

A New Approach for Automatic Sleep Staging: Siamese Neural Networks

Enes Efe^{1*}, Seral Özşen²

¹ Department of Electrical and Electronics Engineering, Faculty of Engineering, Hitit University, Corum 19030, Turkey

² Department of Electrical and Electronics Engineering, Faculty of Engineering and Natural Sciences, Konya Technical University, Konya 42130, Turkey

Corresponding Author Email: enesefe@hitit.edu.tr



<https://doi.org/10.18280/ts.380517>

ABSTRACT

Received: 13 September 2021

Accepted: 16 October 2021

Keywords:

electroencephalogram (EEG), Siamese neural networks (SNNs), automatic sleep staging, convolutional neural networks (CNNs), classification, data augmentation

Sleep staging aims to gather biological signals during sleep, and categorize them by sleep stages: waking (W), non-REM-1 (N1), non-REM-2 (N2), non-REM-3 (N3), and REM (R). These stages are distributed irregularly, and their number varies with sleep quality. These features adversely affect the performance of automatic sleep staging systems. This paper adopts Siamese neural networks (SNNs) to solve the problem. During the network design, seven distance measurement methods, namely, Euclidean, Manhattan, Jaccard, Cosine, Canberra, Bray-Curtis, and Kullback Leibler divergence (KLD), were compared, revealing that Bray-Curtis (83.52%) and Cosine (84.94%) methods boast the best classification performance. The results of our approach are promising compared to traditional methods.

1. INTRODUCTION

Sleep is as important to human life as essential elements like water and food [1]. Sleep slows down and relaxes our biological processes, making us feel physically stronger when we wake up [2]. However, these functions could be disrupted by missing or excessive sleep time. The disruption of sleep hours causes various disorders to the body [3]. To prevent the disorders, the physiological data of the patients are recorded in sleep labs, and used to make correct diagnosis, and select appropriate treatment methods. The recording is usually performed with a device called polysomnography (PSG), which allows for detailed monitoring of the stages and physiological parameters of sleep, as well as the functions and interactions of various organ systems during sleep and wakefulness [4].

Sleep staging aims to gather biological signals during sleep, and categorize them by sleep stages. Two basic standards are preferred for sleep staging, namely, American Academy of Sleep Medicine (AASM) standard [5], and Rechtschaffen and Kales (R&K) standard [6]. The AASM standard is recommended to process electroencephalogram (EEG), electrooculogram (EOG), and electromyogram (EMG) recordings. These biological signals are categorized by sleep stages: waking (W), non-REM-1 (N1), non-REM-2 (N2), non-REM-3 (N3), and REM (R), with REM being short for rapid eye movements. Each stage can be separated into 30s-long epochs. The sleep/wake intervals are split once the sleep stages have been identified [7, 8].

The physiological data recorded by PSG are evaluated by medical specialists. The evaluation mainly aims to determine whether the patient is asleep, and the specific stage of his/her sleep during the night [7]. But the evaluation process is long, laborious, and prone to human errors, calling for automatic sleep staging systems are needed. As a result, automatic sleep staging has been studied extensively each year, using data from multiple channels (e.g., EEG, EOG, and EMG) or a

single channel [9-13]. Single-channel signals enable light, wearable, and portable devices that does not affect sleep quality, because they require fewer electrodes and connections than multi-channel signals [10]. EEG signals are commonly favored in the literature for two reasons: First, EEG signals are not deterministic, i.e., the frequency and level content are not consistent for a long time; Second, EEG signals do not have specific forms like electrocardiogram (ECG) signals. EEG signals are commonly investigated by statistical and parametric analysis methods, such as cross-correlation, time-frequency analysis, and auto-correlation [14].

During automatic sleep staging, the time, frequency, and time-frequency domains are utilized to extract features from each epoch of the signals to be employed. The extracted time features, frequency features, and nonlinear features [15] are utilized to train classifiers that predict the stage of sleep [13]. This popular approach is recommended for networks with traditional machine learning classifiers. For instance, some scholars [12, 16-18] extracted features through continuous wavelet transform and Hilbert-Huang transform (HHT), and introduced contemporary mathematical methods to networks with classic machine learning classifiers, namely, support vector machine (SVM), random forest (RF), or k-nearest neighbors (k-NN).

Since the above approach is time-consuming and tedious, deep learning algorithms like convolutional neural networks (CNNs) have lately been adopted to automatically extract features from input signals. However, neither traditional classifiers with manually extracted features [12, 16-20] nor classifiers with features automatically mined by deep learning [13, 21-23] cannot effectively work on unbalanced datasets. This is because conventional classifier networks require a large amount of balanced data from each class [24]. The problem could be solved by Siamese neural networks (SNNs), which do well on unbalanced data. In the 1990s, Bromley et al. [25] were the first to adopt the SNNs for signature verification. In 2005, Chopra et al. formalized Siamese architecture by

applying CNN to face verification based on raw images [26]. More recently, SNNs have been successfully implemented in various fields, such as image analysis [27], speech processing [28], biology [29], optics and physics [30], and medicine and health [31].

This paper employs the SNNs because of its excellence on unbalanced data. During network design, seven distance measurement methods were selected to compute the similarity score, including Euclidean, Manhattan, Jaccard, Cosine, Canberra, Bray-Curtis, and Kullback-Leibler divergence (KLD). Data augmentation was introduced to increase the data size for comparison. In this way, a new competitive method was derived for automatic sleep staging based on deep learning and SNNs. To the best of our knowledge, it is the first time to develop such a method in the field of sleep staging. Besides, the proposed method was proved suitable for deep learning-based automatic sleep staging systems, providing a competitive new approach for automatic sleep staging.

The remainder of this paper is organized as follows: Section 2 explains the dataset, network, and analysis methods; Sections 3-5 compare and evaluate the performance of the proposed approach.

2. METHODOLOGY

2.1 Dataset and data preparation

The PhysioNet Sleep EDF database [32] is widely adopted in the research of automatic sleep staging [33]. The dataset contains 61 nocturnal polysomnography records of 42 people, which are sampled at the rate of 100Hz. The records include EEG, EOG, and EMG signals, as well as event markers. The dataset was established on two studies: sleep cassette (SC) and sleep telemetry (ST). The former investigates the effect of age on healthy people, and the latter investigates the effect of temazepam on sleep.

The recordings were evaluated by sleep staging experts in 30s epochs, according to the R&K standard [34]. During the staging phase, the following labels were used: W, N1, N2, N3, N4, REM, Movement, and Unknown. Each EEG recording was acquired by Fpz-Cz and Pz-Oz electrodes. The recordings from Fpz-Cz electrodes were utilized, because they were crisper than those from Pz-Oz electrodes in the Sleep-EDF database [12, 35]. Firstly, the Movement and Unknown data were eliminated from the dataset. Next, the N3 stage was merged with the N4 stage by the AASM standard, reducing the total number of stages from 6 to 5 (W, N1, N2, N3 / N4, REM).

2.2 Overlap technique

The overlap method is adopted more widely and more successfully than the other strategies [36-39], owing to the following advantages: the method is simple to use and reproduce; the current training set can be expanded several times, reducing the size of each training sample; the resulting trