. Res. 84 (2014) 64–67.
[51] E. Saifutdinova, V. Gerla, L. Lhotska, J. Koprivova, P. Sos, Sleep spindles detection using empirical mode decomposition, in: Computational Intelligence for Multimedia Understanding (IWCIM), 2015 International Workshop on, IEEE, 2015, pp. 1–5.
[52] S.V. Schönwald, L. Emerson, R. Rossatto, M.L. Chaves, G.J. Gerhardt, Benchmarking matching pursuit to find sleep spindles, J. Neurosci. Methods 156 (2006) 314–321.
[53] B. Şen, M. Peker, A. Çavuşoğlu, F.V. Çelebi, A comparative study on classification of sleep stage based on EEG signals using feature selection and classification algorithms, J. Med. Syst. 38 (2014) 18.
[54] R.K. Sinha, Artificial neural network and wavelet based automated detection of sleep spindles, REM sleep and wake states, J. Med. Syst. 32 (2008) 291–299.
[55] S. Siuly, Y. Li, Designing a robust feature extraction method based on optimum allocation and principal component analysis for epileptic EEG signal classification, Comput. Methods Programs Biomed. 119 (2015) 29–42.
[56] O. Sourina, Y. Liu, A fractal-based algorithm of emotion recognition from EEG using arousal-Valence model, in: Biosignals, 2011, pp. 209–214.
[57] A. Tsanas, G.D. Clifford, Stage-independent, single lead EEG sleep spindle detection using the continuous wavelet transform and local weighted smoothing, Front. Hum. Neurosci. 9 (2015) 181.
[58] E.M. Ventouras, E.A. Monoyiou, P.Y. Ktonas, T. Paparrigopoulos, D.G. Dikeos, N.K. Uzunoglu, C.R. Soldatos, Sleep spindle detection using artificial neural networks trained with filtered time-domain EEG: a feasibility study, Comput. Methods Progr. Biomed. 78 (2005) 191–207.
[59] E.J. Wamsley, M.A. Tucker, A.K. Shinn, K.E. Ono, S.K. McKinley, A.V. Ely, D.C. Goff, R. Stickgold, D.S. Manoach, Reduced sleep spindles and spindle coherence in schizophrenia: mechanisms of impaired memory consolidation? Biol. Psychiatry 71 (2012) 154–161.
[60] S.C. Warby, S.L. Wendt, P. Welinder, E.G. Munk, O. Carrillo, H.B. Sorensen, P. Jennum, P.E. Peppard, P. Perona, E. Mignot, Sleep-spindle detection: crowdsourcing and evaluating performance of experts, non-experts and automated methods, Nat. Methods 11 (2014) 385–392.
[61] B. Weiss, Z. Clemens, R. Bódizs, P. Halász, Comparison of fractal and power spectral EEG features: effects of topography and sleep stages, Brain Res. Bull. 84 (2011) 359–375.
[62] H. Xiao, W. Zhi-zhong, R. Xiao-mei, Classification of surface EMG signal with fractal dimension, J. Zhejiang Univ. Sci. B 6 (2005) 844–848.
[63] J. Yang, Y. Zhang, Y. Zhu, Intelligent fault diagnosis of rolling element bearing based on SVMs and fractal dimension, Mech. Syst. Sig. Process. 21 (2007) 2012–2024.
[64] C. Yücelbaş, Ş. Yücelbaş, S. Özşen, G. Tezel, S. Küççüktürk, Ş. Yosunkaya, Automatic detection of sleep spindles with the use of STFT, EMD and DWT methods, Neural Comput. Appl. (2016) 1–17.
[65] X. Zhuang, Y. Li, N. Peng, Enhanced automatic sleep spindle detection: a sliding window-based wavelet analysis and comparison using a proposal assessment method, in: Applied Informatics, Springer, Berlin Heidelberg, 2016, pp. 11.
[66] J. Żygierewicz, K.J. Blinowska, P.J. Durka, W. Szelenberger, S. Niemcewicz, W. Androsiuk, High resolution study of sleep spindles, Clin. Neurophysiol. 110 (1999) 2136–2147.

Appendix B

An Efficient DDoS Flood Attack Detection and Prevention in a Cloud Environment

The main contribution in this paper represents in analysing the results in Figures 8-10 and Tables 4-7.

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 16, 2017, accepted March 19, 2017, date of publication April 6, 2017, date of current version May 17, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2688460

An Efficient DDoS TCP Flood Attack Detection and Prevention System in a Cloud Environment

AQEEL SAHI[1,2], DAVID LAI[2], YAN LI[2], (Member, IEEE), AND MOHAMMED DIYKH[1,2]

[1]Thi-Qar University, Nasiriya 64001, Iraq
[2]School of Agricultural, Computational and Environmental Sciences, University of Southern Queensland, Toowoomba, QLD 4350, Australia

Corresponding author: A. Sahi (akeel_sahy@yahoo.co.uk)

ABSTRACT Although the number of cloud projects has dramatically increased over the last few years, ensuring the availability and security of project data, services, and resources is still a crucial and challenging research issue. Distributed denial of service (DDoS) attacks are the second most prevalent cybercrime attacks after information theft. DDoS TCP flood attacks can exhaust the cloud's resources, consume most of its bandwidth, and damage an entire cloud project within a short period of time. The timely detection and prevention of such attacks in cloud projects are therefore vital, especially for eHealth clouds. In this paper, we present a new classifier system for detecting and preventing DDoS TCP flood attacks (CS_DDoS) in public clouds. The proposed CS_DDoS system offers a solution to securing stored records by classifying the incoming packets and making a decision based on the classification results. During the detection phase, the CS_DDOS identifies and determines whether a packet is normal or originates from an attacker. During the prevention phase, packets, which are classified as malicious, will be denied to access the cloud service and the source IP will be blacklisted. The performance of the CS_DDoS system is compared using the different classifiers of the least squares support vector machine (LS-SVM), naïve Bayes, K-nearest, and multilayer perceptron. The results show that CS_DDoS yields the best performance when the LS-SVM classifier is adopted. It can detect DDoS TCP flood attacks with about 97% accuracy and with a Kappa coefficient of 0.89 when under attack from a single source, and 94% accuracy with a Kappa coefficient of 0.9 when under attack from multiple attackers. Finally, the results are discussed in terms of accuracy and time complexity, and validated using a K-fold cross-validation model.

INDEX TERMS Classification, cloud computing, DDoS attacks, LS-SVM.

I. INTRODUCTION

Distributed denial of service (DDoS) TCP flood attacks are DoS attacks in which attackers flood a victim machine with packets in order to exhaust its resources or consume bandwidth [1]. As the attack may be distributed over multiple machines, it will be very hard to differentiate authentic users from attackers. In fact, a DDoS flood attack is not only a widespread attack; it is the second most common cybercrime attack to cause financial losses [2] according to the United States Federal Bureau of Investigation (FBI).

The use of cloud computing is quickly increasing in many sectors, and especially in the health sector, as a result of its vital features, such as availability and on-demand services [3]. Most people think of cloud computing as virtual network which can offer flexible and accessible on-demand services [4]. However, the author in [5] pointed out that cloud computing involves much more than this, which has led researchers to re-consider its security more seriously. In addition, as mentioned in an electronic cybercrime study published by KPMG in collaboration with eCrime Congress in 2009, most of the cloud's virtual clients are under threat, and these threats increase as time passes [6].

There are many procedures [7] which can be adopted to mitigate the DDoS flood attacks, such as classifications [8], [9], encryption techniques [10]–[12]. As DDoS flood attacks can be implemented in many forms, the form of these attacks cannot be foreseen. Therefore, our new proposed classifier system for the detection and prevention of DDoS TCP flood attacks (CS_DDoS) is classification based, and can identify these attacks data regardless of the form in which they arrive at the cloud system. Classification can be defined as a common procedure for classifying, distinguishing and

differentiating multiple objects. Different classifiers, such as least squares support vector machine (LS-SVM), naïve Bayes, K-nearest and multilayer perceptron [13], [14] are used in this study to perform the classification process.

This paper is organized as follows: in Section 2, we review related work, and Section 3 introduces the simulation platform, with and without DDoS TCP flood attacks. Section 4 presents our proposed CS_DDoS system, and its performance is evaluated and validated in Section 5. Finally, we conclude this study and discuss future work in Section 6.

II. RELATED WORK

Many detection and prevention methods for mitigating DDoS flood attacks have been reported in the last few years [15].

The rank correlation-based detection (RCD) scheme was proposed by Wei et al. in [16]. The authors of the RCD claimed that their scheme could distinguish whether the incoming requests were from genuine users or from attackers. In [17], the ALPi algorithm was introduced, which decreased difficulties in packet flows and improved functionality by extending the concept of packet scoring. The ALPi therefore raises the detection accuracy percentage and attack recognition. Another DDoS attack prevention architecture, known as secure overlay services (SOS), was presented in [18]. The SOS architecture is a combination of three parts: secure overlay tunneling, routing via consistent hashing, and filtering. The authors claimed that the SOS can successfully decrease the probability of these attacks using filtering close to the secure edge and randomness close to the front edge.

Moreover, Wang and Reiter proposed the web referral architecture for privileged service (WRAPS) [19]. The WRAPS adopted the structure of a web graph to resist DDoS flood attacks, and requires authentic users to be authenticated using a referral hyperlink from a trusted site. Another approach was introduced to detect application DoS attacks on backend servers called the group testing-based approach [20]. The authors extended the existing group testing approach by reallocating users' requests to several servers. Markov Chain probability theory was adopted by Salah et al. when proposing an analytical queuing approach which examines the performance of firewalls under DDoS attacks [21].

In addition, Dou et al. [22] presented a confidence-based filtering (CBF) scheme for cloud projects. In the CBF, packets of information from authentic users is gathered during non-attack periods to extract features, which can generate an information profile of these non-attack periods. With this profile, the CBF scheme will be endorsed using a packet-scoring calculation during attacks to make a decision on whether to remove these packets or not. Another approach to detecting flood attacks, the fast lightweight detection approach, was presented by Yu et al. [23]. This approach utilized SNMP-MIB (simple network management protocol-management information base) statistical data as an alternative to raw data, as well as a SVM classifier for attack classification. Lee et al. [24] introduced a practical DDoS detection scheme based on DDoS architecture. In this scheme, they selected variables based on particular features that were extracted from a DDoS architecture. A cloud trace back (CTB) method was proposed in [25]. The authors of the CTB claimed that their method could identify the sources of the attacks. They also proposed a cloud Protector (CP), which made use of a back-propagation classifier in order to detect such attacks.

Furthermore, a new framework was presented by Lu et al. in [26]. This framework was able to effectively identify compromised packets. It analyzed these packets at the router end using a perimeter-based DDoS prevention system. Wang et al. introduced a graphics-based DDoS attack prevention and detection scheme, which was able to work with the data shift issue [7]. This scheme works by prevention, using network monitoring and a precise response with an elastic control structure. In [27], an adaptive selective verification (ASV) system was proposed. The ASV does not rely on network assumptions, and utilizes bandwidth efficiently. Another approach was presented based on five features (average number of packets per flow, percentage of correlative flow, one-direction generating speed, ports generating speed, and percentage of abnormal packets) combined with a Bloom filter [28]. In this approach, only users on the whitelist are allowed to reach their destinations; this whitelist is generated to include legitimate users only. However, this approach was implemented on the switches side (i.e. in hardware), which makes any future amendments or updates challenging [29].

While many mechanisms have been proposed to detect and prevent DDoS flood attacks, most of these do not provide high accuracy and are not efficient or fast detection and prevention techniques [30]. Furthermore, many of the DDoS attack protection mechanisms described here face scalability issues due to the fact that networks are becoming larger and faster; in addition, industrial deployment needs to be considered [17].

Therefore, cloud computing needs an efficient DDoS mitigation approach that can offer fast and accurate detection while remaining scalable. The proposed CS_DDoS was designed with all of these factors in mind.

III. DDoS TCP FLOOD ATTACKS

DDoS attacks can be established in two different ways: either directly and/or indirectly [31], [32]. Direct attacks target a weakness in the system of the victim machines and damage the machines directly. On the other hand, indirect attacks do not target victim machines directly; they prey on other elements with which the victim machines are associated and hinder their work [33]. In the following discussion, the TCP flood attack is used; this is an indirect attack, as it consumes most of the network's resources, meaning that they are not readily available to other users.

A TCP flood attack was carried out using software on a virtual cloud network; Wireshark Network Analyzer 2.0.0 [34] was used to capture and analyze traffic both before and during the attack.

A. BEFORE THE ATTACK

The network was simulated as shown in Figure 1.

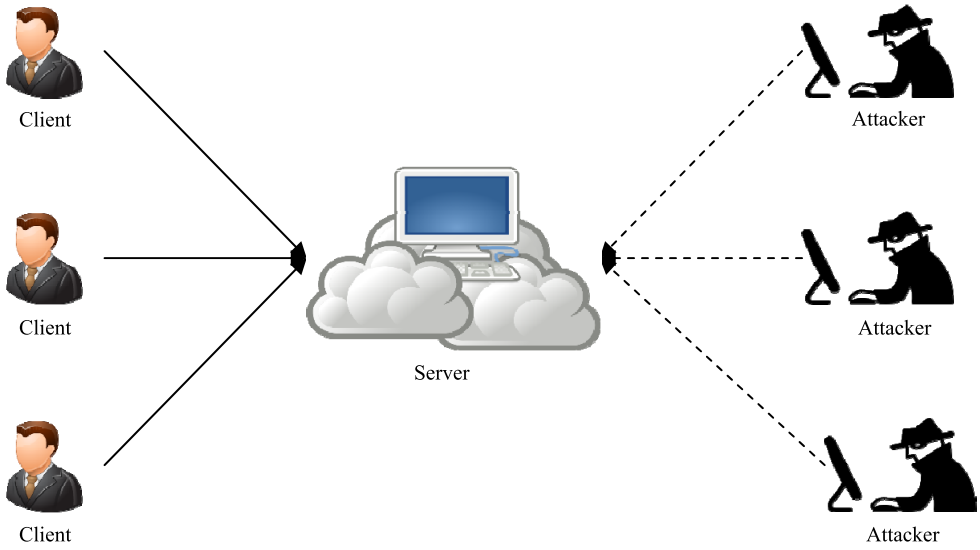


FIGURE 1. Test network architecture.

Firstly, using TCP Ping, we sent 50 TCP test probes (pings) to a server (server machine 10.25.129.5:80). The reply took 1.3 ms on average, as shown below:

```
Ping statistics for 10.25.129.5:80
  50 probes sent.
Approximate trip times in milliseconds:
  Minimum = 0.25 ms, Maximum = 26.065 ms,
  Average = 1.323 ms
```

The TCP protocol uses several flags to manage the state of a connection in the packet header [35]. We focused on two of these, which are used in establishing TCP connections:

- SYN (Synchronize) which represents the initiation of a connection; and
- ACK (Acknowledge) which represents data received.

We monitored the traffic of the 50 probes at the server machine using Wireshark, by capturing the packets that were associated with the server using the filter "ip.addr = = 10.25.129.5". As the traffic was normal, the server machine replied to all requested packets according to the TCP protocol, as shown in Figure 2 (a and b).

In addition, the I/O graph was stable. All packets were answered and almost no TCP errors occurred. Note that the number of requesting packets was approximately less than 10 per second, as shown in Figure 3.

B. DURING THE ATTACK

An attack was launched using a software program which performed a DDoS TCP flood attack on a particular server. Once the DDoS TCP flood attack commenced on the victim machine in the cloud, the arriving packets were much more numerous than the server could handle. Consequently, the server could not respond to all the requesting packets from either normal users or the attackers. Note that 10.25.129.5 was the IP address of the victim server and 10.31.133.235 was the IP address of the attacker. The first request packet from the attacker was successful, as it was treated like a normal requesting packet. The subsequent ones were not successful, as the server was too busy and could not respond. A screen shot of the packet capture is shown in Figure 4 (a and b).

Finally, we sent 50 TCP test probes within a few seconds to the victim machine during the attack period to test the connection. The reply time was 9.6 ms on average, which differs considerably from the first test as shown below:

```
Ping statistics for 10.25.129.5:80
  50 probes sent.
Approximate trip times in milliseconds:
  Minimum = 0.181 ms, Maximum = 152.341 ms,
Average = 9.586 ms
```

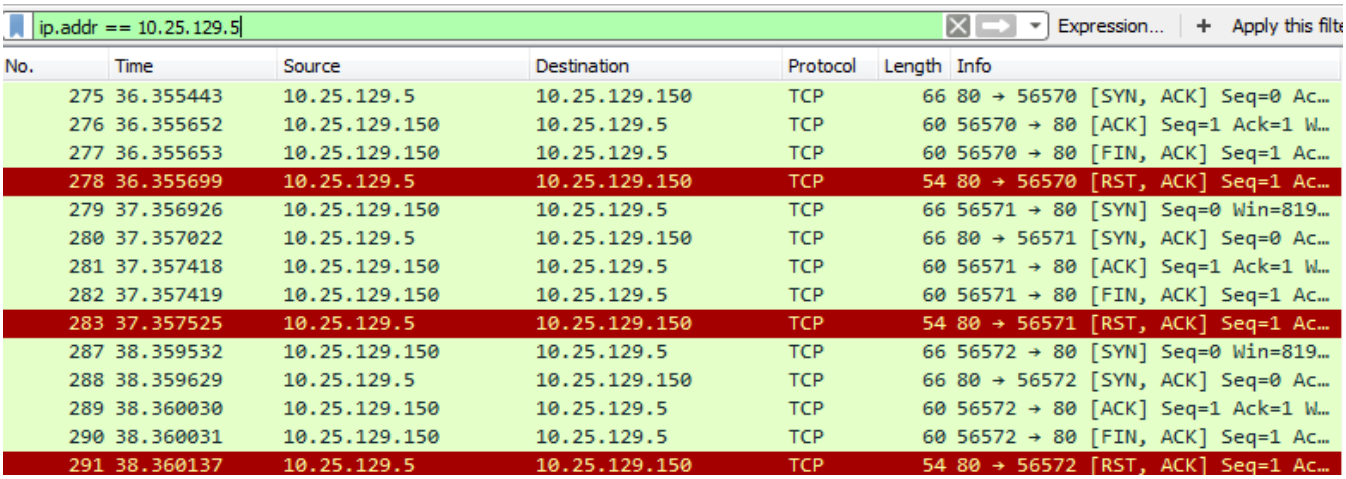
To sum up, the DDoS TCP flood attack can affect the cloud server's performance within a short time, slowing down the response, and can even stop the service completely. TCP errors will also be increased, as shown in Figure 5. Therefore, an efficient and effective detection and prevention technique is required.

IV. THE PROPOSED CS_DDoS SYSTEM

In this section we present the proposed CS_DDoS system, which can prevent DDoS TCP flood attacks. Firstly, it was assumed that the IP addresses of the attackers are not spoofed. Examples of how to prevent IP spoofing can be found in [36]. Our proposed system includes two sub-systems: the detection sub-system and prevention sub-system, as shown in Figure 6.

A. DETECTION PHASE

During the detection phase, the detection sub-system collects the incoming packets within a time frame, for example 60 seconds. The collected packets are subjected to a blacklist check to test whether their sources are blacklisted as attackers of the cloud system. If the packet source is listed in the attacker blacklist, the detection system will send the packets directly to the prevention sub-system without further processing.



ip.addr == 10.25.129.5 — Expression... — Apply this filt

No.	Time	Source	Destination	Protocol	Length	Info
275	36.355443	10.25.129.5	10.25.129.150	TCP	66	80 → 56570 [SYN, ACK] Seq=0 Ac…
276	36.355652	10.25.129.150	10.25.129.5	TCP	60	56570 → 80 [ACK] Seq=1 Ack=1 W…
277	36.355653	10.25.129.150	10.25.129.5	TCP	60	56570 → 80 [FIN, ACK] Seq=1 Ac…
278	36.355699	10.25.129.5	10.25.129.150	TCP	54	80 → 56570 [RST, ACK] Seq=1 Ac…
279	37.356926	10.25.129.150	10.25.129.5	TCP	66	56571 → 80 [SYN] Seq=0 Win=819…
280	37.357022	10.25.129.5	10.25.129.150	TCP	66	80 → 56571 [SYN, ACK] Seq=0 Ac…
281	37.357418	10.25.129.150	10.25.129.5	TCP	60	56571 → 80 [ACK] Seq=1 Ack=1 W…
282	37.357419	10.25.129.150	10.25.129.5	TCP	60	56571 → 80 [FIN, ACK] Seq=1 Ac…
283	37.357525	10.25.129.5	10.25.129.150	TCP	54	80 → 56571 [RST, ACK] Seq=1 Ac…
287	38.359532	10.25.129.150	10.25.129.5	TCP	66	56572 → 80 [SYN] Seq=0 Win=819…
288	38.359629	10.25.129.5	10.25.129.150	TCP	66	80 → 56572 [SYN, ACK] Seq=0 Ac…
289	38.360030	10.25.129.150	10.25.129.5	TCP	60	56572 → 80 [ACK] Seq=1 Ack=1 W…
290	38.360031	10.25.129.150	10.25.129.5	TCP	60	56572 → 80 [FIN, ACK] Seq=1 Ac…
291	38.360137	10.25.129.5	10.25.129.150	TCP	54	80 → 56572 [RST, ACK] Seq=1 Ac…

(a)

(b)

FIGURE 2. Captured packets and TCP flags (normal). (a) Captured packets. (b) TCP flags.

If the packet source is not blacklisted, the incoming packet will be passed to the classifier to decide whether the packets are normal (originating from a client) or abnormal (originating from an attacker). A packet is considered to be an attacking one if the source requests connections to the same destination more frequently than an assumed threshold. The threshold can be manually adjusted by the system administrator to cater for the varying requirements of a particular network. If a packet is considered to be normal, the detection system will send it to its destination (the cloud service provider). Otherwise, the detection sub-system will send the packet to the prevention sub-system.

Four different classifiers are used in the detection sub-system for the classification operation. The classifiers used are explained and evaluated in Section 5.

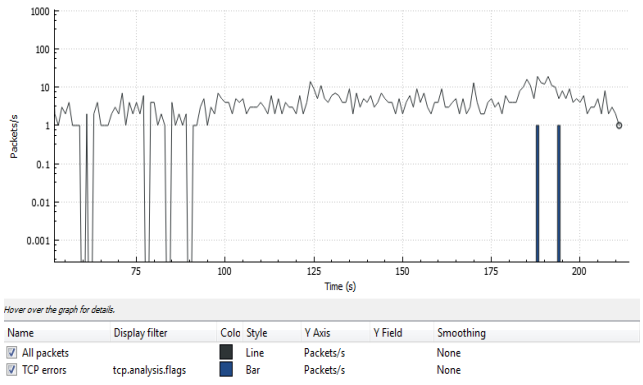
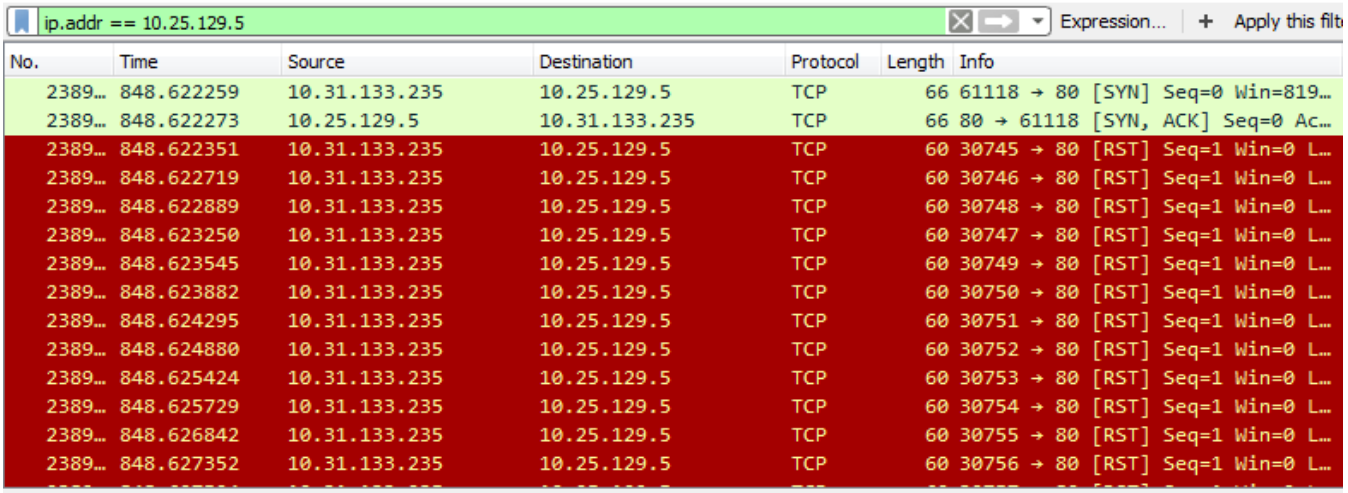


FIGURE 3. The I/O graph (no TCP errors).

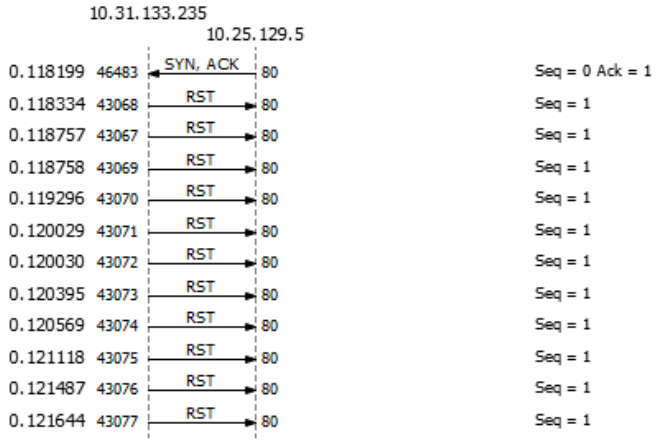
B. PREVENTION PHASE

When the packets reach the prevention system, they are considered to be attacking packets by the detection sub-system. The prevention sub-system first alerts the system administrator of the attacks. Then, the prevention sub-system will add the attacking source address to the attacker blacklist used by the detection sub-system, if it is not already on the list. Finally, the attacking packet will be dropped. The overall architecture of the CS_DDoS system is shown in Figure 6.

Algorithm 1 is used to determine whether these packets are normal or abnormal by counting the number of requests for a connection from an IP address and checking whether it



(a)



(b)

FIGURE 4. Captured packets and TCP flags (abnormal). (a) Captured packets. (b) TCP flags.

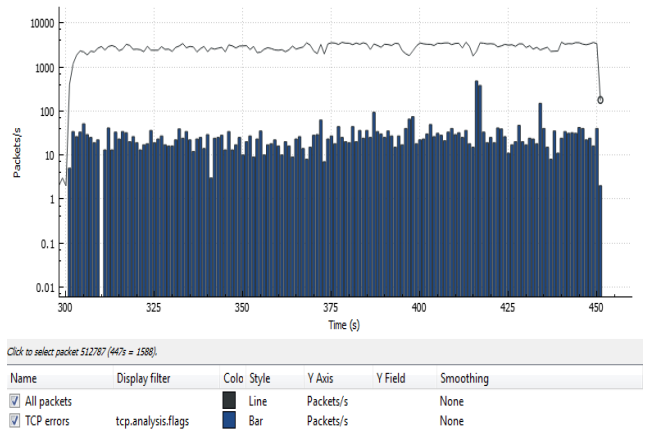


FIGURE 5. The I/O graph (with TCP errors).

exceeds a predefined threshold within a certain time frame. This algorithm is applied to the training data used for each classifier. As a result, each classifier used will predict the behavior of the attackers according to Algorithm 1.

Algorithm 1 Pre-Processing

```
Load data
For I=1: n
    P=data (I, 2)
    P2= (I, 1)
    For J=1: n
        N=find (data (J, 1) ==P2) & (data (J, 2) ==P)
        If N>=K
            New_data (I, 1) =data (I, 1)
            New_data (I, 2) =−1
        Else
            New_data (I, 1) =data (I, 1)
            New_data (I, 2) =1
End
```

where:
n is the number of packets
P is the destination IP address

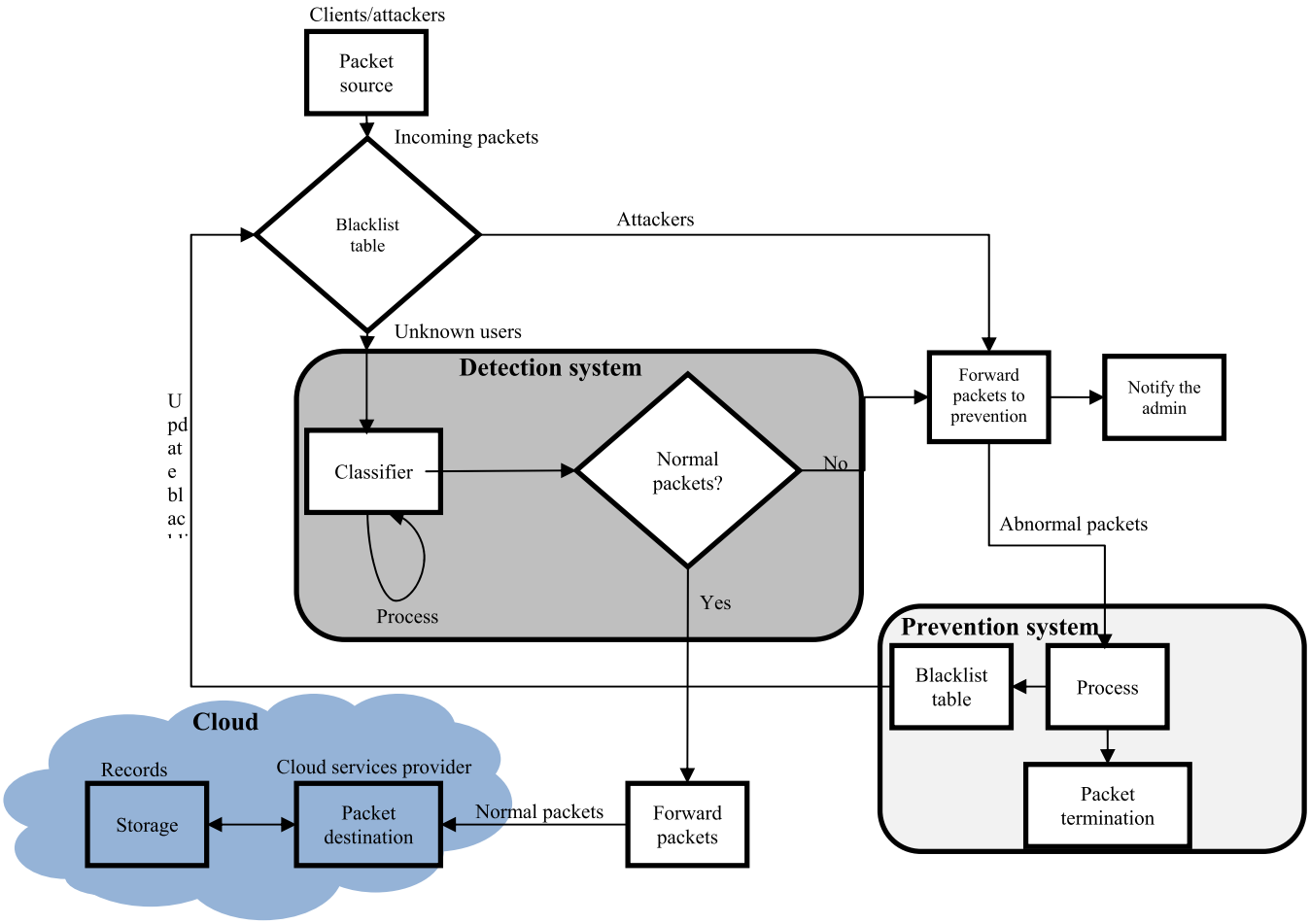


FIGURE 6. The overall architecture of the proposed CS_DDoS system.

P2 is the source IP address

N is the number of packets from the same source to the same destination within 60 seconds

K is the threshold for a packet to be considered an attacking packet

−1 indicates abnormal packets (blacklist array)

1 indicates normal packets

New_data () is a new entry list with tag "1" or "−1"

The proposed CS_DDoS system can be implemented in three possible scenarios. The first scenario is a normal service request packet. The requested service will be delivered as usual. The next scenario is when the source IP address is not blacklisted but the number of service requesting packets exceeds a predefined threshold within a certain time frame. The packet in this scenario will be considered a DDoS attack packet. The source address will be blacklisted and the packet will be dropped. The last scenario is when the source address of a packet is blacklisted and the packet is dropped without any further processing.

The three scenarios are illustrated using Quick Sequence Diagram Editor 4.2 [37]. The code used is shown in Table 11. The resulting sequence diagrams are shown in Figure 7 (a, b and c).

In case of flash crowd scenario, all packets must wait in a queue to be served sequentially.

The proposed CS_DDoS system can be used in any type of cloud, such as eHealth clouds, to ensure the security and availability of health records against DDoS TCP flood attacks.

V. EXPERIMENTAL RESULTS

A. CLASSIFICATION ALGORITHMS

In this section, we briefly explain the four commonly used classification algorithms used in our experiments. The classification algorithms are as follows:

1) LS-SVM

The LS-SVM is a powerful classifier in the field of pattern recognition for the detection of abnormalities from signals, images and time series signals. The LS-SVM is an efficient method of classifying two different sets of observations into their relevant classes. It is capable of handling high

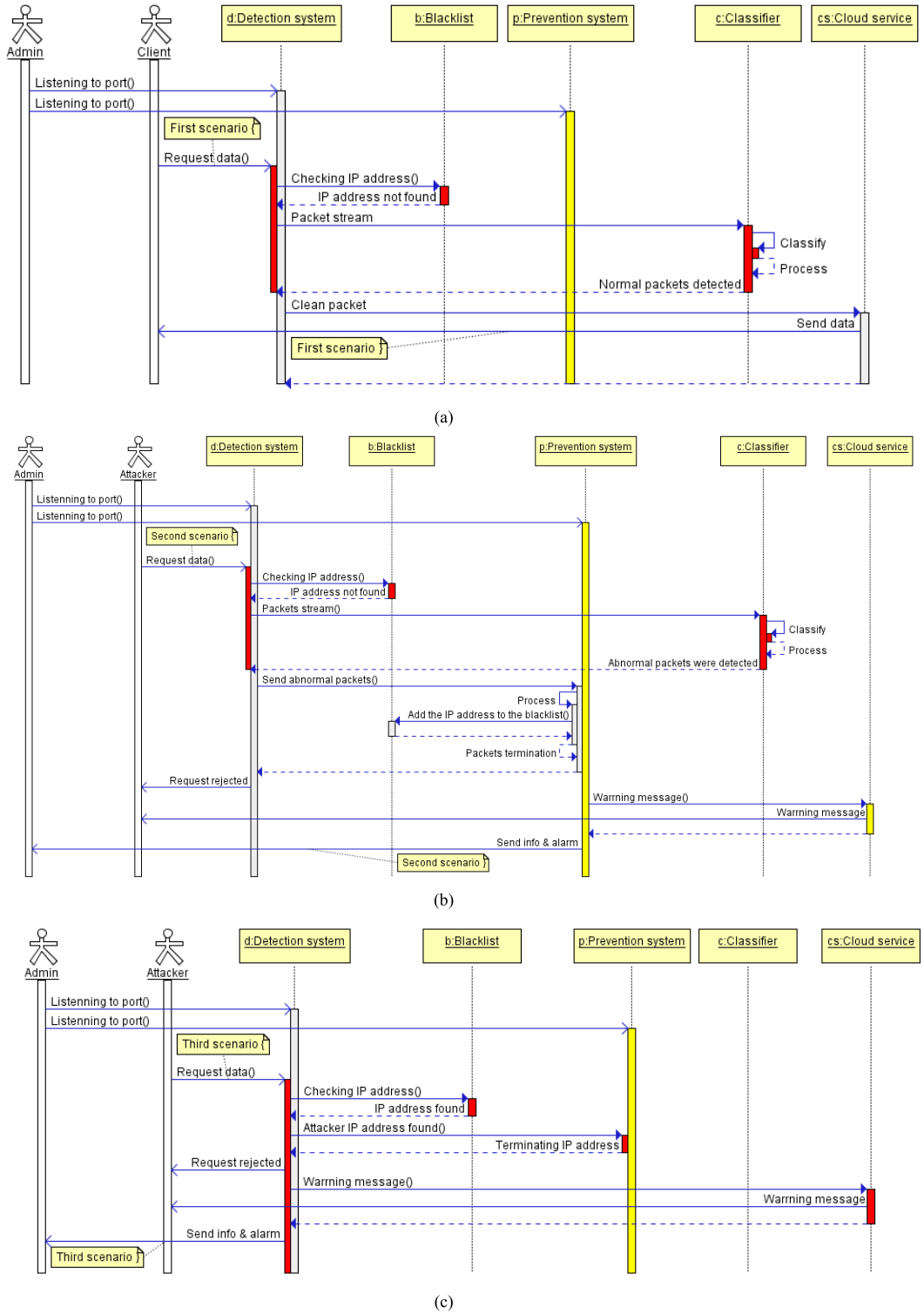


FIGURE 7. CS_DDoS possible scenarios. (a) First scenario (normal packets). (b) Second scenario (store abnormal packets in the blacklist). (c) Third scenario (abnormal packets already in the blacklist).

dimensional and non-linear data. In this work, the LS-SVM is employed to detect illegal activities in a network. The parameters of the LS-SVM are set during the training session to obtain a high proportion of detected results [38].

2) Naïve Bayes

Naïve Bayes is a frequently used classifier and has a straightforward approach based on the application of Bayes' theorem [39]. It is a simple approach which relies on proba-

TABLE 1. Classification performance measurements (n=1000 and K=100).

	Classifiers	Detection results			
		Accuracy	Sensitivity	Specificity	Kappa coefficient
1	LS-SVM	99.5%	95.3%	96%	0.91
2	Naïve Bayes	80%	92.3%	93%	0.82
3	K-nearest	75%	93.5%	95%	0.74
4	Multilayer perceptron	88.3%	95.3%	97%	0.78

TABLE 2. Classification performance measurements (n=2000 and K=200).

	Classifiers	Detection results			
		Accuracy	Sensitivity	Specificity	Kappa coefficient
1	LS-SVM	94.6%	94%	96%	0.89
2	Naïve Bayes	82%	93%	94%	0.75
3	K-nearest	80%	95%	93%	0.87
4	Multilayer perceptron	92%	97%	97%	0.65

TABLE 3. Classification performance measurements (n=5000 and K=300).

	Classifiers	Detection results			
		Accuracy	Sensitivity	Specificity	Kappa coefficient
1	LS-SVM	96%	98%	97%	0.90
2	Naïve Bayes	96%	94%	92%	0.82
3	K-nearest	82%	96%	94%	0.68
4	Multilayer perceptron	95%	99%	97%	0.75

TABLE 4. Classification performance measurements (n=6000 and K=400).

	Classifiers	Detection results			
		Accuracy	Sensitivity	Specificity	Kappa coefficient
1	LS-SVM	98%	99%	98%	0.85
2	Naïve Bayes	95%	95%	96%	0.67
3	K-nearest	85%	98%	97%	0.62
4	Multilayer perceptron	97%	99%	97%	0.58

bilistic knowledge to accurately predict test instances. This algorithm assumes that predictive attributes are conditionally independent and that there are no hidden attributes which can affect the prediction process [39]. The naïve Bayes classifier uses small training sets to provide relatively good performance, which generally overcomes any overtraining issues.

3) K-NEAREST

K-nearest is one of the most straightforward learning algorithms. In this algorithm, the similarity function relies on distance measurements to compute the similarity between training members [40]. The value of k is adjusted during the training session to assign each instance during training to the correct class. The k-nearest classifier is very sensitive to data size and dimensionality, and this affects the feature space and homogeneous areas, which represent the distribution of various classes [41].

4) MULTILAYER PERCEPTRON

The multilayer perceptron is a particular type of neural network-based classifier [42], [43]. This classifier employs a multilayer feed-forward neural network with one or more layers of nodes between the inputs and output layers. These nodes at different layers are interconnected through weighted networks. Using different training algorithms, the parameters (weights) of the networks are optimized. In this classifier, the data are transferred from input to output. Each feature is used as an input in the multilayer perceptron, and the outputs are the class categories. The multilayer perceptron may be linear, when it is used with a single layer of nodes. It can also be a nonlinear perceptron, when it is applied using multiple layers of nodes with several hidden layers [40].

B. PERFORMANCE EVALUATION AND VALIDATION

In this section, the performance of the CS_DDoS system is evaluated and validated using classification performance measurements and K-fold cross-validation.

1) PERFORMANCE EVALUATION

In this section, the performance of the CS_DDoS method is evaluated using the four classifiers of the LS-SVM, naïve Bayes, k-nearest, and multilayer perceptron. Various training data sizes (window sizes) and thresholds are used in the experiments. Algorithm 1 is applied to the training data for all the classifiers.

The CS_DDoS system was evaluated in terms of accuracy, sensitivity (detection rate) and specificity (false alarm

TABLE 5. Classification performance average.

	Classifiers	Average			
		Accuracy	Sensitivity	Specificity	Kappa coefficient
1	LS-SVM	97%	97%	97%	0.8875
2	Naïve Bayes	88%	94%	94%	0.765
3	K-nearest	81%	96%	95%	0.7275
4	Multilayer perceptron	93%	98%	97%	0.69

TABLE 6. Classification performance measurements (n=6000 and K=400).

	Classifiers	Detection results			
		Accuracy	Sensitivity	Specificity	Kappa coefficient
1	LS-SVM	98%	93%	94%	0.91
2	Naïve Bayes	82%	91.3%	91%	0.82
3	K-nearest	80%	91.5%	92%	0.74
4	Multilayer perceptron	83.3%	92.3%	95%	0.78

TABLE 7. Classification performance measurements (n=6000 and K=400).

	Classifiers	Detection results			
		Accuracy	Sensitivity	Specificity	Kappa coefficient
1	LS-SVM	93%	91%	94%	0.91
2	Naïve Bayes	85%	92%	95%	0.81
3	K-nearest	79%	96%	92%	0.82
4	Multilayer perceptron	87%	95%	96%	0.71

TABLE 8. Classification performance measurements (n=6000 and K=400).

	Classifiers	Detection results			
		Accuracy	Sensitivity	Specificity	Kappa coefficient
1	LS-SVM	94%	97%	95%	0.92
2	Naïve Bayes	95%	92%	94%	0.85
3	K-nearest	88%	90%	92%	0.69
4	Multilayer perceptron	89%	97%	93%	0.81

TABLE 9. Classification performance measurements (n=6000 and K=400).

	Classifiers	Detection results			
		Accuracy	Sensitivity	Specificity	Kappa coefficient
1	LS-SVM	92%	97%	94%	0.87
2	Naïve Bayes	91%	93%	95%	0.65
3	K-nearest	87%	91%	96%	0.69
4	Multilayer perceptron	94%	97%	94%	0.60

TABLE 10. Classification performance average.

	Classifiers	Average			
		Accuracy	Sensitivity	Specificity	Kappa coefficient
1	LS-SVM	94%	95%	94%	0.9025
2	Naïve Bayes	88%	92%	94%	0.7825
3	K-nearest	84%	92%	93%	0.735
4	Multilayer perceptron	88%	95%	95%	0.725

rate), as well as the descriptive statistic Kappa coefficient. Kappa coefficients are procedures used to connect between categorical variables, and are frequently used as consistency or legitimacy coefficients [44].

The accuracy represents the rate of correctly identified results over the entire data used by the CS_DDoS. or true negatives (TN), while incorrectly identified results are false positives (FP) and false negatives (FN). The accuracy of the CS_DDoS system is measured by Equation (1).

- True positives (TP): correctly identified abnormal packets in this research.
- False positives (FP): incorrectly identified abnormal packets.

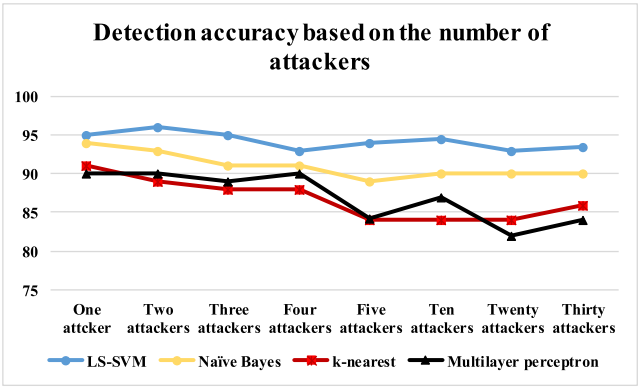


FIGURE 8. Detection accuracy for multiple attacks.

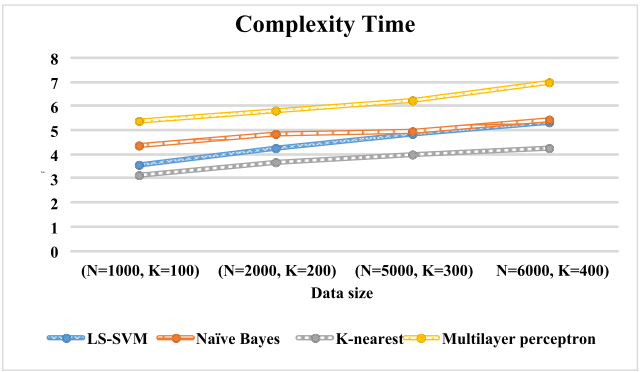


FIGURE 9. Complexity times.

- True negatives (TN): correctly identified normal packets.
- False negatives (FN): incorrectly identified normal packets.

$$CS\_DDoS_{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \times 100\% \quad (1)$$

The sensitivity represents the rate of correctly identified abnormal packets over the entire range of positive results obtained by the CS_DDoS. The sensitivity of the CS_DDoS system is measured by Equation (2).

$$CS\_DDoS_{Sensitivity} = \frac{TP}{TP + FN} \times 100\% \quad (2)$$

The specificity represents the rate of incorrectly identified abnormal packets over the entire range of negative results produced by the CS_DDoS.

The specificity of the CS_DDoS system is measured by Equation (3).

$$CS\_DDoS_{Specificity} = \frac{FP}{FP + TN} \times 100\% \quad (3)$$

The proposed CS_DDoS system is evaluated under both single source and multiple source attack environments, as described below.

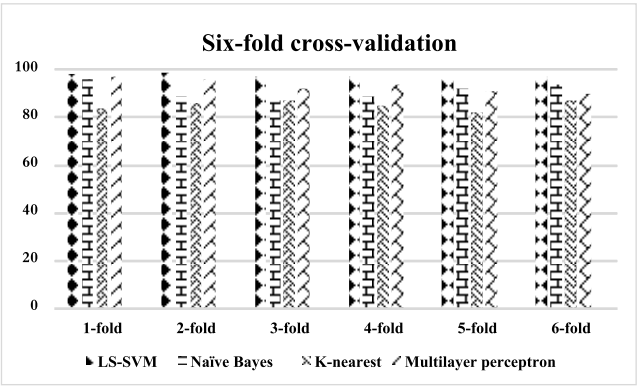


FIGURE 10. Six-fold cross-validation diagram.

a: EVALUATION UNDER SINGLE SOURCE ATTACK

Four test data sizes (n) of 1000, 2000, 5000 and 6000 packets were randomly selected, and four thresholds (K) of 100, 200, 300 and 400 requests. Algorithm 1 was applied to the data according to the window size, n, and was tested according to the threshold K. We have two features fed to each classifier; these two features are the source IP address and the destination IP address. Each classifier was used to classify the data using the four windows and four thresholds. The results are shown in Tables 1-5:

Tables 1 to 4 show the classification performances of the proposed CS_DDoS system with different data sizes and thresholds. The performance measurements are accuracy (correctly detected data over the entire dataset), sensitivity (correctly detected attacks, detection rate), specificity (incorrectly detected attacks, false alarm rate), and Kappa coefficient (stability rate).

According to Tables 1 to 4, the results of each classifier were not significantly affected by the window sizes and thresholds, since there are only small differences between the tables. Tables 1 to 4 are summarized in Table 5.

From Table 5, it can be seen that the LS-SVM classifier has the highest average percentage accuracy (97%) and the highest Kappa coefficient (0.89). Conversely, the k-nearest classifier achieved the lowest accuracy percentage of about 81%, and the multilayer perceptron classifier had the lowest Kappa coefficient 0.69. Overall, the proposed CS_DDoS system is more effective and stable in resisting a single-source attack when adopting the LS-SVM classifier regardless of the window size and threshold.

b: EVALUATION UNDER MULTIPLE-SOURCE ATTACKS

To evaluate the performance under attacks from multiple sources, the same four window sizes were used (1000, 2000, 5000 and 6000) and the same four thresholds (100, 200, 300 and 400). Algorithm 1 was also used. The results are shown in Tables 6–10: Tables 6–9 show the results of the classification accuracy of the proposed CS_DDoS system when under multiple DDoS attacks. Tables 6–9 also show that the results of each classifier were not significantly affected

TABLE 11. Sequence diagram generation codes implemented using quick sequence diagram editor 4.2.

First scenario	Second scenario	Third scenario
Admin:Actor Client:Actor d:Detection system b:Blacklist p:Prevention system c:Classifier cs:Cloud service Admin:d.Listening to port() Admin:p.Listening to port() +1 Client First scenario { +1 (1) Client:d.Request data() d:IP address not found=b.Checking IP address() d:Normal packets detected=c.Packet stream c:Process=c.Classify c:stop d:cs.Clean packet (2)cs:Client.Send data +2 d First scenario } +2	Admin:Actor Attacker:Actor d:Detection system b:Blacklist p:Prevention system c:Classifier cs:Cloud service Admin:d.Listening to port() Admin:p.Listening to port() +3 Attacker Second scenario { +3 (3)Attacker:d.Request data() d:IP address not found=b.Checking IP address() d:Abnormal packets detected=c.Packets stream() c:Process=c.Classify c:stop d:p.Send abnormal packets() p:Packet termination=p.Process p:b.Add IP address to blacklist() d:Attacker.Request rejected p:cs.Warrning message() cs:Attacker.Warrning message (4)p:Admin.Send info & alarm +4 b Second scenario } +4	Admin:Actor Attacker:Actor d:Detection system b:Blacklist p:Prevention system c:Classifier cs:Cloud service Admin:d.Listening to port() Admin:p.Listening to port() +5 Attacker Third scenario { +5 (5)Attacker:d.Request data() d:IP address found=b.Checking IP address() d:Terminating IP address=p.Attacker IP address found() d:Attacker.Request rejected d:cs.Warrning message() cs:Attacker.Warning message (6)d:Admin.Send info & alarm +6 Admin Third scenario } +6

by the window size and the threshold. LS-SVM was again the best performing classifier with percentage accuracy of around 94%, and a Kappa coefficient of about 0.9. Tables 6-9 are summarized in Table 10.

Overall, the proposed CS_DDoS system is also effective and stable in resisting both multiple-source and single-source attacks when using the LS-SVM classifier, regardless of the window size and threshold. Therefore, the proposed CS_DDoS system can be implemented in a large-scale cloud project, such as a health cloud, as well as in smaller projects such as a private cloud for a medium-sized company. CS_DDoS can prevent DDoS attacks with a 94% accuracy and is highly stable (Kappa coefficient 0.9). CS_DDoS outperforms previous approaches, since either the percentage accuracy of previous approaches is lower than those achieved by CS_DDoS, for example 91% in [45], or are without Kappa coefficient stability measurements for example in [46].

In addition, the false alarm rate (specificity) of the benchmark algorithms are 69.57% on average [47]. Thus, we can claim that our proposed CS_DDoS system is more effective.

To shed more light on the performance evaluation of the proposed CS_DDoS system, the simulation was repeated with various numbers of attackers (source IP) under similar conditions and the performance measurements were calculated.

Figure 8 shows the performance of CS_DDoS with an increasing number of attackers. There are slight fluctuations in the performance measurements of all four classifiers, although LS-SVM was still the best performer of the four.

In addition, the process complexity times of the four classification algorithms is shown in Figure 9. While LS-SVM is only the second least time-consuming, the fastest classifier, k-nearest, has lower performance measurements and a smaller Kappa coefficient compared to LS-SVM. It can therefore be considered that the LS-SVM is the most efficient and effective classifier for use in the CS_DDoS system to resist DDoS TCP flood attacks.

2) K-FOLD CROSS-VALIDATION

K-fold cross-validation is a validation model for measuring how the outcomes of a numerical examination will simplify to an independent dataset. Generally, it is utilized to validate the estimation of performance accuracy in practice for a predictive model [48-51].

K-fold cross-validation was used to carry out a performance comparison of the four predictive modeling algorithms used in CS_DDoS: LS-SVM, naïve Bayes, k-nearest, and multilayer perceptron. These four algorithms were compared in terms of their prediction results.

The dataset was divided into six equal-sized chunks, k=6. As a validation for model testing, one of the six chunks was retained, and the remainder (five chunks) were used as training data. Then, the process of the six-cross model was repeated six times, so that each of the six chunks were used as validation data for each model. The results are shown in

Figure 10. We can see that the values of all folds are almost the same, which means that each fold has approximately the same rate for each of the four classification algorithms. Thus, we can claim that the classification results are stable and accurate, since each algorithm gives almost the same results for each fold.

VI. CONCLUSION

The use of cloud computing in many sectors is becoming widespread, as this helps to improve the system in many respects. However, this cloud project is vulnerable to certain types of attacks, such as DDoS TCP flood attacks. Therefore, we propose a new approach called CS_DDoS for the detection and prevention of DDoS TCP flood attacks. The system is based on classification to ensure the security and availability of stored data, especially important for eHealth records for emergency cases. In this approach, the incoming packets are classified to determine the behavior of the source within a time frame, in order to discover whether the sources are associated with a genuine client or an attacker. The results show that using LS-SVM the CS_DDoS system can identify the attacks accurately. The system has an accuracy of about 97 percent with a Kappa coefficient of about 0.89 when under single attack; it is 94 percent accurate with a Kappa coefficient of about 0.9 when under multiple attacks. The performance is validated using K-fold validation and is shown to be stable and accurate. Thus, the proposed approach can efficiently improve the security of records, reduce bandwidth consumption and mitigate the exhaustion of resources. In the future, we aim to extend CS_DDoS to overcome the problem of DDoS using spoofed IP addresses as well as to improve the proposed work to identify the attackers even when they satisfy the threshold value.

APPENDIX

CONFLICTS OF INTEREST

The authors declare that there are no conflicts of interest regarding the publication of this paper.

REFERENCES

[1] A. Girma, K. Abayomi, and M. Garuba, "The design, data flow architecture, and methodologies for a newly researched comprehensive hybrid model for the detection of DDoS attacks on cloud computing environment," in *Proc. Inf. Technol. Generat. 13th Int. Conf. Inf. Technol.*, 2016, pp. 377–387.

[2] B. Cashell, W. D. Jackson, M. Jickling, and B. Webel, *The Economic Impact of Cyber-Attacks*, document CRS RL32331, Congressional Research Service Documents, Washington, DC, USA, 2004.

[3] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 602–622, 1st Quart., 2016.

[4] P. A. Laplante, J. Zhang, and J. Voas, "What's in a name? Distinguishing between SaaS and SOA," *IT Prof.*, vol. 10, no. 3, pp. 46–50, May 2008.

[5] C. Balding. (2012). *What Everyone Ought to Know About Cloud Security*. [Online]. Available: http://www.slideshare.net/craigbalding/what-everyone-ought-to-know-about-cloud-security

[6] I. M. Khalil, A. Khreishah, and M. Azeem, "Cloud computing security: A survey," *Computers*, vol. 3, pp. 1–35, 2014.

[7] B. Wang, Y. Zheng, W. Lou, and Y. T. Hou, "DDoS attack protection in the era of cloud computing and software-defined networking," *Comput. Netw.*, vol. 81, pp. 308–319, Apr. 2015.

[8] M. Xia, W. Lu, J. Yang, Y. Ma, W. Yao, and Z. Zheng, "A hybrid method based on extreme learning machine and k-nearest neighbor for cloud classification of ground-based visible cloud image," *Neurocomputing*, vol. 160, pp. 238–249, Jul. 2015.

[9] A. Taravat, F. D. Frate, C. Cornaro, and S. Vergari, "Neural networks and support vector machine algorithms for automatic cloud classification of whole-sky ground-based images," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 3, pp. 666–670, Mar. 2015.

[10] A. Sahi, D. Lai, and Y. Li, "Security and privacy preserving approaches in the eHealth clouds with disaster recovery plan," *Comput. Biol. Med.*, vol. 78, pp. 1–8, Nov. 2016.

[11] A. S. Khader and D. Lai, "Preventing man-in-the-middle attack in Diffie–Hellman key exchange protocol," in *Proc. 22nd Int. Conf. Telecommun. (ICT)*, 2015, pp. 204–208.

[12] A. Sahi, D. Lai, and Y. Li, "Parallel encryption mode for probabilistic scheme to secure data in the cloud," in *Proc. 10th Int. Conf. Inf. Technol. Appl. (ICITA)*, Sydney, NSW, Australia, 2015.

[13] A. A. Hameed, B. Karlik, and M. S. Salman, "Back-propagation algorithm with variable adaptive momentum," *Knowl.-Based Syst.*, vol. 114, pp. 79–87, Dec. 2016.

[14] U. R. Acharya *et al.*, "Automated characterization and classification of coronary artery disease and myocardial infarction by decomposition of ECG signals: A comparative study," *Inf. Sci.*, vol. 377, pp. 17–29, Jan. 2017.

[15] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 2, pp. 39–53, 2004.

[16] W. Wei, F. Chen, Y. Xia, and G. Jin, "A rank correlation based detection against distributed reflection DoS attacks," *IEEE Commun. Lett.*, vol. 17, no. 1, pp. 173–175, Jan. 2013.

[17] P. E. Ayres, H. Sun, H. J. Chao, and W. C. Lau, "ALPi: A DDoS defense system for high-speed networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 10, pp. 1864–1876, Oct. 2006.

[18] A. D. Keromytis, V. Misra, and D. Rubenstein, "SOS: An architecture for mitigating DDoS attacks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 1, pp. 176–188, Jan. 2004.

[19] X. Wang and M. K. Reiter, "Using Web-referral architectures to mitigate denial-of-service threats," *IEEE Trans. Depend. Sec. Comput.*, vol. 7, no. 2, pp. 203–216, Apr. 2010.

[20] Y. Xuan, I. Shin, M. T. Thai, and T. Znati, "Detecting application denial-of-service attacks: A group-testing-based approach," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 8, pp. 1203–1216, Aug. 2010.

[21] K. Salah, K. Elbadawi, and R. Boutaba, "Performance modeling and analysis of network firewalls," *IEEE Trans. Netw. Service Manage.*, vol. 9, no. 1, pp. 12–21, Mar. 2012.

[22] W. Dou, Q. Chen, and J. Chen, "A confidence-based filtering method for DDoS attack defense in cloud environment," *Future Generat. Comput. Syst.*, vol. 29, no. 7, pp. 1838–1850, 2013.

[23] J. Yu, H. Lee, M.-S. Kim, and D. Park, "Traffic flooding attack detection with SNMP MIB using SVM," *Comput. Commun.*, vol. 31, pp. 4212–4219, Nov. 2008.

[24] K. Lee, J. Kim, K. H. Kwon, Y. Han, and S. Kim, "DDoS attack detection method using cluster analysis," *Expert Syst. Appl.*, vol. 34, no. 3, pp. 1659–1665, 2008.

[25] A. Chonka, Y. Xiang, W. Zhou, and A. Bonti, "Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks," *J. Netw. Comput. Appl.*, vol. 34, no. 4, pp. 1097–1107, 2011.

[26] K. Lu, D. Wu, J. Fan, S. Todorovic, and A. Nucci, "Robust and efficient detection of DDoS attacks for large-scale Internet," *Comput. Netw.*, vol. 51, no. 18, pp. 5036–5056, 2007.

[27] S. Khanna, S. S. Venkatesh, O. Fatemieh, F. Khan, and C. A. Gunter, "Adaptive selective verification: An efficient adaptive countermeasure to thwart DoS attacks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 3, pp. 715–728, Jun. 2012.

[28] R. Guo, H. Yin, D. Wang, and B. Zhang, "Research on the active DDoS filtering algorithm based on IP flow," *Int. J. Commun., Netw. Syst. Sci.*, vol. 7, pp. 600–607, Sep. 2009.

[29] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *Proc. IEEE 35th Conf. Local Comput. Netw. (LCN)*, Oct. 2010, pp. 408–415.

[30] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection," *Pattern Recognit. Lett.*, vol. 51, pp. 1–7, Jan. 2015.

Appendix C

Classification of Epileptic EEG signals Based on Simple Random Sampling and Sequential Feature Selection

A contribution is made representing in editing the paper, analysing some of the obtained results.

Brain Informatics (2016) 3:85–91
DOI 10.1007/s40708-016-0039-1

Classification of epileptic EEG signals based on simple random sampling and sequential feature selection

Hadi Ratham Al Ghayab · Yan Li · Shahab Abdulla · Mohammed Diykh · Xiangkui Wan

Received: 4 December 2015 / Accepted: 3 February 2016 / Published online: 27 February 2016

Abstract Electroencephalogram (EEG) signals are used broadly in the medical fields. The main applications of EEG signals are the diagnosis and treatment of diseases such as epilepsy, Alzheimer, sleep problems and so on. This paper presents a new method which extracts and selects features from multi-channel EEG signals. This research focuses on three main points. Firstly, simple random sampling (SRS) technique is used to extract features from the time domain of EEG signals. Secondly, the sequential feature selection (SFS) algorithm is applied to select the key features and to reduce the dimensionality of the data. Finally, the selected features are forwarded to a least square support vector machine (LS_SVM) classifier to classify the EEG signals. The LS_SVM classifier classified the features which are extracted and selected from the SRS and the SFS. The experimental results show that the method achieves 99.90, 99.80 and 100 % for classification accuracy, sensitivity and specificity, respectively.

H. R. A. Ghayab (✉) · Y. Li · S. Abdulla · M. Diykh
Faculty of Health, Engineering and Sciences, University of Southern Queensland, Toowoomba, QLD 4350, Australia
e-mail: HadiRathamGhayab.AlGhayab@usq.edu.au

Y. Li
e-mail: Yan.Li@usq.edu.au

S. Abdulla
e-mail: Shahab.Abdulla@usq.edu.au

M. Diykh
e-mail: Mohammed.Diykh@usq.edu.au

X. Wan
School of Electrical and Electronic Engineering, Hubei University of Technology, Wuhan 430068, China
e-mail: wanxiangkui@163.com

Keywords Electroencephalogram · Epileptic seizures · Simple random sampling · Sequential feature selection · Least square support vector machine

1 Introduction

Epilepsy is a disorder which affects the human brain and hugely impairs patients' daily lives. It is characterized by recurrent and sudden incidence of epileptic seizures [1]. According to an estimation of the World Health Organization, more than 50 million of population are affected by epilepsy [2, 3]. Approximately, almost 1 % population have the neurological disorders [4–6]. It leads to numerous research works to identify epilepsy and related treatments. Electroencephalogram (EEG) signals have been proved as a powerful tool for detecting and diagnosing different neurological diseases. EEG signals are often used to detect and classify epilepsy [7]. It is often difficult for the experts to recognize the people who have a brain disorder through visual inspection of EEG signals [8]. In addition, visual inspection for discriminating EEG signals is a time consuming, error prone, costly process and not sufficient enough for reliable information. The analysis and classification of EEG signals can lead to better diagnostic techniques for brain-related disorders. It is thus important to develop better EEG classification methods.

Many researchers developed new techniques to extract the significant information from EEG signals. The information is used as the input to different classifiers. There are many approaches used to extract the key features as well as to further select features. Most of these fall under five broad categories: time domain, frequency domain, time–frequency domain, traditional non-linear methods and graph theory approaches [9].

One of the methods used in this paper for extracting epileptic EEG data is sample random sampling (SRS) technique. Researchers often applied the SRS in time domain. In this technique, each sample of the population has the same chance to be selected as a subject. The complete process of sampling is done in a single step, with each subject can be selected independently from the other samples of the population [10]. Then, we forwarded all these samples to the sequential feature selection (SFS) method for selecting the best features.

This study uses the selected features as the input for a classifier. One of the most popular classifiers, the least square support vector machines (LS_SVMs) [11], is used to classify EEG data. This technique is used to identify the EEG data from healthy people and epileptic patients for epileptic seizures.

A lot of approaches for EEG signals classification have been developed [12]. There were reported a diverse of classification precisions for epileptic EEG data. Brief discussions of the previous research are provided below.

Gajic et al. [13] extracted different features from time, frequency, time–frequency domain and non-linear analysis.

These features were obtained from sub-bands with good representative characteristics. The researchers reduced the dimension of the features by using scatter matrices. This method yielded 98.7 % accuracy.

An optimum allocation-based principal component analysis method was proposed by Siuly and Li [8] to extract key features for the classification of multi-class EEG signals from epileptic EEG data. They used four different classifiers which were LS_SVM, naive Bayes classifier, *k*-nearest neighbour (KNN) algorithm and linear discriminant analysis, to find out which one was the best classifier. They used four different output coding approaches for the multi-class LS_SVM. These were error correcting output codes, minimum output codes, one versus one (1vs1) and one versus all. That method achieved a 100 % accuracy with LS_SVM_1vs1.

Feature extraction was carried out through an empirical mode decomposition. The extracted features were forwarded to two classifiers, the classification and regression tree and the C4.5 classifiers. The method using the C4.5 classifier suggested by Martis et al. [14] obtained good experimental results of 95.33, 98 and 97 % for accuracy, sensitivity and specificity, respectively.

Chua et al. [15] gained features from raw EEG recordings by using higher order spectra. They used a Gaussian mixture model (GMM) and a SVM classifiers to detect epileptic EEG signals. They achieved average accuracies of 93.11 and 92.56 % for the HOS based GMM classifier and the SVM classifier, respectively, for different EEG classes, such as normal, pre-ictal and epileptic EEGs.

On the other hand, a genetic algorithm (GA) was used by Guo et al. [16] to automatically extract features from EEG data in order to enhance the classifier's performance, as well as, to reduce the feature's dimensionality. They used two groups of epileptic datasets. The first group was two classes of healthy people and epileptic patients. The second group was three classes of healthy people, inter-ictal and ictal. The KNN classifier was used in the work to classify the two groups. They gained 88.6 and 99.2 % accuracies for the first group without GA and with GA, respectively. They obtained of a 67.2 % accuracy without GA, and 93.5 % within GA, respectively, for the second group.

Ocak decomposed EEG signals, which were recorded from normal subjects and epileptic patients, by using discrete wavelet transform [17]. An approximate entropy (ApEn) was extracted from the approximation and the detail coefficients. The methodology achieved more than 96 % accuracy.

Srinivasan et al. used the ApEn to extract features and an artificial neural network classifier to identify epileptic EEG signals [18]. That approach achieved a high overall accuracy of 100 %.

Srinivasan et al. also proposed a special type of recurrent neural network, Elman network [19]. They used the feature extracted in time domain and frequency domain as the input to the proposed classifier. The Elman network method yielded a 99.6 % accuracy with a single input feature.

A wavelet transform method was used by Gajic et al. [20] to extract the key features. They also used scatter matrices to reduce the dimensionality of the features. These features were used as the input to a quadratic classifier. The EEG epileptic database was classified into healthy subjects, epileptic subjects during a seizure-free (inter-ictal) and epileptic patients during the seizure activity (ictal). They obtained a 99 % classification accuracy.

Shen et al. [12] proposed a cascade of wavelet-ApEn for feature selection. They used Fisher scores for adaptive feature selection, and SVM for feature classification to detect epileptic seizures. They applied the method to different epileptic EEG recordings: open source EEG data and clinical EEG data. The method obtained the overall classification accuracies of 99.97 and 98.73 %, respectively.

A sampling technique (ST) based on a LS_SVM was proposed by Siuly et al. [21]. Firstly, they used the ST to extract features from two classes of, normal persons with eyes open and epileptic patients during a seizure activity. They applied the LS_SVM to the extracted features. The total classification accuracy by that approach for both the training and testing datasets was 80.31 and 80.05 %, respectively.

Husain and Rao [22] presented an artificial neural network model using back propagation algorithm for the classification of epileptic EEG signals. They decomposed the EEG signals into a finite set of band limited signals termed as intrinsic mode functions. They also applied

Hilbert transform on these intrinsic mode functions to calculate instantaneous frequencies. They achieved a 99.80 % overall classification accuracy.

Rückstieß et al. [23] performed a SFS method to select the most representative features at each time step. Each successive features depended on the previous features. All the features were put into one vector and were forwarded to a classifier. This approach was applied for handwritten digits classification and a medical diabetes prediction task.

A sequential floating forward selection (SFFS) algorithm was proposed to detect epileptic seizures in EEG signals by Choi et al. [24]. They selected the most energy power as the features from frequency bands by using the SFFS algorithm. The total accuracy obtained by that method was 97.2 %.

In this study, we developed a new method combining the SRS with the SFS to acquire the best features set, and then we use the features as the input of the LS_SVM classifier for the EEG classification. All the techniques are discussed in Sects. 3 and 4. The conclusion is presented in Sect. 5.

2 Experimental data

The data used in this study are open source EEG recordings and are publicly available[1] [25]. The database includes five sets of EEG recordings (sets **A**–**E**), with each containing 100 single-channel EEG signals of 23.6 s from five separate classes. References [13, 26] presented all details of these datasets from set **A** to **E**. This study selected set **A** which was taken from surface EEG recordings of five healthy people with eye open, and set **E** which was taken from EEG records of five pre-surgical epileptic patients during epileptic seizure activity.

3 Methodology

The big EEG datasets cause the curse of dimensionality and make it difficult to estimate the accuracy of classification from a limited number of samples. This study develops a new structure for classifying epileptic EEG signals, as presented in Fig. 1. This work investigates and explores whether the SRS combined with SFS give the best features for epileptic EEG signals classification.

3.1 Simple random sampling (SRS) technique

SRS technique is a popular type of random or prospect sampling [21]. In this technique, each sample of the population has the same chance of being selected as a subject.

[1] http://www.meb.unibonn.de/epileptologie/science/physik/eegdata.html.

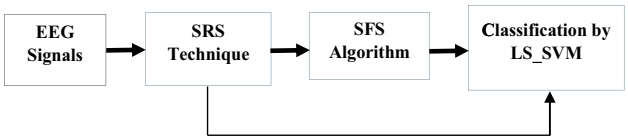


Fig. 1 The structure of the proposed system

We put the number of population in a sample size calculator of the "Creative Research System" (available in sample size calculator online), to determine the sample size for both samples and subsamples. In this work, the dataset used are set **A** and set **E** (repeated). Each set has 100 data files, and each file has 4097 observations.

This research uses the sample size calculator to find the sample size needed as well as to find the subsample size. The sizes of the samples and the subsamples in this work are 3288 and 2746, respectively. The sizes were selected because they reflect the limitation of time to select samples and subsamples. Firstly, we randomly select 10 samples from size 3288 for each dataset (set **A** or **E**). Secondly, 5 subsamples are also random chosen from each 10 random samples, with a size of 2746. In each step, this study takes into account a 99–100 % confidence interval and a 99 % confidence level. In the last step, nine statistical features are extracted from each subsample. These features are {maximum value (Max), minimum value (Min), mean value, median value, mode, first quartile (Xq1), second quartile (Xq2), range value and standard deviation (Std)}. Figure 2 shows how samples, subsamples and features are taken from each class. We used MATLAB software package version 8.4, R2014b, for the experiments.

3.2 Sequential feature selection (SFS) algorithms

The SFS is used to reduce the dimensionality of the dataset selected randomly from the SRS. This method is used to generate fewer numbers of uncorrelated variables which are utilized as the features for the better classification of EEG signals. The aim of the presented sequential selection algorithm is to decrease the feature space, $\mathbf{D} = x_1, x_2, \ldots, x_n$, to a subset of features, $\mathbf{D} - n$. It aims at enhancing or optimizing the computational execution of the classifier, as well as avoiding the curse of dimensionality [27]. This method is used to select a sufficiently reduced subset from the feature space **D** without affecting the performance of the classifier. In order to choose a suitable feature subset size k, namely, a criterion function typically estimates the recognition rate of the classifier [28]. The SFS algorithm starts with an empty set $\boldsymbol{S}$, and progressively fills the set $\boldsymbol{S}$ through adding features selected by the criterion function [29, 30]. It is searching on the feature space from bottom to up. Figure 3 illustrates how the SFS picks features from the original data. The SFS is applied to select the best features

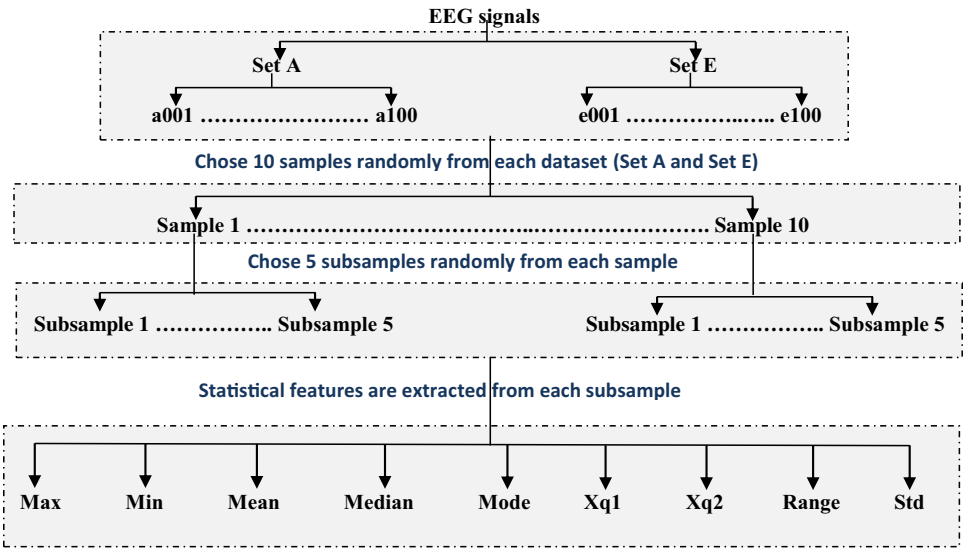


Fig. 2 The SRS technique to select samples, subsamples and statistical features

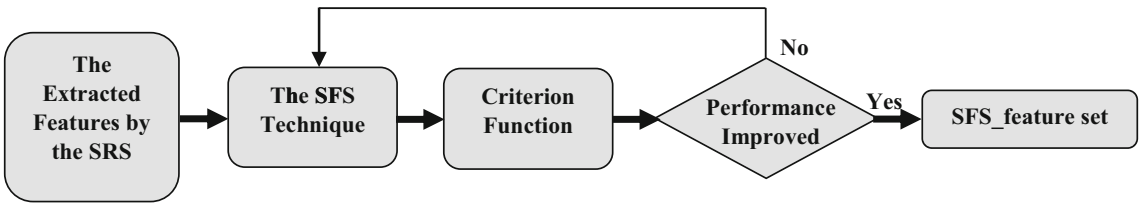


Fig. 3 Features selection from the extracted features by the SRS

from the statistical features. The criterion is empirically chosen based on the experimental results. In this study, several experiments are made to define the best criterion. The criterion value is calculated based on the statistical relations among the features. Firstly, the Max value is chosen as the criterion as shown in Eq. (1).

$$\delta = \rho \sum_{i=1}^{n} fs2(i) \quad i = 1,\ 2, \ldots, n, \tag{1}$$

where δ refers to the criterion, ρ is one of the nine statistical features, n is the number of the features and fs2 is the statistical feature set. Secondly, all the features are selected in the same way for Min, Mean, Mode and Std, in order to find the best features by the SFS algorithm. The best features (denoted as SFS_feature) are selected based on Eqs. (2) and (3) as below:

$$\delta \leq fs2, \tag{2}$$

$$\delta > fs2. \tag{3}$$

3.3 The feature set

After decreasing the dimensions of the features through the SFS, the new feature set is forwarded to the LS_SVM classifier. In this study, we obtain a feature set that has 2000 data points of 35 dimensions. These features are divided into two groups, which are the training set and the testing set. The training set is directed to train a classifier. The testing set is employed to evaluate the performance of the methodology and it is utilized as the input of the classifier.

3.4 Least square support vector machines

In this subsection, we briefly review some basic work on LS_SVMs for classification. LS_SVMs are proposed by Suykens and Vandewalle. LS_SVMs are the least square versions of SVMs, which are a set of related supervised learning methods that analyse data and recognize patterns. Moreover, they are used for classification and regression analysis [31]. In this research, the LS_SVM classifier with a radial basis function kernel is used for the classification of epileptic EEG signals. These classifiers can avoid the problem of convex quadratic programming from the classical SVMs by using a set of linear equations [8]. In this paper, the classification is performed by LS_SVMlab (version 1.8) toolbox in MATLAB[2] [32].

[2] http://www.esat.kuleuven.ac.be/sista/lssvmlab/.

3.5 Performance measures

This subsection presents assessing how the proposed method performs. The assessments include accuracy (also known as recognition rate), sensitivity (or recall) and specificity. The accuracy of a classifier is the percentage of the test set which is correctly classified by the classifier. The sensitivity is referred to the true positive rate which is the proportion of the positive set correctly identified.

The specificity is the true negative average which is the proportion of the negative set correctly identified. The following Eqs. (4)–(6) provide the definitions for the terms [33]:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}}, \tag{4}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{P}}, \tag{5}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{N}}, \tag{6}$$

where TP is the number of true positives, TN is the number of true negatives and P and N are the positive and negative samples, respectively.

4 Results and discussions

In this study, we involved two datasets: sets **A** and **E** as mentioned in Sect. 2. SRS technique was used to extract features from the datasets. This technique selected features randomly by choosing 10 samples from each dataset (sets **A** and **E**). A five subsamples were selected from each sample. From each subsample, nine statistical features, such as minimum, maximum, mean, median, mode, first quartile, third quartile, inter-quartile range and Std were extracted as aforementioned in Sect. 3.1.

A set of features obtained from the SRS included 2000 × 45 dimensions. These features were used in two different ways. Firstly, the statistical features were directly fed to the LS_SVM classifier and yielded the results, as shown in Table 1. Secondly, the SFS based on the criterion was employed to select the key features from the extracted features as mentioned in Sect. 3.2. As shown in the results, the good results of the best features are presented in Table 2. In Table 2, the good results are obtained by using the SRS algorithm and the SFS technique with the LS_SVM classifier depending on the best criterion chosen. Furthermore, the LS_SVM has two important parameters, which are γ and σ^2 which should be suitably selected for achieving a desirable performance too. The LS_SVM was affected by the value of these two parameters. This study trained the LS_SVM with different groups of the parameters γ and σ^2 to obtain best results. In this proposed method, we conducted with one group of the five EEG datasets and gained the best classification result with sets **A** and **E** when $\gamma = 10$ and $\sigma^2 = 1$ for the two methods applied in this paper. The results of the proposed method were compared with the results that were obtained from the SRS method and the LS_SVM classifier. The experimental results showed that our approach yielded 99.90 % classification accuracy for the epileptic EEG data. Table 3 gives a better view for the results by the two different classification methods. On the other hand, in this study, the evaluation of time complexity between the presented approach and the SRS was conducted.

The SRS_SFS_LS_SVM method took 0.16 s to classify the extracted features in Sect. 3.2. While the SRS_LS_SVM tackled the same features with 1.52 s as shown in Table 3. The performance of the proposed method is also compared with two existing methods in the literature. For fair comparison, the same dataset was used in comparison. The results show that the proposed method outperforms over the other two existing methods: a Huang–Hilbert transform and an artificial neural network model by Husain and Rao [22] and a ST and LS_SVM methods by

Table 1 Classification accuracy for epileptic EEG signals (sets **A** and **E**)

Statistical parameters	Results (%)
Accuracy	100
Sensitivity	100
Specificity	100

Table 2 Experimental results using different statistic features as the criterion

Choose criterion	Accuracy (%)	Sensitivity (%)	Specificity (%)
Mean ≥ fs2 (SFS_feature)	99.90	99.80	100.00
Mean ≤ fs2 (SFS_feature)	98.90	98.00	99.80
Max ≤ fs2 (SFS_feature)	97.20	100.00	94.40
Min ≥ fs2 (SFS_feature)	99.10	99.20	99.00
Mode ≥ fs2 (SFS_feature)	97.70	95.40	100.00
Median ≤ fs2 (SFS_feature)	95.30	92.80	97.80
Std ≥ fs2 (SFS_feature)	95.60	91.20	100.00

Table 3 Comparison of the results and time complexity for the proposed method with other methods

Methods	Accuracy (%)	Sensitivity (%)	Specificity (%)	Time (s)
SRS_LS_SVM	100.00	100.00	100.00	1.52
The proposed method with the best criterion (SRS_SFS_LS_SVM)	99.90	99.80	100.00	0.16

Table 4 Comparison of performance of our proposed method with two recently reported methods for sets **A** and **E** of the EEG epileptic database

Different methods	Accuracy (%)	Sensitivity (%)	Specificity (%)
The proposed method with the best criterion (SRS_SFS_LS_SVM)	99.90	99.80	100
A Huang–Hilbert transform and an artificial neural network model [22]	99.80	99.75	100
A sampling technique and LS_SVM method [21]	80.05	74.97	87.70

Siuly et al. [21]. The performance comparison of the proposed method with the two reported methods to classify sets **A** and **E** is shown in Table 4. Husain and Rao in 2014 applied a Huang–Hilbert transform and an artificial neural network model on sets **A** and **E** (the same datasets used in this paper). They achieved a 99.80 % classification accuracy. While Siuly et al. in 2009 obtained 80.05 % classification accuracy when they used a ST and the LS_SVM methods to classify the EEG signals for the same datasets. Moreover, the proposed method gains a 99.90 % classification accuracy for the same group of datasets. The results shown that the proposed technique in this paper has the potential to classify the EEG signals from healthy people and epileptic patients using the extracted and selected features from the SRS and SFS techniques.

5 Conclusions

This research concentrates on two classes of EEG signals from healthy people and epileptic patients. The study presents a SRS_SFS method to extract and select the key features for classifying EEG signals into two classes. The LS_SVM classifier is used to classify two-category EEG data after the feature extraction and selection. This method yields the results of 99.90, 99.80 and 100 % for classification accuracy, sensitivity and specificity, respectively. In addition, the proposed method is faster than the SRS technique. It means that the SRS_SFS is useful for extracting and selecting the EEG features. To sum up, the proposed method is very efficient for analysing and classifying epileptic EEG signals. It will be also useful for the classification of other biomedical data.

References

1. Buck D, Baker GA, Jacoby A et al (1997) Patients' experiences of injury as a result of epilepsy. Epilepsia 38(4):439–444
2. World Health Organization (WHO) (2011). Report: WHO. http://www.who.int/mediacentre/factssheets/fs999/en/index.html. Accessed Dec 2015
3. Mcgrogan N (1999) Neural network detection of epileptic seizures in the electroencephalogram. http://www.new.ox.ac.uk/~nmcgroga/work/transfer
4. Boer H, Engel J, Prilipko L (2005) Global campaign against epilepsy. Epilepsy Atlas 82–83
5. Iasemidis LD (2003) Epileptic seizure prediction and control. IEEE Trans Biomed Eng 50(5):549–558
6. Kumar TS, Kanhangad V, Pachori RB (2015) Classification of seizure and seizure-free EEG signals using local binary patterns. Biomed Signal Process Control 15:33–40
7. Adeli H, Zhou Z, Dadmehr N (2003) Analysis of EEG records in an epileptic patient using wavelet transform. J Neurosci Methods 123(1):69–87
8. Siuly S, Li Y (2015) Designing a robust feature extraction method based on optimum allocation and principal component analysis for epileptic EEG signal classification. Comput Methods Programs Biomed 119(1):29–42
9. Acharya UR, Sree SV, Swapna G et al (2013) Automated EEG analysis of epilepsy: a review. Knowl Based Syst 45:147–165
10. Barreiro PL, Albandoz JP (2001) Population and sample. Sampling techniques. Management mathematics for European schools, MaMaEusch (994342-CP-1-2001-1-DECOMENIUS-C21)
11. Wu F, Zhao Y (2005) Least squares support vector machine on Moret wavelet kernel function. In: International conference on neural networks and brain, 2005. ICNN&B'05. IEEE, Beijing, p 327–333
12. Shen C-P, Chen C-C, Hsieh S-L et al (2013) High-performance seizure detection system using a wavelet-approximate entropy-fSVM cascade with clinical validation. Clin EEG Neurosci 44:247–256

13. Gajic D, Djurovic Z, Gligorijevic J et al (2015) Detection of epileptiform activity in EEG signals based on time–frequency and non-linear analysis. Front Comput Neurosci. doi:10.3389/fncom.2015.00038
14. Martis RJ, Acharya UR, Tan JH et al (2012) Application of empirical mode decomposition (EMD) for automated detection of epilepsy using EEG signals. Int J Neural Syst 22(6):1250027
15. Chua KC, Chandran V, Acharya UR et al (2011) Application of higher order spectra to identify epileptic EEG. J Med Syst 35(6):1563–1571
16. Guo L, Rivero D, Dorado J et al (2011) Automatic feature extraction using genetic programming: an application to epileptic EEG classification. Expert Syst Appl 38(8):10425–10436
17. Ocak H (2009) Automatic detection of epileptic seizures in EEG using discrete wavelet transform and approximate entropy. Expert Syst Appl 36(2):2027–2036
18. Srinivasan V, Eswaran C, Sriraam N (2007) Approximate entropy-based epileptic EEG detection using artificial neural networks. IEEE Trans Inf Technol Biomed 11(3):288–295
19. Srinivasan V, Eswaran C, Sriraam N (2005) Artificial neural network based epileptic detection using time-domain and frequency-domain features. J Med Syst 29(6):647–660
20. Gajic D, Djurovic Z, Di Gennaro S et al (2014) Classification of EEG signals for detection of epileptic seizures based on wavelets and statistical pattern recognition. Biomed Eng Appl Basis Commun 26:1450021
21. Siuly S, Li Y, Wen P (2009) Classification of EEG signals using sampling techniques and least square support vector machines. In: Rough sets and knowledge technology. Springer, Berlin, pp 375–382
22. Husain SJ, Rao K (2014) An artificial neural network model for classification of epileptic seizures using Huang–Hilbert transform. Int J Soft Comput 5(3):23
23. Rückstieß T, Osendorfer C, Van Der Smagt P (2011) Sequential feature selection for classification. In: AI 2011: advances in artificial intelligence. Springer, Berlin, pp 132–141
24. Choi K-S, Zeng Y, Qin J (2012) Using sequential floating forward selection algorithm to detect epileptic seizure in EEG signals. In: 2012 IEEE 11th international conference on signal processing (ICSP). IEEE, Beijing, pp 1637–1640
25. EEG time series (Nov 2005). http://www.meb.unibonn.de/epileptologie/science/physik/eegdata.html. Accessed Nov 2015
26. Andrzejak RG, Lehnertz K, Mormann F et al (2001) Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: dependence on recording region and brain state. Phys Rev E 64:061907
27. Marcano-Cedeño A, Quintanilla-Domínguez J, Cortina-Januchs M et al (2010) Feature selection using sequential forward selection and classification applying artificial metaplasticity neural network. In: 36th annual conference on IEEE Industrial Electronics Society. USA, pp 2845–2850
28. Ferri F, Pudil P, Hatef M et al (1994) Comparative study of techniques for large-scale feature selection. Pattern Recognit Pract IV:403–413
29. Pudil P, Novovičová J, Kittler J (1994) Floating search methods in feature selection. Pattern Recognit Lett 15(11):1119–1125
30. Reunanen J (2003) Overfitting in making comparisons between variable selection methods. J Mach Learn Res 3:1371–1382
31. Suykens JA, Vandewalle J (1999) Least squares support vector machine classifiers. Neural Process Lett 9(3):293–300
32. LS-SVMlab toolbox (version 1.8). http://www.esat.kuleuven.ac.be/sista/lssvmlab/. Accessed Nov 2015
33. Han J, Kamber M, Pei J (2011) Data mining: concepts and techniques, 3rd edn. Elsevier, Amsterdam

Hadi Ratham Al Ghayab received his BSc Degree in Computer Science from Thi_Qar University in 2007 and MSc Degree in Information Technology from University Utara Malaysia in 2010. He is currently a PhD Student in the Faculty of Health, Engineering and Sciences, University of Southern Queensland Toowoomba. His research interests include biomedical signal analysis, data mining and image processing.

Yan Li is currently an Associate Professor of Computer Sciences in the Faculty of Health, Engineering and Sciences at the University of Southern Queensland, Australia. Her research interests are in the areas of Big Data Technologies, Artificial Intelligence, Biomedical Engineering, and Signal/Image Processing.

Shahab Abdulla received his BSc and MSc Degrees from University of Technology Baghdad and PhD from USQ. He is currently a Lecturer in Language Centre, University of Southern Queensland. His research interests are in the areas of biomedical engineering, complex medical engineering, networked system, intelligent control, computer control systems, robotics and mathematics research, etc.

Mohammed Diykh received his BSc Degree in Computer Science from Thi_Qar University and MSc Degree in Information Technology from Voronezh State University in 2002 and 2010, respectively. He is currently a PhD Student in the Faculty of Health, Engineering and Sciences, University of Southern Queensland Toowoomba. His research interests include biomedical signal analysis, data mining and image processing, etc.

Xiangkui Wan is a Professor of the school of Electrical and Electronic Engineering, Hubei University of Technology. He received MS Degree and PhD Degree in Mechanical and Electronic Engineering from the Chongqing University, in 2002 and 2005, respectively. From 2005 to 2008, he was with the Department of Information Engineering, Guangdong University of Technology, as a Lecturer. Since 2008, he was as an Associate Professor in the same Department. And since 2014, he is with the School of Electrical and Electronic Engineering, Hubei University of Technology. His professional research interests are in signal processing of biomedical signals, artifact and noise analysis and linear and nonlinear time-series analysis.

Appendix D

Matlab simulation code to detect epileptic seizures

A simulation code to detect epileptic seizures in EEG signals is presented. In this simulation code, some of functions used were from Matlab tool for network analysis[3]. The obtained results were analysed using Microssfot Excel and Matlab.

[3] http://strategic.mit.edu/downloads.php?page=matlab_networks

```
function []=Epilptic_identification()
clear all;
for i=1:100
   N=[];
   if i<10
      N= [ 'C:\Users\U1054786\Documents\Sleep
Database\Epileptic_Classification_Mohammed_code\Data_set\Set B\O00' num2str(i) '.txt'];
   else
     N=[ 'C:\Users\U1054786\Documents\Sleep
Database\Epileptic_Classification_Mohammed_code\Data_set\Set B\O0' num2str(i) '.txt'];
%      elseif (i==100)
%    N=[ 'C:\Users\U1054786\Downloads\EEG record\Set A\S0' num2str(i) '.txt']
   end
   if (i==100)
      N=[ 'C:\Users\U1054786\Documents\Sleep
Database\Epileptic_Classification_Mohammed_code\Data_set\Set B\O' num2str(i) '.txt'];end;
A = importdata(N);
Set_B(:,i)=A;
end
N=[];
i=[];
% End of Reading Data


disp('Set B was read, press any key to containue reading other sets!')  % Press a key here.You can see
the message 'Paused: Press any key' in        % the lower left corner of MATLAB window.
pause;
%%%%%%%%%%%%%%%%%%%%%Read Set
A%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:100
   N=[];
   if i<10
      N= [ 'C:\Users\U1054786\Documents\Sleep
Database\Epileptic_Classification_Mohammed_code\Data_set\Set A\Z00' num2str(i) '.txt'];
   else
     N=[ 'C:\Users\U1054786\Documents\Sleep
Database\Epileptic_Classification_Mohammed_code\Data_set\Set A\Z0' num2str(i) '.txt'];
%      elseif (i==100)
%    N=[ 'C:\Users\U1054786\Downloads\EEG record\Set A\S0' num2str(i) '.txt']
   end
   if (i==100)
      N=[ 'C:\Users\U1054786\Documents\Sleep
Database\Epileptic_Classification_Mohammed_code\Data_set\Set A\Z' num2str(i) '.txt'];end;
AA= importdata(N);
Set_A(:,i)=AA;
end
disp('Set A was read, press any key to containue reading other sets!')  % Press a key here.You can see
the message 'Paused: Press any key' in        % the lower left corner of MATLAB window.
pause;
N=[];
i=[];
AA=[];

%%%%%%%%%%%%%%%%%%%%%% Read Set
D%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:100
   N=[];
```

```
   if i<10
      N= [ 'C:\Users\U1054786\Documents\Sleep
Database\Epileptic_Classification_Mohammed_code\Data_set\Set D\F00' num2str(i) '.txt'];
   else
     N=[ 'C:\Users\U1054786\Documents\Sleep
Database\Epileptic_Classification_Mohammed_code\Data_set\Set D\F0' num2str(i) '.txt'];
%      elseif (i==100)
%    N=[ 'C:\Users\U1054786\Downloads\EEG record\Set A\S0' num2str(i) '.txt']
   end
   if (i==100)
      N=[ 'C:\Users\U1054786\Documents\Sleep
Database\Epileptic_Classification_Mohammed_code\Data_set\Set D\F' num2str(i) '.txt'];end;
AA= importdata(N);
Set_D(:,i)=AA;
end
disp('Set D was read, press any key to containue reading other sets!')  % Press a key here.You can see
the message 'Paused: Press any key' in        % the lower left corner of MATLAB window.
pause;
%%%%%%%%%%%%%%%%%%%%%Read Set
C%%%%%%%%%%%%%%%%%%%%%%%%
AA=[];
N=[];
i=[];
for i=1:100
   N=[];
   if i<10
      N= [ 'C:\Users\U1054786\Documents\Sleep
Database\Epileptic_Classification_Mohammed_code\Data_set\Set C\N00' num2str(i) '.txt'];
   else
     N=[ 'C:\Users\U1054786\Documents\Sleep
Database\Epileptic_Classification_Mohammed_code\Data_set\Set C\N0' num2str(i) '.txt'];
%      elseif (i==100)
%    N=[ 'C:\Users\U1054786\Downloads\EEG record\Set A\S0' num2str(i) '.txt']
   end
   if (i==100)
      N=[ 'C:\Users\U1054786\Documents\Sleep
Database\Epileptic_Classification_Mohammed_code\Data_set\Set C\N' num2str(i) '.txt'];end;
AA= importdata(N);
Set_C(:,i)=AA;
end
disp('Set C was read, press any key to containue reading other sets!')  % Press a key here.You can see
the message 'Paused: Press any key' in        % the lower left corner of MATLAB window.
pause;
i=[];
N=[];
AA=[];
for i=1:100
   N=[];
   if i<10
      N= [ 'C:\Users\U1054786\Documents\Sleep
Database\Epileptic_Classification_Mohammed_code\Data_set\Set E\S00' num2str(i) '.txt'];
   else
     N=[ 'C:\Users\U1054786\Documents\Sleep
Database\Epileptic_Classification_Mohammed_code\Data_set\Set E\S0' num2str(i) '.txt'];
%      elseif (i==100)
%    N=[ 'C:\Users\U1054786\Downloads\EEG record\Set A\S0' num2str(i) '.txt']
   end
   if (i==100)
      N=[ 'C:\Users\U1054786\Documents\Sleep
Database\Epileptic_Classification_Mohammed_code\Data_set\Set E\S' num2str(i) '.txt'];end;
```

```
AA= importdata(N);
Set_E(:,i)=AA;
end
disp('Set E was read, press any key to start dimention reduction pahse!')  % Press a key here.You can see the message 'Paused: Press any key' in        % the lower left corner of MATLAB window.
pause;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Dimension reduction
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%extraction%%
%%%%%%%%%%%%%%%%%%%

Set_A1=Dimention_reduct(Set_A);
Set_B1=Dimention_reduct(Set_B);
Set_C1=Dimention_reduct(Set_C);
Set_D1=Dimention_reduct(Set_D);
Set_E1=Dimention_reduct(Set_E);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('Set E was read, press any key to start graphs attributes extraction pahse!')  % Press a key here.You can see the message 'Paused: Press any key' in        % the lower left corner of MATLAB window.
pause;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Graphs attributes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%extraction%%
%%%%%%%%%%%%%%%%%%%
G_Set_A=Gra_Fe_Extr(Set_A1);
G_Set_B=Gra_Fe_Extr(Set_B1);
G_Set_C=Gra_Fe_Extr(Set_C1);
G_Set_D=Gra_Fe_Extr(Set_D1);
G_Set_E=Gra_Fe_Extr(Set_E1);
G_Set_A=G_Set_A';
G_Set_B=G_Set_B';
G_Set_C=G_Set_C';
G_Set_D=G_Set_D';
G_Set_E=G_Set_E';
G_Set_A_B=[G_Set_A;G_Set_B];


G_Set_C_D=[G_Set_C;G_Set_D];
G_Set_A_C_D=[G_Set_A;G_Set_C;G_Set_D];
disp('data reduction has been done, press any key to start classifciation pahse!')  % Press a key here.You can see the message 'Paused: Press any key' in        % the lower left corner of MATLAB window.
pause;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Classifciation phase

disp('Classifciation results using LS_SVM')
disp('           Accuracy      ')
disp('_____________________________________________')

ACC_LS(1)=LS_SVM_classifciation(G_Set_A,G_Set_E);
ACC_LS(2)=LS_SVM_classifciation(G_Set_E,G_Set_B);
ACC_LS(3)=LS_SVM_classifciation(G_Set_C,G_Set_E);
ACC_LS(4)=LS_SVM_classifciation(G_Set_D,G_Set_E);
ACC_LS(5)=LS_SVM_classifciation(G_Set_A,G_Set_E);

disp('   A vs E   B vs E  C vs E  D vs E' ); disp(ACC_LS);
disp('-----------------------------------------------------');
```

```
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%
 %%%%%%%%%%%Display Classifciation Results on screen%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%

disp('Classifciation results using Naive')
disp('           Accuracy    ')
disp('----------------------------------------------------');
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ACC_NAVE(1)=Nai_classification(G_Set_A,G_Set_E);
ACC_NAVE(2)=Nai_classification(G_Set_B,G_Set_E);
ACC_NAVE(3)=Nai_classification(G_Set_E,G_Set_D);
ACC_NAVE(4)=Nai_classification(G_Set_C,G_Set_E);
ACC_NAVE(5)=Nai_classification(G_Set_A_B,G_Set_E);
ACC_NAVE(6)=Nai_classification(G_Set_C_D,G_Set_E);
ACC_NAVE(7)=Nai_classification(G_Set_A_C_D,G_Set_E);
disp('Classifciation results using Naive')
disp('           Accuracy    ')
disp('____________________________________________')
disp('   A vs E   B vs E  C vs E  D vs E  A&B vs E C&D vs E  A,C&D vs E' ); disp(ACC_NAVE);
disp('----------------------------------------------------');
end


 function [ Y ] = Dimention_reduct( M )
p=M;
[n m]=size(p);
for i=1:m
  K=p(:,i);
  KF=K(1:4096);
  K1=reshape(KF,1024,4)';
    for j=1: 4
       N=K1(j, :);
       LP=reshape(N,32,[])';
       for r=1:32;
        FStage1(r,:)=mainpoint(LP(r,:));
       end
       YY(j,:)=reshape(FStage1',1,[]);
    end


     Y(:,i)=reshape(YY',1,[]);
end


end


function [ XX ] = Gra_Fe_Extr(Y )
[k1 k2]=size(Y);
```

```
F=Y;
for k=1:k2
  F1=F(:,k);
D=Wei_Graph_constract(F1');
A=Adj_mat1(D);
Degree_matrix=averageDegree(A);
Deg_n(k)=Degree_matrix;
Cluster_coff_matrix=clustering1(A, 'undirected');
Clus_nod(k)=mean(Cluster_coff_matrix);
TTT=simpleSpectralPartitioning(A);
TRTR=modularityMetric(TTT, A);
Mod_n(k)=TRTR;
Bet_m=betweenness_centrality(A)
end
XX=[Clus_nod;Deg_n;Mod_n; Bet_m];
end


function [bc,E] = betweenness_centrality(A,varargin)
% BETWEENNESS_CENTRALITY Compute the betweenness centrality for vertices.
%
% bc = betweenness_centrality(A) returns the betweenness centrality for
% all vertices in A.
%
% [bc,E] = betweenness_centrality(A) returns the betweenness centrality for
% all vertices in A along with a sparse matrix with the centrality for each
% edge.
%
% This method works on weighted or weighted directed graphs.
% For unweighted graphs (options.unweighted=1), the runtime is O(VE).
% For weighted graphs, the runtime is O(VE + V(V+E)log(V)).
%
% ... = betweenness_centrality(A,...) takes a set of
% key-value pairs or an options structure.  See set_matlab_bgl_options
% for the standard options.
%   options.unweighted: use the slightly more efficient unweighted
%     algorithm in the case where all edge-weights are equal [{0} | 1]
%   options.ec_list: do not form the sparse matrix with edge [{0} | 1]
%   options.edge_weight: a double array over the edges with an edge
%       weight for each node, see EDGE_INDEX and EXAMPLES/REWEIGHTED_GRAPHS
%       for information on how to use this option correctly
%       [{'matrix'} | length(nnz(A)) double vector]
%
% Note: the edge centrality can also be returned as an edge list using the
% options.ec_list options.  This option can eliminate some ambiguity in the
% output matrix E when the edge centrality of an edge is 0 and Matlab drops
% the edge from the sparse matrix.
%
% Note: if the edge centrality matrix E is not requested, then it is not
% computed and not returned.  This yields a slight savings in computation
% time.
%
%
%% History
%  2006-04-19: Initial version
%  2006-05-31: Added full2sparse check
%  2007-03-01: Added edge centrality output
%  2007-04-20: Added edge weight option
%  2007-07-09: Restricted input to positive edge weights
%  2007-07-12: Fixed edge_weight documentation.
```

```
%  2008-10-07: Changed options parsing
%%

[trans check full2sparse] = get_matlab_bgl_options(varargin{:});
if full2sparse && ~issparse(A), A = sparse(A); end

options = struct('unweighted', 0, 'ec_list', 0, 'edge_weight', 'matrix');
options = merge_options(options,varargin{:});

% edge_weights is an indicator that is 1 if we are using edge_weights
% passed on the command line or 0 if we are using the matrix.
edge_weights = 0;
edge_weight_opt = 'matrix';

if strcmp(options.edge_weight, 'matrix')
   % do nothing if we are using the matrix weights
else
   edge_weights = 1;
   edge_weight_opt = options.edge_weight;
end

if check
   % check the values
   if options.unweighted ~= 1 && edge_weights ~= 1
      check_matlab_bgl(A,struct('values',1,'noneg',1));
   else
      check_matlab_bgl(A,struct());
   end
   if edge_weights && any(edge_weights < 0)
      error('matlab_bgl:invalidParameter', ...
            'the edge_weight array must be non-negative');
   end
end

if trans
   A = A';
end

weight_arg = options.unweighted;
if ~weight_arg
   weight_arg = edge_weight_opt;
else
   weight_arg = 0;
end
if nargout > 1
   [bc,ec] = betweenness_centrality_mex(A,weight_arg);

   [i j] = find(A);
   if ~trans
      temp = i;
      i = j;
      j = temp;
   end

   if options.ec_list
      E = [j i ec];
   else
      E = sparse(j,i,ec,size(A,1),size(A,1));
   end
```

```
else
   bc = betweenness_centrality_mex(A,weight_arg);
end

% Computing the modularity for a given module/commnunity partition.
% Defined as: Q=sum_over_modules_i (eii-ai^2) (eq 5) in Newman and Girvan.
% eij = fraction of edges that connect community i to community j, ai=sum_j (eij)
%
% Source: Newman, Girvan, "Finding and evaluating community structure in networks"
%         Newman, "Fast algorithm for detecting community structure in networks"
%
% INPUTs: adjacency matrix, nxn
%         set of modules as cell array of vectors, ex: {[1,2,3],[4,5,6]}
% OUTPUTs: modularity metric, in [-1,1]
%
% Note: This computation makes sense for undirected graphs only.
% Other functions used: numEdges.m
% GB: last updated, October 16, 2012

function Q=modularityMetric(modules,adj)

nedges=numEdges(adj); % total number of edges

Q = 0;
for m=1:length(modules)

  e_mm=numEdges(adj(modules{m},modules{m}))/nedges;
  a_m=sum(sum(adj(:,modules{m})))/nedges - e_mm;
  Q = Q + (e_mm - a_m^2);

end


% Uses the sorted fiedler vector to assign nodes to groups.
%
% INPUTS: adjacency matrix (nxn), k - desired number
%         of nodes in groups [n1, n2, ..], [optional].
%         The default k is 2.
% OUTPUTs: modules - vector of size 1x(number of desired modules);
%                    each entry contains the number of nodes in
%                    that module
%
% Example:
% simpleSpectralPartitioning(randomModularGraph(100,4,0.15,0.9),
%                                        [25 25 25 25])
% Other functions used: fiedlerVector.m
% Note: To save the plot at the end of the routine, uncomment:
%                  print filename.pdf (or filename.extension)

function modules = simpleSpectralPartitioning(adj,k)

% find the Fiedler vector: eigenvector corresponding to the second smallest eigenvalue of the
Laplacian matrix
fv = fiedlerVector(adj);
[~,I]=sort(fv);

% depending on k, partition the nodes
if nargin==1

   modules{1}=[]; modules{2}=[];
```

```
    % choose 2 groups based on signs of fv components
    for v=1:length(fv)
        if fv(v)>0; modules{2} = [modules{2}, v]; end
        if fv(v)<=0; modules{1} = [modules{1}, v]; end
    end
end

if nargin==2

  k = [0 k];  % adding 0 to aid indexing in line 43

  for kk=1:length(k)

    modules{kk}=[];
    for x=1:k(kk);
        modules{kk} = [modules{kk} I( x+sum(k(1:(kk-1))) )];
    end

  end

  modules = modules(2:length(modules));  % removing the "0" module
end

set(gcf,'Color',[1 1 1])
subplot(1,2,1)
plot(fv(I),'k.');
xlabel('index i')
ylabel('fv(i)')
title('sorted fiedler vector')
axis tight
axis square

subplot(1,2,2)
spy(adj(I,I),'k.')
axis square
title('sorted adjacency matrix')

%print spec_part_example.pdf

 % Compute the average degree of a node in a graph, defined as
% 2 times the number of edges divided by the number of nodes
%          (every edge is counted towards the degrees twice).
%
% Inputs: adjacency matrix, nxn
% Outputs: float, the average degree, a number between 0 and max(sum(adj))
%
% Note: The average degree is related to the link density, namely:
%       link_density = ave_degree/(n-1), where n is the number of nodes
%
% Other routines used: numNodes.m, numEdges.m
% GB: last update, September 20, 2012

function k=averageDegree(adj)

k=2*numEdges(adj)/numNodes(adj);



function d = Graph_construct(w,varargin)
if ischar(w)
```

```
  d = nnet7.weight_fcn(mfilename,w,varargin{:});
  return
end

% Distance
if (nargin < 2) || ~isnumeric(varargin{1})
  d = dist.distance(w,varargin{:});
else
  % Apply Weight
  d = dist.apply(w,varargin{:});
end
function out1 = weight_fcn(fcn,varargin)
%NNET7.WEIGHT_FCN Weight function NNET 7.0 backward compatibility

info = nnModuleInfo(fcn);
in1 = varargin{1};
switch(in1)

 case 'apply'
   [args,param,nargs] = nnparam.extract_param(varargin(2:end),info.defaultParam);
   if nargs < 2, error(message('nnet:Args:NotEnough')); end
   w = nntype.matrix_data('format',args{1},'Weight');
   p = nntype.matrix_data('format',args{2},'Inputs');
   out1 = info.apply(w,p,param);

 case 'dz_dp'
   [args,param,nargs] = nnparam.extract_param(varargin(2:end),info.defaultParam);
   if nargs < 2, error(message('nnet:Args:NotEnough')); end
   w = nntype.matrix_data('format',args{1},'Weight');
   p = nntype.matrix_data('format',args{2},'Inputs');
   if nargs < 3
     z = info.apply(w,p,info.defaultParam);
   else
     z = nntype.matrix_data('format',args{3},'Net input');
   end
   out1 = info.dz_dp(w,p,z,param);

 case 'dz_dw'
   [args,param,nargs] = nnparam.extract_param(varargin(2:end),info.defaultParam);
   if nargs < 2, error(message('nnet:Args:NotEnough')); end
   w = nntype.matrix_data('format',args{1},'Weight');
   p = nntype.matrix_data('format',args{2},'Inputs');
   if nargs < 3
     z = info.apply(w,p,info.defaultParam);
   else
     z = nntype.matrix_data('format',args{3},'Net input');
   end
   out1 = info.dz_dw(w,p,z,param);

 case {'info','subfunctions'}
   out1 = info;

 case 'defaultParam'
   out1 = info.defaultParam;

 case 'fpnames'
   out1 = fieldnames(info.defaultParam);

 case 'name'
   out1 = info.name;
```

```
  case 'size',
    % this('size',numNeurons,numInputs')
    % Weight size
    [args,param,nargs] = nnparam.extract_param(varargin(2:end),info.defaultParam);
    if nargs < 2, error(message('nnet:Args:NotEnough')); end
    s = nntype.pos_int_scalar('format',args{1},'Layer size');
    r = nntype.pos_int_scalar('format',args{2},'Input size');
    out1 = info.size(s,r,param);

  % NNET 6.0 Compatibility

  case 'pfullderiv', out1 = info.inputDerivType;
  case 'wfullderiv', out1 = info.weightDerivType;
  case 'dp'
    if nargin < 4,error(message('nnet:Args:NotEnough')); end
    if nargin < 6, varargin{5} = info.defaultParam; end
    out1 = info.dz_dp(varargin{2:5});
  case 'dw'
    if nargin < 4,error(message('nnet:Args:NotEnough')); end
    if nargin < 6, varargin{5} = info.defaultParam; end
    out1 = info.dz_dw(varargin{2:5});

  case 'simulinkParameters'
    out1 = info.simulinkParameters(varargin{2:end});
end

 function [ ACC ] = LS_SVM_classifciation( XX,XX_1 )

DD=[XX;XX_1]

[n m]=size(XX);
[n1 m1]=size(XX_1);
[n3 m3]=size(DD);
R=m3+1;
DD(1:n,R)=1;
DD(n1+1:n3,R)=-1;

nrows = size(DD,1);
r80 = round(0.52 * nrows);
trainingset = DD(1:r80,:,:);
testset = DD(r80+1:end,:,:);

[k1 k2]=size(trainingset);
x = trainingset(:,1:k2-1);
y = trainingset(:,k2);
% test set
Xtest= testset(:,1:k2-1)
Ytest=testset(:,k2);
size(x)
size(y)
%%
gam=8;
sig2=1;
type='classification';
L_fold=10;

[gam, sig2] = tunelssvm({x,y,'c',[],[],'RBF_kernel'}, 'simplex',...
       'leaveoneoutlssvm', {'misclass'});
```

```
[alpha,b] = trainlssvm({x,y,type,gam,sig2,'RBF_kernel'});
Yh=simlssvm({x,y,type,gam,sig2,'RBF_kernel'},{alpha,b},Xtest);

[perc,n,which]=misclass(Ytest,Yh);
n
perc
[C,order] = confusionmat(Ytest,Yh);
C
order

%%
Y_latent=latentlssvm({x,y,type,gam,sig2,'RBF_kernel'},{alpha,b},x);
[area,se,thresholds,oneMinusspec,sens,TN,TP,FN,FP]=roc(Y_latent,y);
%[thresholds oneMinusspec sens ];

%%
Accuracy=TN/(TN+FP)*100;
ACC=Accuracy;
end

function [ ACC ] = Nai_classification( XX,XX_1 )

 distr='normal';
distr='kernel';
DD=[XX;XX_1]

[n m]=size(XX);
[n1 m1]=size(XX_1);
[n3 m3]=size(DD);
R=m3+1;
DD(1:n,R)=1;
DD(n1+1:n3,R)=-1;

nrows = size(DD,1);
r80 = round(0.75 * nrows);
trainingset = DD(1:r80,:,:);
testset = DD(r80+1:end,:,:);

[k1 k2]=size(trainingset);
x = trainingset(:,1:k2-1);
y = trainingset(:,k2);
% test set
u= testset(:,1:k2-1)
v=testset(:,k2);

Y=DD(:,R);

 %  X = [X1;X2];
%  [n m]=size(X1);
% [n1 m1]=size(X2);
%  Y (1:n)=1;
%  Y (n+1:n+n1)=-1;
%  Y=Y';

% Create a cvpartition object that defined the folds
c = cvpartition(Y,'holdout',.2);

% Create a training set
% x = trainingset;
```

```
% y = Xtest;
% % test set
% u=testset ;
% v=Ytest;
yu=unique(y);
nc=length(yu); % number of classes
ni=size(x,2); % independent variables
ns=length(v); % test set

% compute class probability
for i=1:nc
   fy(i)=sum(double(y==yu(i)))/length(y);
end

switch distr

   case 'normal'

      % normal distribution
      % parameters from training set
      for i=1:nc
         xi=x((y==yu(i)),:);
         mu(i,:)=mean(xi,1);
         sigma(i,:)=std(xi,1);
      end
      % probability for test set
      for j=1:ns
         fu=normcdf(ones(nc,1)*u(j,:),mu,sigma);
         P(j,:)=fy.*prod(fu,2)';
      end

   case 'kernel'

      % kernel distribution
      % probability of test set estimated from training set
      for i=1:nc
         for k=1:ni
            xi=x(y==yu(i),k);
            ui=u(:,k);
            fuStruct(i,k).f=ksdensity(xi,ui);
         end
      end
      % re-structure
      for i=1:ns
         for j=1:nc
            for k=1:ni
               fu(j,k)=fuStruct(j,k).f(i);
            end
         end
         P(i,:)=fy.*prod(fu,2)';
      end

   otherwise

      disp('invalid distribution stated')
      return

end

% get predicted output for test set
```

```
[pv0,id]=max(P,[],2);
for i=1:length(id)
    pv(i,1)=yu(id(i));
end

% compare predicted output with actual output from test data
confMat=myconfusionmat(v,pv);
conf=sum(pv==v)/length(pv);
ACC=conf*100;

end
```

Appendix E

A Matlab Simulation code to classify Sleep EEG signals

A simulation code to identify EEG sleep stages is presented. In this simulation code, some of functions used were from Matlab tool for network analysis[4]. The obtained results were analysed using Microssfot Excel and Matlab.

[4] http://strategic.mit.edu/downloads.php?page=matlab_networks

```
function[]= SLeep_Classifciation_based_SF_G()
% Receives:
% dose not receive any input.
%
% Returns:
% Two Matrix of classification results: Accuracy, Sensitivity// each matrix is included the
classification results of each pair of sleep stages.
%
%
%
% The major steps of the methodology:
%   1. Read EEG recordings {the data set is downloaded from
%   https://www.physionet.org/physiobank/database/sleep-edf/}
%   2. Segment each EEG recording into epochs of 3000 data points, 30
%   seconds
%   3. Each EEG segment also is divided into 75 sub-segments
%   4. A set of statistical features are extracted from each sub-segment, as a result, each EEG segment
is represented by a vector of 900 statistical features (75x12).
%   5. Each vector of features is mapped into an undirected graph.
%   6. A set of graph attributes is extracted and fed to k-means
%   7. In the classicisation phase, at each time, one pair of sleep stages is classified.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% this code was designed by Mohammed DiyKh in 2016, to classify EEG sleep
% stages based on Statistical features and Graphs.
%this code explains all necessary step to analyse an EEG recording


clear all;
% reading EEG data phase%
[filename pathname]=uigetfile({'*.rec';'*.edf'},'Select EEG recording'); % selecting the EEG.rec file%
fulname=strcat(pathname, filename);
[Su2 Su22]=edfread(fulname);
A=filename(1:9);
fileID = fopen(A,'w');
fprintf(fileID,'%6s %12s\n','Accuracy','Senstivity');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Read the EEG.rec file, the channel pz-oz (No.2) is selected in this study%
V=Su22(2,:);
[filename pathname]=uigetfile({'*.hyp'},'Select EEG HYP file');% selecting the EEG.Hyp file%
fulname=strcat(pathname, filename);
[pathstr,name,ext] = fileparts(filename)
if ext=='.hyp'
[Y Y1]=edfread(fulname);
else
   Y1=importdata('1_1.txt');
end
% Read the EEG.Hyp file%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Samp=length (Y1);
LL=Samp*3000;
sleep_data=reshape(Y1, 3000,[])';
%the EEG signals are divided into smaller segments.
```

```
%The interval of each segment is 30 seconds (3000 data points)
% according to the Rechtschaffen and Kales
[AW,S1,S2,S3,S4,REM]=Sleep_EEG_Seg(sleep_data, Y1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Graphs Features Extraction using Graphs_features Function from each sleep stages, more details is presnted in this function%
 AW11=Graphs_Features(AW);
 S11=Graphs_Features(S1);
 S22=Graphs_Features(S2);
 S33=Graphs_Features(S3);
 S44=Graphs_Features(S4);
 REMM=Graphs_Features(REM);

 %Classifying the extracted Graphs attributes of each sleep stages  using K-means classifier%
 SWS=[S44;S33];
 [Acc(1), Sen(1)]=K_Means_classi(S11,S22);
 [Acc(2), Sen(2)]=K_Means_classi(AW11,S33);
 [Acc(3), Sen(3)]=K_Means_classi(AW11,REMM);
 [Acc(4), Sen(4)]=K_Means_classi(S22,S11);
 [Acc(5), Sen(5)]=K_Means_classi(S33,AW11);
 Result_f=[Acc;Sen];

 s= dir(A);
if s.bytes == 0
 fileID = fopen(A,'w');
fprintf(fileID,'%6s %12s\n','Accuracy','Senstivity');
fprintf(fileID,'%6.2f %12.8f\n',Result_f);
fclose(fileID);
else
fileID = fopen(A,'a');
fprintf(fileID,'%6.2f %12.8f\n',Result_f);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%%%%%%%Display Classifciation Results on screen%%%%%%%%

disp('Classifciation results using SGSKM')
disp('Accuracy')
disp('_________')
disp('   S1 vs S2   AW vs S1  AW vs REM  AW vs S2' ); disp(Acc);
disp('-----------------------------------------------------');
disp('Sensitivity')
disp('   S1 vs S2   AW vs S1  AW vs REM  AW vs S2' ); disp(Sen);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  End


function [ XX ] = Graphs_Features ( X )

%  This Function calcauted graph features for each segment in Matrix X
% Receives:
% EEG signal X
%
% Returns:
%  A vector of graph attributes XX
```

```
%  Each EEG segment is partitioned into 75 sub-segments.
%  Secondly, 12 statistical features are extracted from each sub-segment.
%  As a results, each segment reduced from 3000 datapoints into 900 statistical features
%  This code was designed by MOhammed DiyKh in 2016, to classify EEG sleep
%  Stages.
Feat_sleep=X;
[index1, index2]=size(Feat_sleep);
% meaures the length of input matrix%
for i=1:index1
  R1=Feat_sleep(i,:);
  % Take the first EEG segment that has 3000 data points%

  R2=reshape(R1, 40,[])';
  %each segment is separated into 75 sub-segments with each sub-segment includes 40 data points,
  % total of 75X12
for j=1:75
   T1=Statistical_Fea(R2(j,:));
   % 12 statistical features are extracted from each sub-segments using
   % Statistical_feature
   T2(j,:)=T1;
end
RRR=reshape(T2', [],1);
% each segment is reduced from 3000 data points into 900 statistical features
WW=Graph_const(RRR');
% Consdider each Statistical features as node and calculated distance among them
WW1=Adj_Cal(WW);
Jacc_1=Jaccard(WW1);
% Calculated Adjacency Matrix%
Jacc_vect=Jacc_1;
% calculate Jaccard%
% calculate average degree%
Cluster_coff=clustering1(WW1, 'undirected');
% calculate Clustering Coeffiecnets
Cluster_coff_matrix=Cluster_coff';
XXX=[Jacc_vect,Cluster_coff_matrix, Degree_2];
Final_M(i,:)=XXX;
 end
XX=Final_M;

end


function [AW1,S11,S22,S33,S44,REMM ] = Sleep_EEG_Seg(Data, index )
sleep_data=Data;
Y1=index;
R0=1;
   R1=1;
   R2=1;
   R3=1;
   R4=1;
   R5=1;
for i=1:length(Y1)
   fi=Y1(i);

   switch fi
      case 0
       AW1(R0,:)=sleep_data(i,:);
       R0=R0+1;
```

```
      case 1
           S11(R1,:)=sleep_data(i,:);
           R1=R1+1;
       case 2
           S22(R2,:)=sleep_data(i,:);
           R2=R2+1;
       case 3
           S33(R3,:)=sleep_data(i,:);
            R3=R3+1;
        case 4
           S44(R4,:)=sleep_data(i,:);
           R4=R4+1;
       case 5
           REMM(R5,:)=sleep_data(i,:);
             R5=R5+1;
   end
end

end


function [X ] = Statistical_Fea( Y )
% Receives:
% A set of EEG data Y
%
% Returns:
% a vector of statistical features X
 X(1)=max(Y);
 X(2)=range(Y);
 X(3)=std(Y);
 X(4)=min(Y);
 X(5)=mean(Y);
 X(6)=mode(Y);
 X(7)=median(Y);
 [q1 q2 q3 m g d s]=quartile(Y);
 X(8)=q1;
 X(9)=q3;
 X(10)=var(Y);
X(11)=skewness(Y);
X(12)= kurtosis(Y);
end

function [ Y ] = Jaccard  P )
% Receives:
% Adjacency matrix p
%
% Returns:
% Jaccard coefficient's Y
A=P;
[n m]=size (A);
for i=1:n
   mm1=0;
   for j=1:m
   if (A(i,j)==1);
     mm1=mm1+1;
    nn{i}.mm(mm1)=j;
   end
   end
```

```
end
for i=1:n
   for j=1:m
      if (i~=j)
      p= nn{i}.mm;
      p2= nn{j}.mm;
      interse=intersect(p,p2);
      uni=union(p,p2);
      L1=numel(interse);
      L2=numel(uni);
      Jacca_coff(i,:)=L1/L2;

      end
   end
end
Y=Jacca_coff;
end



function coeff = clustering(A, type)
% CLUSTERING   Calculates the clustering coefficient for each node from an adjacency matrix.
%   The clustering coefficient for each node in the graph is calculated
%   from the given adjacency matrix. If the type is given, then the
%   adjacency matrix is assumed to represent a graph of that type (either
%   directed or undirected). If the type is not given, the graph is assumed
%   to be undirected if the adjaceny matrix is symmetric, and directed
%   otherwise.
%
%      USAGE:
%         coeff = clustering(A)
%         coeff = clustering(A, 'directed')
%         coeff = clustering(A, 'undirected')
%
%      coeff
%         The column vector containing the clustering coefficient of each node.
%
%      A
%         The adjacency matrix.
%
%      type = 'directed'/'undirected'
%         The type of graph the adjacency matrix represents. If not
%         given, the graph is assumed to be undirected if it is
%         symmetric.

n = size(A,1);

if (nargin>1)
   if strcmp(type,'directed')
      digraph = true;
   elseif strcmp(type,'undirected')
      digraph = false;
   else
      error('Type must be either "directed" or "undirected"')
   end
else
   if all(all(A == A'))
      digraph = false;
   else
      digraph = true;
```

```
    end
end

if digraph
    c = sum((A^2) .* A, 2);
else
    c = sum((A^3) .* eye(n), 2);
end

% Calculate the out degree of the nodes
out = sum(A,2);

% Calculate the clustering coefficient
s = warning('off','MATLAB:divideByZero');
coeff = c ./ (out .* (out - 1));
warning(s);

% Remove the Inf's from the possible divide by 0
coeff(out == 0) = 0;
coeff(out == 1) = 0;

% Compute the average degree of a node in a graph, defined as
% 2 times the number of edges divided by the number of nodes
%          (every edge is counted towards the degrees twice).
%
% Inputs: adjacency matrix, nxn
% Outputs: float, the average degree, a number between 0 and max(sum(adj))
%
% Note: The average degree is related to the link density, namely:
%       link_density = ave_degree/(n-1), where n is the number of nodes
%
% Other routines used: numNodes.m, numEdges.m
% GB: last update, September 20, 2012

function k=averageDegree(adj)

k=2*numEdges(adj)/numNodes(adj);
function [Acc,rand_index,match]=AccMeasure(T,idx)
%Measure percentage of Accuracy and the Rand index of clustering results
% The number of class must equal to the number cluster
% This Code was designed by Mohammed Diykh University of Southern
% Queensland

%Output
% Acc = Accuracy of clustering results
% rand_index = Rand's Index,  measure an agreement of the clustering results
% match = 2xk mxtrix which are the best match of the Target and clustering results

%Input
% T = 1xn target index
% idx =1xn matrix of the clustering results

% EX:
% X=[randn(200,2);randn(200,2)+6,;[randn(200,1)+12,randn(200,1)]];
T=[ones(200,1);ones(200,1).*2;ones(200,1).*3];
% idx=kmeans(X,3,'emptyaction','singleton','Replicates',5);
%  [Acc,rand_index,match]=AccMeasure(T,idx)
k=max([T(:);idx(:)]);
n=length(T);
```

```
for i=1:k
   temp=find(T==i);
   a{i}=temp; %#ok<AGROW>
end

b1=[];
t1=zeros(1,k);
for i=1:k
   tt1=find(idx==i);
   for j=1:k
     t1(j)=sum(ismember(tt1,a{j}));
   end
   b1=[b1;t1]; %#ok<AGROW>
end
   Members=zeros(1,k);

P = perms((1:k));
   Acc1=0;
for pi=1:size(P,1)
   for ki=1:k
      Members(ki)=b1(P(pi,ki),ki);
   end
   if sum(Members)>Acc1
      match=P(pi,:);
      Acc1=sum(Members);
   end
end

rand_ss1=0;
rand_dd1=0;
for xi=1:n-1
   for xj=xi+1:n
      rand_ss1=rand_ss1+((idx(xi)==idx(xj))&&(T(xi)==T(xj)));
      rand_dd1=rand_dd1+((idx(xi)~=idx(xj))&&(T(xi)~=T(xj)));
   end
end
rand_index=200*(rand_ss1+rand_dd1)/(n*(n-1));
Acc=Acc1/n*100;
match=[1:k;match];

function [ Accuracy, Sensitivity] = K_Means_classi( X1,X2 )
% This Code was designed by Mohammed Diykh University of Southern
% Queensland in 2016
%Input: X1 and X2, are any pair of sleep stages such as (Sleep stage 1 vs stage 2 or awake vs stage 1)
to be classified
%Ouput:Accuracy the precentage os corrected classififcation and sensitivity
x=[X1;X2];
[P1 P2]=size(X1);
Y1(1:P1)=1;
[Z1 Z2]=size(X2)
Y1(P1+1:Z1+P1)=2
Y1=Y1';
     k=2;
    p=100;
    [n m]=size(Y1);
    opts = statset('Display','final');
    [idx,ctrs,sumd] =
kmeans(x,k,'Distance','city','Replicates',p,'Options',opts,'start','uniform','emptyaction','drop');
    [Accuracy, Sensitivity, Bestmatch]=AccMeasure(idx, Y1);
```

```
end


 function [ A1] =Adj_Cal(D)
% Receives:
% A Graph distance matrix D
%
% Returns:
% Adjacency Matrix A1, a threshold

[n m]=size(D);
A=zeros(n,n);
for i=1 : n
    e=D(i,:);
    ee=sum(e)/n;
for j=1:n
 if i ~= j
 if( D(i,j)<=ee)
     A(i,j)=1;
    else
      A(i,j)=0;
end
end
end
end
A1=A;
end
```

Appendix F

A Matlab simulation code to detect the DoA

In this appendix, a simulation code to assess the DoA using EEG signals is presented. In this simulation cod, some of functions used were from spectral graph wavelet transform toolbox5 and Matlab tool for network analysis6.

[5] http://wiki.epfl.ch/sgwt
[6] http://strategic.mit.edu/downloads.php?page=matlab_networks

```
 load Subject_3;% You can select any suject from Subject_1 to Subject_5
x=Denoised_x;
%%%%%%% load EEG signal%%%%%%%%%%%%%
sample_rate=128;  %%%% Sampling rate%%%%%%%%%%%%%
% give up no mean data
if length(BIS)*sample_rate<=length(x);
BIS=BIS(1:length(BIS));
x=x(1:sample_rate*length(BIS));
else
BIS=BIS(1:floor(length(x)/sample_rate));
x=x(1:sample_rate*floor(length(x)/sample_rate));
end

%%
unit_size=56;%60 means 60 second// window size
number_period=floor(length(BIS)/unit_size);
value_length=unit_size*number_period;
BIS=BIS(1:value_length);
x=x(1:value_length*sample_rate);

%%
xxx=[]; %%%%%% dividing the signal into overlapping window%%%%%%%
for i=1:length(BIS)-unit_size+1
xxx(i,:)=x((i-1)*sample_rate+1:(i-1)*sample_rate+unit_size*sample_rate);
end

n=length(xxx(1,:));
S=floor(n/65);
unit_size=56;
 for i=1:length(BIS)-unit_size+1
    z=xxx(1,:);
    z1=z(1:S*65);
    R2=reshape(z1, S,[])' ; %% sepeerate each window into m blocks
    for k=1:65
       T1=mainpoint_1(R2(k,:));% 10 statistical features are extracted from each sub-segments
       T2(k,:)=T1;
    end
    RRR=reshape(T2', [],1);
    WW=dist(RRR');
    WW1=Adj_mat11(WW); %%% calculate Adjecency matrix%%%%%
    L=full(sgwt_laplacian(WW1));  %%% calculate Laplacian matrix%%%%%
    Nscales=3;  %%% scal level 3
    lmax=sgwt_rough_lmax(L); %%%estimated upper bound on maximum eigenvalue of L
    arange=[0 lmax];
    [g,gp,t]=sgwt_filter_design(lmax,Nscales);% design filter
     m=50; % order of polynomial approximation
    for k=1:numel(g)
    c{k}=sgwt_cheby_coeff(g{k},m,m+1,arange);
    end
    %% compute transform of delta at one vertex;
     % select wavelet kernel
    t=3;
jcenter=32; % vertex to center wavelets to be shown
fprintf('Computing forward transform of delta at vertex %g\n',jcenter);
N=size(L,1);
d=sgwt_delta(N,jcenter);
% forward transform, using chebyshev approximation
t1=1;
t2=2;
a=2;
```

```
b=2;
wpall=sgwt_cheby_op(d,L,c,arange);
gb= @(x) sgwt_kernel(x,'a',a,'b',b,'t1',t1,'t2',t2);
g=@(x) gb(t*x);
wp_e=sgwt_ftsd(d,gb,t,L);%%%%%% r - output wavelet coefficients
GWT_Coff(i,:)=wp_e';
GWT_Cber_coff{1}.D=c;

 end
 [j1 j2]=size(GWT_Coff)
 for i=1:j1
    Tot_en(i)=sum(GWT_Coff(i,:)) %%% total engery of wavelet coefficients
 end
 y=BIS(1:j1);
 offset=Offset_cal(y,Tot_en);
 Tot_en=Tot_en';
 Newindex9=Tot_en+offset;
 figure(2);
plot(y,'b'), title('BIS & Newindex9');

hold on;
plot(Newindex9,'r');
hold on;
%plot(SQ,'k');
legend('BIS','Newindex9')
hold off;

function [X ] = feature( Y )
%UNTITLED7 Summary of this function goes here
%   Detailed explanation goes here
 X(1)=max(Y);
 X(2)=range(Y);
 X(3)=std(Y);
 X(4)=min(Y);
 X(5)=mean(Y);
 X(6)=mode(Y);
 X(7)=median(Y);
 [q1 q2 q3 m g d s]=quartile(Y);
 X(8)=q1;
 X(9)=q2;
 X(10)=var(Y);
end


% sgwt_demo2 : Allows exploring wavelet scale and approximation accuracy
%
% This demo builds the SGWT for the minnesota traffic graph, a graph
% representing the connectivity of the minnesota highway system. One center
% vertex is chosen, and then the exact (naive forward transform) and the
% approximate (via chebyshev polynomial approximation) wavelet transforms
% are computed for a particular value of the wavelet scale t. The relative
% error of the exact and approximate wavelets is computed. The user may
% then adjust the value of t, the degree m of the chebyshev polynomial
% approximation, and the center vertex in order to explore their effects.

function sgwt_demo2
close all
fprintf('Welcome to SGWT demo #2\n');

% touch variables to be shared among sub-functions
```

```
gb=[]; c=[];

% create UI elements
fh=figure('Visible','on','Name','demo 2 ui','Position',[425,920,400,150]);
uipanelh=uipanel('Parent',fh,'Title','','Units','pixels','BorderType','none');
tsliderh=uicontrol(uipanelh,'style','slider','max',50,'min',0,'value',1,...
                'sliderstep',[.005 .1],'position',[25,10,300,20],...
                'callback',{@tslider_callback});
msliderh=uicontrol(uipanelh,'style','slider','max',100,'min',1,'value',20,...
                'sliderstep',[.001 .1],'position',[25,60,300,20],...
                'callback',{@mslider_callback});
jbuttonh=uicontrol(uipanelh,'style','pushbutton','position',[50,110,150,20],...
                'string','Select center vertex','callback',{@jbutton_callback});
ttexth=uicontrol(uipanelh,'style','text','string','','position',[325,10,100,20]);
mtexth=uicontrol(uipanelh,'style','text','string','','position',[325,60,100,20]);
jtexth=uicontrol(uipanelh,'style','text','string','','position',[325,100,100,20]);
uicontrol(uipanelh,'style','text','string',...
        'Chebyshev polynomial order (m)','position',...
         [60,80,200,20]);
uicontrol(uipanelh,'style','text','string',...
        'Wavelet scale (t)','position',...
        [60,30,200,20]);


%% Load graph and compute Laplacian
fprintf('Loading minnesota traffic graph\n');
A=load('WWW');

fprintf('Computing graph laplacian\n')
[ki,kj]=find(A);
L=sgwt_laplacian(A);
fprintf('Measuring largest eigenvalue, lmax = ');
lmax=sgwt_rough_lmax(L);
fprintf('%g\n',lmax);
arange=[0 lmax];

msize=100;

% initial values
t=3; % wavelet scale

m=20; % chebyshev polynomial order, for approximation
jcenter=550;

fprintf('\n');
update_uitext;
update_graphfig
update_kernel
update_waveletfigs

function update_graphfig
  figure(2)
  set(gcf,'renderer','zbuffer');
  fprintf('Displaying traffic graph\n');
  set(gcf,'position',[0,600,400,400]);
  %clf('reset');
  hold on
  scatter(x,y,msize,[.5 .5 .5],'.');
  plot([x(ki)';x(kj)'],[y(ki)';y(kj)'],'k');
  set(gca,'Xtick',[]);
```

```
  set(gca,'Ytick',[]);
  axis equal
  axis off
  scatter(x(jcenter),y(jcenter),msize,'r.');
  drawnow
  end

  function update_kernel
  % select wavelet kernel
  t1=1;
  t2=2;
  a=2;
  b=2;
  tmin=t1/lmax;
% scales t<tmin will show same wavelet shape as t=tmin, as
% wavelet kernel g is monomial in interval [0,1)
set(tsliderh,'min',tmin);
  gb= @(x) sgwt_kernel(x,'a',a,'b',b,'t1',t1,'t2',t2);
  g=@(x) gb(t*x);
  % polynomial approximation
  for k=1:numel(g)
    c=sgwt_cheby_coeff(g,m,m+1,arange);
  end
  lambda=linspace(0,lmax,1e3);
  figure(3)
  set(gcf,'position',[425,580,600,250])
  plot(lambda,g(lambda),lambda,sgwt_cheby_eval(lambda,c,arange));
  legend('Exact Wavelet kernel','Chebyshev polynomial approximation');
end

function update_waveletfigs

  fprintf('\nReomputing wavelets with t=%g, m=%g\n',t,m);
  d=sgwt_delta(N,jcenter);
  fprintf('Computing wavelet by naive forward transform\n');
  figure(4)
  set(gcf,'position',[0,100,400,400])
  wp_e=sgwt_ftsd(d,gb,t,L);
  show_wavelet(wp_e,x,y);
  % show wavelet (naive)
  title('exact wavelet (naive forward transform)');
  fprintf('Computing wavelet by Chebyshev approximation\n');
  figure(5)
  set(gcf,'position',[425,100,400,400])
  % show wavelet (chebyshev)
  wp_c=sgwt_cheby_op(d,L,c,arange);
  show_wavelet(wp_c,x,y);
  title('approximate wavelet (transform via chebyshev approximation)');
  relerr=norm(wp_e-wp_c)/norm(wp_e);
  fprintf('Relative error between exact and approximate wavelet %g\n',relerr)
end

function show_wavelet(wp,x,y)
[Fs,s_ind]=sort(abs(wp),'descend');
scatter(x(s_ind),y(s_ind),msize,wp(s_ind),'.');
caxis([-1 1]*max(abs(wp)));
hcb=colorbar('location','north');
set(gca,'Xtick',[]);
set(gca,'Ytick',[]);
cxt=get(hcb,'Xtick');
```

```
cxt=[cxt(1),0,cxt(end)];
set(hcb,'Xtick',cxt);
cpos=get(hcb,'Position');
cpos(4)=.02; % make colorbar thinner
set(hcb,'Position',cpos);
axis equal
axis off
end

function update_uitext
set(ttexth,'string',sprintf('t=%0.3f',t));
set(mtexth,'string',sprintf('m=%g',m));
set(jtexth,'string',sprintf('j=%g',jcenter));
end

function tslider_callback(source,eventdata)
t=get(tsliderh,'value');
update_uitext;
update_kernel;
update_waveletfigs;
end

function mslider_callback(source,eventdata)
newm=get(msliderh,'value');
if newm<m
  m=floor(newm);
else
  m=ceil(newm);
end
set(msliderh,'value',m);
update_uitext;
update_kernel;
update_waveletfigs;
end

function jbutton_callback(source,eventdata)
figure(2)
fprintf('Select new center vertex\n');
[xp,yp]=ginput(1);
oldjcenter=jcenter;
jcenter=argmin((xp-x).^2+(yp-y).^2);
scatter(x(jcenter),y(jcenter),msize,'r.');
scatter(x(oldjcenter),y(oldjcenter),msize,[.5 .5 .5],'.');
drawnow
update_uitext
update_waveletfigs
end

end

function i = argmin(x)
  i=min(find(x==min(x)));
end

%% compute transform of delta at one vertex
jcenter=32; % vertex to center wavelets to be shown
fprintf('Computing forward transform of delta at vertex %g\n',jcenter);
N=size(L,1);
d=sgwt_delta(N,jcenter);
% forward transform, using chebyshev approximation
```

```
wpall=sgwt_cheby_op(d,L,c,arange);

fprintf('Displaying wavelets\n');
msize=100;
cp=[-1.4,-16.9,3.4]; % camera position
%% Visualize result

% show original point
ws=300;
figure;
xp=0; yp=ws+100;
set(gcf,'position',[xp,yp,ws-10,ws+10]);
scatter3(x(1,:),x(2,:),x(3,:),msize,d,'.');
set(gcf,'Colormap',[.5 .5 .5;1 0 0]);
clean_axes(cp);
title(sprintf('Vertex %g',jcenter));

% show wavelets
for n=1:Nscales+1
   wp=wpall{n};
   figure
   xp=mod(n,3)*(ws+10);
   yp=(1-floor((n)/3))*(ws+100);
   set(gcf,'position',[xp,yp,ws-10,ws+10]);
   scatter3(x(1,:),x(2,:),x(3,:),msize,wp,'.');
   colormap jet
   caxis([-1 1]*max(abs(wp)));
   clean_axes(cp);

   hcb=colorbar('location','north');
   cxt=get(hcb,'Xtick');
   cxt=[cxt(1),0,cxt(end)];
   set(hcb,'Xtick',cxt);
   cpos=get(hcb,'Position');
   cpos(4)=.02; % make colorbar thinner
   set(hcb,'Position',cpos);
   set(hcb,'Position',[.25 .91 .6 .02]);

   if n==1
     title('Scaling function');
   else
     title(sprintf('Wavelet at scale j=%g, t_j = %0.2f',n-1,t(end+1-(n-1))));

   end
end


function clean_axes(cp)
xlim([-1 1]);ylim([-1 1]);zlim([-1 1]);
set(gca,'Xtick',[-1 0 1]);
set(gca,'Ytick',[-1 0 1]);
set(gca,'Ztick',[-1 0 1]);
axis square
set(gca,'CameraPosition',cp);

% rescale_center
% center input data at origin, then rescale so that all coordinates
% are between -1 and 1
%
% x should be dxN
```

```
function r=rescale_center(x)
N=size(x,2);
d=size(x,1);
x=x-repmat(mean(x,2),[1,N]);
c=max(abs(x(:)));
r=x/c;
```

```
function data = create_synthetic_dataset(data)
% create_synthetic_dataset creates test data for running nldr algorithms.
%
% inputs:
%    data      a struct describing the test data
%              .dataset the number of the example, see code for more infos
%              .n       the number of data points (default=400)
%              .state   the initial state for the random numbers (default=0)
%              .noise   the variance of Gaussian noise to add (default=0)
%              other options for some of the data sets (see code)
%              alternatively, data = 1 chooses the dataset directly,
%              the number of points defaults to 1000
%
% outputs:
%    data      a struct containing .x the generated data, each column is
%              a data point, and other stuff:
%              .z     the "correct" embedding
%              .e     some random noise of same dimensionality
%              .x_noisefree  the noisefree version of .x, i.e.
%                    .x = .xnoise_free + sqrt(.noise) * .e
%
% Adapted from create.m, originally written by
% (c) Stefan Harmeling, 2006
% using the examples of the original LLE and ISOMAP code.

if ~isfield(data, 'dataset'),
  number = data;
  clear data
  data.dataset = number;
end
if ~isfield(data, 'n'), data.n = 400; end
if ~isfield(data, 'noise'), data.noise = 0.0; end
if ~isfield(data, 'state'), data.state = 0; end

% set the randomness
rand('state', data.state);
randn('state', data.state);

data.typ = 'data';
switch data.dataset
 case 0 % "swiss roll with hole"
  data.name = 'swiss roll with hole';
  n = data.n;
  a = 1;   % swiss roll goes from a*pi to b*pi
  b = 4;
  y = rand(2,n);
```

```
 % punch a rectangular hole at the center
 l1 = 0.05; l2 = 0.15;
 y = y - 0.5;
 ok = find((abs(y(1,:))>l1) | (abs(y(2,:))>l2));
 i = length(ok);
 y(:, 1:i) = y(:, ok);
 while (i<n)
   p = rand(2,1) - 0.5;
   if (abs(p(1))>l1) | (abs(p(2))>l2)
     i = i + 1;
     y(:,i) = p;
   end
 end
 y = y + 0.5;
 tt = (b-a)*y(1,:) + a;
 tt = pi*tt;
 height = 21*y(2,:);
 data.col = tt;
 data.x = [tt.*cos(tt); height; tt.*sin(tt)];
 data.z = [tt; height]; % the ground truth
 data.az = -4;
 data.el = 13;

case -1 % "swiss roll" dataset extracted from LLE's swissroll.m
 data.name = 'uniform swiss roll';
 n = data.n;
 a = 1;   % swiss roll goes from a*pi to b*pi
 b = 4;
 y = rand(2,n);
 data.z = y;  % the ground truth
 switch 1
  case 1
   % uniform distribution along the manifold (in data space)
   tt = sqrt((b*b-a*a)*y(1,:)+a*a);
  case 2
%    error('do not use this case')
   % nonuniform distribution along the manifold (in data space)
   tt = (b-a)*y(1,:) + a;
 end
 tt = pi*tt;
 % now tt should go from a*pi to b*pi
 height = 21*y(2,:);
 data.col = tt;
 data.x = [tt.*cos(tt); height; tt.*sin(tt)];
 data.az = -4;
 data.el = 13;

case 1 % "swiss roll (uniform in embedding space)"
 % dataset extracted from LLE's swissroll.m
 data.name = 'classic swiss roll';
 n = data.n;
 a = 1;   % swiss roll goes from a*pi to b*pi
 b = 4;
 y = rand(2,n);
 tt = (b-a)*y(1,:) + a;
 tt = pi*tt;
 height = 21*y(2,:);
 data.col = tt;
 data.x = [tt.*cos(tt); height; tt.*sin(tt)];
 data.z = [tt; height]; % the ground truth
```

```
 data.az = -4;
 data.el = 13;

case 11 % "undersampled swiss roll"
 % dataset extracted from LLE's swissroll.m
 data.name = 'undersampled swiss roll';
 data.n = 100;
 n = data.n;
 a = 1;   % swiss roll goes from a*pi to b*pi
 b = 4;
 y = rand(2,n);
 tt = (b-a)*y(1,:) + a;
 tt = pi*tt;
 height = 21*y(2,:);
 data.col = tt;
 data.x = [tt.*cos(tt); height; tt.*sin(tt)];
 data.z = [tt; height]; % the ground truth
 data.az = -4;
 data.el = 13;

case 12 % "swiss roll"
 % dataset extracted from LLE's swissroll.m
 data.name = 'classic swiss roll';
 data.n = 400;
 n = data.n;
 a = 1;   % swiss roll goes from a*pi to b*pi
 b = 4;
 y = rand(2,n);
 tt = (b-a)*y(1,:) + a;
 tt = pi*tt;
 height = 21*y(2,:);
 data.col = tt;
 data.x = [tt.*cos(tt); height; tt.*sin(tt)];
 data.z = [tt; height]; % the ground truth
 data.az = -4;
 data.el = 13;

case 2 % "scurve" dataset extracted from LLE's scurve.m
 data.name = 'scurve';
 n = data.n;
 % I added 'ceil' and 'floor' to account for the case that n is odd
 angle = pi*(1.5*rand(1,ceil(n/2))-1); height = 5*rand(1,n);
 data.x = [[cos(angle), -cos(angle(1:floor(n/2)))]; height;[ sin(angle), 2-sin(angle)]];
 data.col = [angle, 1.5*pi + angle];
 data.z = [angle, 1.5*pi+angle; height]; % the ground truth

case 3 % "square" dataset, a uniformly sampled 2D square randomly
       % rotated into higher dimensions
 data.name = 'square';
 n = data.n;
 d = 2;     % intrinsic dimension
 % optional parameter for dataset==3
 % data.D      dimension of the data
 if ~isfield(data, 'D'), data.D = 3; end
 % generate random rotation matrix
 D = data.D;
 A = randn(D, D);
 options.disp = 0;
 [R, dummy] = eigs(A*A', d, 'LM', options);
 tt = rand(d, n);
```

```
 data.col = tt(1,:);
 data.x = R*tt;
 data.z = tt;   % the ground truth
 data.az = 7;
 data.el = 40;

case 4 % spiral: two dimensional "swiss roll"
 data.name = 'spiral';
 n = data.n;
 tt = (3*pi/2)*(1+2*rand(1, n));
 data.col = tt;
 data.x = [tt.*cos(tt); tt.*sin(tt)];
 data.z = tt; % the ground truth

case -4 % spiral: two dimensional "swiss roll"
 data.name = 'noisy spiral';
 n = data.n;
 tt = (3*pi/2)*(1+2*rand(1, n));
 data.col = tt;
 data.x = [tt.*cos(tt); tt.*sin(tt)];
 data.x = data.x + randn(size(data.x));
 data.z = tt; % the ground truth

case 5 % hole: a dataset with a hole
 data.name = 'hole';
 n = data.n;
 data.x = rand(2,n) - 0.5;
 % punch a rectangular hole at the center
 l1 = 0.2; l2 = 0.2;
 ok = find((abs(data.x(1,:))>l1) | (abs(data.x(2,:))>l2));
 i = length(ok);
 data.x(:, 1:i) = data.x(:, ok);
 while (i<n)
   p = rand(2,1) - 0.5;
   if (abs(p(1))>l1) | (abs(p(2))>l2)
     i = i + 1;
     data.x(:,i) = p;
   end
 end
 data.col = data.x(2,:);
 data.z = data.x;

case 6 % P : taken from Saul's slides
 % note that for k=20, isomap and lle work fine which is very different
 % from the plots that Saul showed in his slides.
 data.name = 'P';
 load x
 x(2,:) = 500-x(2,:);
 data.x = x;
 data.z = x;
 data.col = data.z(2,:);
 data.n = size(x, 2);

case 7 % fishbowl: uniform in data space
 gamma = 0.8;
 data.name = 'fishbowl (uniform in data space)';
 n = data.n;
 data.x = rand(3,n)-0.5;
 %project all data onto the surface of the unit sphere
 data.x = data.x ./ repmat(sqrt(sum(data.x.*data.x, 1)), [3 1]);
```

```
 ok = find(data.x(3,:) < gamma);
 i = length(ok);
 data.x(:, 1:i) = data.x(:, ok);
 while (i < n)
   p = rand(3,1)-0.5;
   p = p / sqrt(p'*p);
   if (p(3) < gamma)
     i = i+1;
     data.x(:, i) = p;
   end
 end
 % the projection on the plane works as follows:
 % start a beam from (0,0,1) through each surface point on the sphere
 % and look where it hits the xy plane.
 data.z = data.x(1:2,:) ./ repmat(1-data.x(3,:), [2 1]);
 data.col = data.x(3,:);
 data.az = -18;
 data.el = 16;
case 8 % fishbowl: uniform in embedding space
 data.name = 'fishbowl (uniform in embedding space)';
 n = data.n;
 data.z = rand(2, n) - 0.5;
 % keep the disc
 ok = find(sum(data.z .* data.z) <= 0.25);
 i = length(ok);
 data.z(:, 1:i) = data.z(:, ok);
 while (i < n)
   p = rand(2,1) - 0.5;
   if (p'*p <= 0.25)
     i = i + 1;
     data.z(:, i) = p;
   end
 end
 gamma = 0.8;  % same role/parameter as in case 7
 data.z = 2*sqrt((1+gamma)/(1-gamma))*data.z;
 % project the disc onto the sphere
 alpha = 2 ./ (1 + sum(data.z .* data.z, 1));
 data.x = [repmat(alpha, [2 1]).*data.z; zeros(1, n)];
 data.x(3,:) = 1-alpha;
 data.col = data.x(3,:);
 data.az = -18;
 data.el = 16;

case 9  % a gaussian blob
 data.name = 'gaussian blob';
 n = data.n;
 data.x = randn(3,n);
 data.z = data.x(2:3,:);
 data.col = data.x(3,:);

end


data.D = size(data.x, 1);  % dimensionality of the data
% finally generate noise
data.e = randn(size(data.x));
data.x_noisefree = data.x;  % the noise free data
data.x = data.x_noisefree + sqrt(data.noise)*data.e;

% precalculate the distanzmatrix
```

```
data.distances = distanz(data.x);



function d = distanz(x,y,type)
% DISTANZ : calculates the distances between all vectors in x and y.
%
% usage:
%    d = distanz(x,y);
%
% inputs:
%    x      matrix with col vectors
%    y      matrix with col vectors (default == x)
%    type   the type of algorithm that is used (default==3)
%
% outputs:
%    d      distance matrix, not squared
%
% note:
%    part of the code is inspired by dist.m of the nntoolbox, other
%    part adapted from Francis Bach who took it from Roland
%    Bunschoten.
%
% sth * 19apr2002
% Adapted from create.m, originally written by
% (c) Stefan Harmeling, 2006

if exist('type')~=1|isempty(type), type = 3; end
switch type
 case 1  % inspired by dist.m
  if exist('y')~=1|isempty(y)
    % here comes code just for x
    [rx,cx] = size(x);
    d = zeros(cx,cx);
    nuller = zeros(cx,1);
    for c = 1:cx
      d(c,:) = sum((x-x(:,c+nuller)).^2,1);
    end
  else
    % here comes code for x and y
    [rx,cx] = size(x);
    [ry,cy] = size(y);
    if rx~=ry, error('x and y do not fit'), end
    d = zeros(cx,cy);
    if cx>cy
      nuller = zeros(cx,1);
      for c = 1:cy
            d(:,c) = sum((x-y(:,c+nuller)).^2,1)';
      end
    else
      nuller = zeros(cy,1);
      for c = 1:cx
            d(c,:) = sum((x(:,c+nuller)-y).^2,1);
      end
    end
  end

 case 2  % same as case 1, but with repmat instead of nuller
  if exist('y')~=1|isempty(y)
    % here comes code just for x
```

```
   [rx,cx] = size(x);
   d = zeros(cx,cx);
   nuller = zeros(cx,1);
   for c = 1:cx
     d(c,:) = sum((x-repmat(x(:,c),[1 cx])).^2,1);
   end
  else
   % here comes code for x and y
   [rx,cx] = size(x);
   [ry,cy] = size(y);
   if rx~=ry, error('x and y do not fit'), end
   d = zeros(cx,cy);
   if cx>cy
     nuller = zeros(cx,1);
     for c = 1:cy
           d(:,c) = sum((x-repmat(y(:,c),[1 cx])).^2,1)';
     end
   else
     nuller = zeros(cy,1);
     for c = 1:cx
           d(c,:) = sum((repmat(x(:,c),[1 cy])-y).^2,1);
     end
   end
  end

 case 3  % inspired by Roland Bunschoten
  if exist('y')~=1|isempty(y)
   % here comes code just for x
   cx = size(x,2);
   xx = sum(x.*x,1); xz = x'*x;
   d = abs(repmat(xx',[1 cx]) - 2*xz + repmat(xx,[cx 1]));
  else
   % here comes code for x and y
   [rx,cx] = size(x);
   [ry,cy] = size(y);
   if rx~=ry, error('x and y do not fit'), end
   xx = sum(x.*x,1); yy = sum(y.*y,1);  xy = x'*y;
   d = abs(repmat(xx',[1 cy]) + repmat(yy,[cx 1]) - 2*xy);
  end
end

d = sqrt(d);


% sgwt_cheby_square : Chebyshev coefficients for square of polynomial
%
% function d=sgwt_cheby_square(c)
%
% Inputs :
% c - Chebyshev coefficients for p(x) = sum c(1+k) T_k(x) ; 0<=K<=M
%
% Outputs :
% d - Chebyshev coefficients for p(x)^2 = sum d(1+k) T_k(x) ;
%     0<=k<=2*M

function d=sgwt_cheby_square(c)
M=numel(c)-1;
cp=c;
cp(1)=.5*c(1);
```

```
% adjust cp so that
% p(x) = sum cp(1+k) T_k(x)
% for all k>=0 (rather than with special case for k=0)
%
% Then formula for dp in terms of cp is simpler.
% Ref: my notes, july 20, 2009
dp=zeros(1,2*M+1);
% keep in mind : due to indexing from 1
% c(1+k) is k'th Chebyshev coefficient

for m=0:(2*M)
   if (m==0)
      dp(1+m)=dp(1+m)+.5*cp(1)^2;
      for i=0:M
         dp(1+m)=dp(1+m)+.5*cp(1+i)^2;
      end
   elseif (m<=M)
      for i=0:m
         dp(1+m)=dp(1+m)+.5*cp(1+i)*cp(1+m-i);
      end
      for i=0:(M-m)
         dp(1+m)=dp(1+m)+.5*cp(1+i)*cp(1+i+m);
      end
      for i=m:M
         dp(1+m)=dp(1+m)+.5*cp(1+i)*cp(1+i-m);
      end
   else % M<m<=2*M
      for i=(m-M):M
         dp(1+m)=dp(1+m)+.5*cp(1+i)*cp(1+m-i);
      end
   end
end
d=dp;
d(1)=2*dp(1);



% sgwt_check_connected : Check connectedness of graph
%
% function r=sgwt_check_connected(A)
%
% returns 1 if graph is connected, 0 otherwise
% Uses boost graph library breadth first search
%
% Inputs :
% A - adjacency matrix
%
% Outputs :
% r - result
%

function r=sgwt_check_connected(A)
  d=bfs(A,1);
  r=~any(d==-1);


% sgwt_delta : Return vector with one nonzero entry equal to 1.
%
% function r=sgwt_delta(N,j)
%
```

```
% Returns length N vector with r(j)=1, all others zero
%
% Inputs :
% N - length of vector
% j - position of "delta" impulse
%
% Outputs:
% r - returned vector

function r=sgwt_delta(N,j)
  r=zeros(N,1);
  r(j)=1;

% sgwt_view_design : display filter design in spectral domain
%
% function sgwt_view_design(g,t,arange)
%
% This function graphs the input scaling function and wavelet
% kernels, indicates the wavelet scales by legend, and also shows
% the sum of squares G and corresponding frame bounds for the transform.
%
% Inputs :
% g - cell array of function handles for scaling function and
%   wavelet kernels
% t - array of wavelet scales corresponding to wavelet kernels in g
% arange - approximation range

function sgwt_view_design(g,t,arange)
x=linspace(arange(1),arange(2),1e3);
clf
hold on

J=numel(g)-1;
co=get(gca,'ColorOrder');
G=0*x;
for n=0:J
    plot(x,g{1+n}(x),'Color',co(1+n,:));
    G=G+g{1+n}(x).^2;
end
plot(x,G,'k');
[A,B]=sgwt_framebounds(g,arange(1),arange(2));

hline(A,'m:');
hline(B,'g:');
leglabels{1}='h';
for j=1:J
    leglabels{1+j}=sprintf('t=%.2f',t(j));
end
leglabels{J+2}='G';
leglabels{J+3}='A';
leglabels{J+4}='B';

%set(gca,'Ytick',0:3);

legend(leglabels)
title(['Scaling function kernel h(x), Wavelet kernels g(t_j x), Sum ' ...
       'of Squares G, and Frame Bounds']);

function hline(y,varargin)
  xl=xlim;
```

```
  plot(xl,y*[1 1],varargin{:});

function s=sgwt_setscales(lmin,lmax,Nscales)
  t1=1;
  t2=2;

  smin=t1/lmax;
  smax=t2/lmin;
  s=exp(linspace(log(smin),log(smax),Nscales));

% sgwt_rough_lmax : Rough upper bound on maximum eigenvalue of L
%
% function lmax=sgwt_rough_lmax(L)
%
% Runs Arnoldi algorithm with a large tolerance, then increases
% calculated maximum eigenvalue by 1 percent. For much of the SGWT
% machinery, we need to approximate the wavelet kernels on an
% interval that contains the spectrum of L. The only cost of using
% a larger interval is that the polynomial approximation over the
% larger interval may be a slightly worse approxmation on the
% actual spectrum. As this is a very mild effect, it is not likely
% necessary to obtain very tight bonds on the spectrum of L
%
% Inputs :
% L - input graph Laplacian
%
% Outputs :
% lmax - estimated upper bound on maximum eigenvalue of L

% This file is part of the SGWT toolbox (Spectral Graph Wavelet Transform toolbox)
%
% The SGWT toolbox is free software: you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation, either version 3 of the License, or
% (at your option) any later version.
%

function lmax=sgwt_rough_lmax(L)
opts=struct('tol',5e-3,'p',10,'disp',0);
lmax=eigs(L,1,'lm',opts);
lmax=lmax*1.01; % just increase by 1 percent to be robust to error
% sgwt_kernel_abspline3 : Monic polynomial / cubic spline / power law decay kernel
%
% function r = sgwt_kernel_abspline3(x,alpha,beta,t1,t2)
%
% defines function g(x) with g(x) = c1*x^alpha for 0<x<x1
% g(x) = c3/x^beta for x>t2
% cubic spline for t1<x<t2,
% Satisfying g(t1)=g(t2)=1
%
% Inputs :
% x : array of independent variable values
% alpha : exponent for region near origin
% beta : exponent decay
% t1, t2 : determine transition region
%
% Outputs :
% r - result (same size as x)

% This file is part of the SGWT toolbox (Spectral Graph Wavelet Transform toolbox)
```

```
%
% The SGWT toolbox is free software: you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation, either version 3 of the License, or
% (at your option) any later version.
%

function r = sgwt_kernel_abspline3(x,alpha,beta,t1,t2)
  r=zeros(size(x));
  % compute spline coefficients
  % M a = v
  M=[[1 t1 t1^2 t1^3];...
     [1 t2 t2^2 t2^3];...
     [0 1 2*t1 3*t1^2];...
     [0 1 2*t2 3*t2^2]];
  %v=[t1^alpha ; t2^(-beta) ; alpha*t1^(alpha-1) ; -beta*t2^(-beta-1)];
  v=[1 ; 1 ; t1^(-alpha)*alpha*t1^(alpha-1) ; -beta*t2^(-beta-1)*t2^beta];
  a=M\v;

  r1=find(x>=0 & x<t1);
  r2=find(x>=t1 & x<t2);
  r3=find(x>=t2);
  r(r1)=x(r1).^alpha*t1^(-alpha);
  r(r3)=x(r3).^(-beta)*t2^(beta);

  x2=x(r2);
  r(r2)=a(1)+a(2)*x2+a(3)*x2.^2+a(4)*x2.^3;
%  tmp=polyval(flipud(a),x2);
%  keyboard


% sgwt_kernel_abspline5 : Monic polynomial / quintic spline / power law decay kernel
%
% function r = sgwt_kernel_abspline5(x,alpha,beta,t1,t2)
%
% Defines function g(x) with g(x) = c1*x^alpha for 0<x<x1
% g(x) = c3/x^beta for x>t2
% quintic spline for t1<x<t2,
% Satisfying g(t1)=g(t2)=1
% g'(t1)=g'(t2)
% g"(t1)=g"(t2)
%
% Inputs :
% x : array of independent variable values
% alpha : exponent for region near origin
% beta : exponent decay
% t1, t2 : determine transition region
%
% Outputs :
% r - result (same size as x)

% This file is part of the SGWT toolbox (Spectral Graph Wavelet Transform toolbox)
%
% The SGWT toolbox is free software: you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation, either version 3 of the License, or
% (at your option) any later version.
%
```

```
function r = sgwt_kernel_abspline5(x,alpha,beta,t1,t2)
  r=zeros(size(x));
  % compute spline coefficients
  % M a = v
  M=[[1 t1 t1^2 t1^3 t1^4 t1^5];...
     [1 t2 t2^2 t2^3 t2^4 t2^5];...
     [0 1 2*t1 3*t1^2 4*t1^3 5*t1^4];...
     [0 1 2*t2 3*t2^2 4*t2^3 5*t2^4];
     [0 0 2 6*t1 12*t1^2 20*t1^3];...
     [0 0 2 6*t2 12*t2^2 20*t2^3]...
     ];
  %v=[t1^alpha ; t2^(-beta) ; alpha*t1^(alpha-1) ; -beta*t2^(-beta-1)];
  v=[1 ; 1 ; ...
     t1^(-alpha)*alpha*t1^(alpha-1) ; -beta*t2^(-beta-1)*t2^beta; ...
     t1^(-alpha)*alpha*(alpha-1)*t1^(alpha-2);-beta*(-beta-1)*t2^(-beta-2)*t2^beta
     ];
  a=M\v;

  r1=find(x>=0 & x<t1);
  r2=find(x>=t1 & x<t2);
  r3=find(x>=t2);
  r(r1)=x(r1).^alpha*t1^(-alpha);
  r(r3)=x(r3).^(-beta)*t2^(beta);

  x2=x(r2);
  r(r2)=a(1)+a(2)*x2+a(3)*x2.^2+a(4)*x2.^3+a(5)*x2.^4+a(6)*x2.^5;
%  tmp=polyval(flipud(a),x2);
%  keyboard

% sgwt_laplacian :  Compute graph laplacian from connectivity matrix
%
% function L = sgwt_laplacian(A,varargin)
%
% Connectivity matrix A must be symmetric.  A may have arbitrary
% non-negative values, in which case the graph is a weighted
% graph. The weighted graph laplacian follows the definition in
% "Spectral Graph Theory" by Fan R. K. Chung
%
% Inputs :
% A - adjacency matrix
% Selectable Input parameters :
% 'opt' - may be 'raw' or 'normalized' (default raw) to select
%         un-normalized or normalized laplacian
%
% Outputs :
% L - graph Laplacian

% This file is part of the SGWT toolbox (Spectral Graph Wavelet Transform toolbox)
%
% The SGWT toolbox is free software: you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation, either version 3 of the License, or
% (at your option) any later version.
%

function L = sgwt_laplacian(A,varargin)
  control_params={'opt','raw'}; % or normalized
  argselectAssign(control_params);
  argselectCheck(control_params,varargin);
  argselectAssign(varargin);
```

```
N=size(A,1);
if N~=size(A,2)
  error('A must be square');
end
degrees=vec(full(sum(A)));
% to deal with loops, must extract diagonal part of A
diagw=diag(A);

% w will consist of non-diagonal entries only
[ni2,nj2,w2]=find(A);
ndind=find(ni2~=nj2); % as assured here
ni=ni2(ndind);
nj=nj2(ndind);
w=w2(ndind);

di=vec(1:N); % diagonal indices

switch opt
 case 'raw'
  % non-normalized laplacian L=D-A
  L=sparse([ni;di],[nj;di],[-w;degrees-diagw],N,N);
 case 'normalized'
  % normalized laplacian D^(-1/2)*(D-A)*D^(-1/2)
  % diagonal entries
  dL=(1-diagw./degrees); % will produce NaN for degrees==0 locations
  dL(degrees==0)=0;% which will be fixed here
  % nondiagonal entries
  ndL=-w./vec( sqrt(degrees(ni).*degrees(nj)) );
  L=sparse([ni;di],[nj;di],[ndL;dL],N,N);
 otherwise
  error('unknown option');
end


function A=sgwt_meshmat(dim,varargin)
control_params={'boundary','rectangle'};
argselectAssign(control_params);
argselectCheck(control_params,varargin);
argselectAssign(varargin);
if (numel(dim)==1)
  dim=[1 1]*dim;
end
% build adjacency matrix : find i,j coordinates of center points
% and right and bottom neighbors, then build connectivity matrix.
% For each valid center,neighbor pair, will add A(center,neighbor)=1
% and A(neighbor,center)=1, so A will be symmetric
N=prod(dim);
[alli,allj]=find(ones(dim));
% (ci(k),cj(k)) has neighbor (ni(k),nj(k))
ci=[alli;alli];
cj=[allj;allj];
ni=[alli  ; alli+1];
nj=[allj+1; allj];
switch boundary
 case 'rectangle'
  % prune edges at boundary
  valid=(ni>=1 & ni<=dim(1) & nj>=1 & nj<=dim(2));
  ni=ni(valid);
  nj=nj(valid);
```

```
    ci=ci(valid);
    cj=cj(valid);
    cind=dim(1)*(cj-1)+ci;
    nind=dim(1)*(nj-1)+ni;
   case 'torus'
    % wrap indices to make torus
    ni=mod(ni,dim(1))+1;
    nj=mod(nj,dim(2))+1;
    cind=dim(1)*(cj-1)+ci;
    nind=dim(1)*(nj-1)+ni;
   otherwise
    error('unknown boundary option');
  end
  % assemble connection matrix
  A=sparse([cind,nind],[nind,cind],ones(1,2*numel(ni)),N,N);

% sgwt_framebounds : Compute approximate frame bounds for given sgw transform
%
% function [A,B,sg2,x]=sgwt_framebounds(g,lmin,lmax)
%
% Inputs :
% g - function handles computing sgwt scaling function and wavelet
% kernels
% lmin,lmax - minimum nonzero, maximum eigenvalue
%
% Outputs :
% A , B - frame bounds
% sg2 - array containing sum of g(s_i*x)^2 (for visualization)
% x - x values corresponding to sg2

% This file is part of the SGWT toolbox (Spectral Graph Wavelet Transform toolbox)
%
% The SGWT toolbox is free software: you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation, either version 3 of the License, or
% (at your option) any later version.
%

function [A,B,sg2,x]=sgwt_framebounds(g,lmin,lmax)
  N=1e4; % number of points for line search
  x=linspace(lmin,lmax,N);
  Nscales=numel(g);

  sg2=zeros(size(x));
  for ks=1:Nscales
    sg2=sg2+(g{ks}(x)).^2;
  end
  A=min(sg2);
  B=max(sg2);


% sgwt_ftsd : Compute forward transform in spectral domain
%
% function r=sgwt_ftsd(f,g,t,L)
%
% Compute forward transform by explicitly computing eigenvectors and
% eigenvalues of graph laplacian
%
% Uses persistent variables to store eigenvectors, so decomposition
% will be computed only on first call
```

```
%
% Inputs:
% f - input data
% g - sgw kernel
% t - desired wavelet scale
% L - graph laplacian
%
% Outputs:
% r - output wavelet coefficients

% This file is part of the SGWT toolbox (Spectral Graph Wavelet Transform toolbox)
% The SGWT toolbox is free software: you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation, either version 3 of the License, or
% (at your option) any later version.
%

function r=sgwt_ftsd(f,g,t,L)
persistent V D Lold
if (isempty(V) || any(vec(L~=Lold)))
  fprintf('Diagonalizing %g x %g L (could take some time ...)\n',size(L,1),size(L,2));
  [V,D]=eig(full(L));
  Lold=L;
end
lambda=diag(D);
fhat=V'*f;
r=V*(fhat.*g(t*lambda));



% sgwt_inverse : Compute inverse sgw transform, via conjugate gradients
%
% function r=sgwt_inverse(y,L,c,arange)
%
% Inputs:
% y - sgwt coefficients
% L - laplacian
% c - cell array of Chebyshev coefficients defining transform
% arange - spectral approximation range
%
% Selectable Control Parameters
% tol - tolerance for conjugate gradients (default 1e-6)
%


%
% The SGWT toolbox is free software: you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation, either version 3 of the License, or
% (at your option) any later version.
%

function r=sgwt_inverse(y,L,c,arange,varargin)
control_params={'tol',1e-6};
argselectAssign(control_params);
argselectCheck(control_params,varargin);
argselectAssign(varargin);

assert(iscell(c));
N=size(L,1);
```

```
% first compute adj = W^*y ( sort of slowly )
adj=zeros(N,1);
fprintf('computing adjoint\n');
for j=1:numel(c)
   tmp=sgwt_cheby_op(y{j},L,c{j},arange);
   adj=adj+tmp;
end

% W^* W
% compute P(x) = p(x)^2
fprintf('computing cheby coeff for P=p^2\n');
for j=1:numel(c)
   M(j)=numel(c{j});
end
maxM=max(M);
% dkh : code below could remove unnecessary use of cell arrays.
d{1}=zeros(1,1+2*(maxM-1));
for j=1:numel(c)
   cpad{j}=zeros(maxM,1);
   cpad{j}(1:M(j))=c{j};
   d{1}=d{1}+sgwt_cheby_square(cpad{j});
end
wstarw = @(x) sgwt_cheby_op(x,L,d{1},arange);
%% conjugate gradients
fprintf('computing inverse by conjugate gradients\n');
r=cgs(wstarw,adj,tol);

% sgwt_filter_design : Return list of scaled wavelet kernels and derivatives
% g{1} is scaling function kernel,
% g{2} ... g{Nscales+1} are wavelet kernels
%
% function [g,gp]=sgwt_filter_design(lmax,Nscales,varargin)
%
% Inputs :
% lmax - upper bound on spectrum
% Nscales - number of wavelet scales
%
% selectable parameters :
% designtype
% lpfactor - default 20. lmin=lmax/lpfactor will be used to determine
%           scales, then scaling function kernel will be created to
%           fill the lowpass gap.

% The SGWT toolbox is free software: you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation, either version 3 of the License, or
% (at your option) any later version.
%

function [g,gp,t]=sgwt_filter_design(lmax,Nscales,varargin)
control_params={'designtype','default','lpfactor',20,...
   'a',2,...
   'b',2,...
   't1',1,...
   't2',2,...
   };
argselectAssign(control_params);
argselectCheck(control_params,varargin);
argselectAssign(varargin);
```

```
switch designtype
   case 'default'
      lmin=lmax/lpfactor;
      t=sgwt_setscales(lmin,lmax,Nscales);
      gl = @(x) exp(-x.^4);
      glp = @(x) -4*x.^3 .*exp(-x.^4);
      gb= @(x) sgwt_kernel(x,'a',a,'b',b,'t1',t1,'t2',t2);
      gbp = @(x) sgwt_kernel_derivative(x,'a',a,'b',b,'t1',t1,'t2',t2);
      for j=1:Nscales
         g{j+1}=@(x) gb(t(end+1-j)*x);
         gp{j+1}=@(x) gbp(t(end+1-j)*x)*t(end+1-j); % derivative
      end
      % find maximum of g's ...
      % I could get this analytically as it is a cubic spline, but
      % this also works.
      f=@(x) -gb(x);
      xstar=fminbnd(f,1,2);
      gamma_l=-f(xstar);
      lminfac=.6*lmin;
      g{1}=@(x) gamma_l*gl(x/lminfac);
      gp{1} = @(x) gamma_l*glp(x/lminfac)/lminfac; % derivative
 case 'mh'
  lmin=lmax/lpfactor;
  t=sgwt_setscales(lmin,lmax,Nscales);
  gb=@(x) sgwt_kernel(x,'gtype','mh');
  gl = @(x) exp(-x.^4);
  for j=1:Nscales
    g{j+1}=@(x) gb(t(end+1-j)*x);
  end
  lminfac=.4*lmin;
  g{1}=@(x) 1.2*exp(-1)*gl(x/lminfac);

 otherwise

      keyboard
      error('Unknown design type');
end

function sgwt_demo2
close all
fprintf('Welcome to SGWT demo #2\n');

% touch variables to be shared among sub-functions
gb=[]; c=[];

% create UI elements
fh=figure('Visible','on','Name','demo 2 ui','Position',[425,920,400,150]);
uipanelh=uipanel('Parent',fh,'Title','','Units','pixels','BorderType','none');
tsliderh=uicontrol(uipanelh,'style','slider','max',50,'min',0,'value',1,...
                'sliderstep',[.005 .1],'position',[25,10,300,20],...
                'callback',{@tslider_callback});
msliderh=uicontrol(uipanelh,'style','slider','max',100,'min',1,'value',20,...
                'sliderstep',[.001 .1],'position',[25,60,300,20],...
                'callback',{@mslider_callback});
jbuttonh=uicontrol(uipanelh,'style','pushbutton','position',[50,110,150,20],...
                'string','Select center vertex','callback',{@jbutton_callback});
ttexth=uicontrol(uipanelh,'style','text','string','','position',[325,10,100,20]);
mtexth=uicontrol(uipanelh,'style','text','string','','position',[325,60,100,20]);
jtexth=uicontrol(uipanelh,'style','text','string','','position',[325,100,100,20]);
uicontrol(uipanelh,'style','text','string',...
```

```
        'Chebyshev polynomial order (m)','position',...
         [60,80,200,20]);
uicontrol(uipanelh,'style','text','string',...
        'Wavelet scale (t)','position',...
        [60,30,200,20]);


%% Load graph and compute Laplacian
fprintf('Loading minnesota traffic graph\n');
Q=load('minnesota.mat');
xy=Q.xy;
A=Q.A;
N=size(A,1);
x=xy(:,1);
y=xy(:,2);

fprintf('Computing graph laplacian\n')
[ki,kj]=find(A);
L=sgwt_laplacian(A);
fprintf('Measuring largest eigenvalue, lmax = ');
lmax=sgwt_rough_lmax(L);
fprintf('%g\n',lmax);
arange=[0 lmax];

msize=100;

% initial values
t=3; % wavelet scale

m=20; % chebyshev polynomial order, for approximation
jcenter=550;

fprintf('\n');
update_uitext;
update_graphfig
update_kernel
update_waveletfigs

function update_graphfig
  figure(2)
  set(gcf,'renderer','zbuffer');
  fprintf('Displaying traffic graph\n');
  set(gcf,'position',[0,600,400,400]);
  %clf('reset');
  hold on
  scatter(x,y,msize,[.5 .5 .5],'.');
  plot([x(ki)';x(kj)'],[y(ki)';y(kj)'],'k');
  set(gca,'Xtick',[]);
  set(gca,'Ytick',[]);
  axis equal
  axis off
  scatter(x(jcenter),y(jcenter),msize,'r.');
  drawnow
  end

  function update_kernel
  % select wavelet kernel
  t1=1;
  t2=2;
  a=2;
```

```
  b=2;
  tmin=t1/lmax;
% scales t<tmin will show same wavelet shape as t=tmin, as
% wavelet kernel g is monomial in interval [0,1)
set(tsliderh,'min',tmin);
  gb= @(x) sgwt_kernel(x,'a',a,'b',b,'t1',t1,'t2',t2);
  g=@(x) gb(t*x);
  % polynomial approximation
  for k=1:numel(g)
    c=sgwt_cheby_coeff(g,m,m+1,arange);
  end
  lambda=linspace(0,lmax,1e3);
  figure(3)
  set(gcf,'position',[425,580,600,250])
  plot(lambda,g(lambda),lambda,sgwt_cheby_eval(lambda,c,arange));
  legend('Exact Wavelet kernel','Chebyshev polynomial approximation');
end

function update_waveletfigs

  fprintf('\nReomputing wavelets with t=%g, m=%g\n',t,m);
  d=sgwt_delta(N,jcenter);
  fprintf('Computing wavelet by naive forward transform\n');
  figure(4)
  set(gcf,'position',[0,100,400,400])
  wp_e=sgwt_ftsd(d,gb,t,L);
  show_wavelet(wp_e,x,y);
  % show wavelet (naive)
  title('exact wavelet (naive forward transform)');
  fprintf('Computing wavelet by Chebyshev approximation\n');
  figure(5)
  set(gcf,'position',[425,100,400,400])
  % show wavelet (chebyshev)
  wp_c=sgwt_cheby_op(d,L,c,arange);
  show_wavelet(wp_c,x,y);
  title('approximate wavelet (transform via chebyshev approximation)');
  relerr=norm(wp_e-wp_c)/norm(wp_e);
  fprintf('Relative error between exact and approximate wavelet %g\n',relerr)
end

function show_wavelet(wp,x,y)
[Fs,s_ind]=sort(abs(wp),'descend');
scatter(x(s_ind),y(s_ind),msize,wp(s_ind),'.');
caxis([-1 1]*max(abs(wp)));
hcb=colorbar('location','north');
set(gca,'Xtick',[]);
set(gca,'Ytick',[]);
cxt=get(hcb,'Xtick');
cxt=[cxt(1),0,cxt(end)];
set(hcb,'Xtick',cxt);
cpos=get(hcb,'Position');
cpos(4)=.02; % make colorbar thinner
set(hcb,'Position',cpos);
axis equal
axis off
end

function update_uitext
set(ttexth,'string',sprintf('t=%0.3f',t));
set(mtexth,'string',sprintf('m=%g',m));
```

```
set(jtexth,'string',sprintf('j=%g',jcenter));
end

function tslider_callback(source,eventdata)
t=get(tsliderh,'value');
update_uitext;
update_kernel;
update_waveletfigs;
end

function mslider_callback(source,eventdata)
newm=get(msliderh,'value');
if newm<m
  m=floor(newm);
else
  m=ceil(newm);
end
set(msliderh,'value',m);
update_uitext;
update_kernel;
update_waveletfigs;
end

function jbutton_callback(source,eventdata)
figure(2)
fprintf('Select new center vertex\n');
[xp,yp]=ginput(1);
oldjcenter=jcenter;
jcenter=argmin((xp-x).^2+(yp-y).^2);
scatter(x(jcenter),y(jcenter),msize,'r.');
scatter(x(oldjcenter),y(oldjcenter),msize,[.5 .5 .5],'.');
drawnow
update_uitext
update_waveletfigs
end

end

function i = argmin(x)
  i=min(find(x==min(x)));
end



function modules = simpleSpectralPartitioning(adj,k)

% find the Fiedler vector: eigenvector corresponding to the second smallest eigenvalue of the
Laplacian matrix
fv = fiedlerVector(adj);
[~,I]=sort(fv);

% depending on k, partition the nodes
if nargin==1

    modules{1}=[]; modules{2}=[];
    % choose 2 groups based on signs of fv components
    for v=1:length(fv)
        if fv(v)>0; modules{2} = [modules{2}, v]; end
        if fv(v)<=0; modules{1} = [modules{1}, v]; end
    end
```

```
end

if nargin==2

  k = [0 k];  % adding 0 to aid indexing in line 43

  for kk=1:length(k)

    modules{kk}=[];
    for x=1:k(kk);
       modules{kk} = [modules{kk} I( x+sum(k(1:(kk-1))) )];
    end

  end

  modules = modules(2:length(modules));  % removing the "0" module
end

set(gcf,'Color',[1 1 1])
subplot(1,2,1)
plot(fv(I),'k.');
xlabel('index i')
ylabel('fv(i)')
title('sorted fiedler vector')
axis tight
axis square

subplot(1,2,2)
spy(adj(I,I),'k.')
axis square
title('sorted adjacency matrix')

%print spec_part_example.pdf
```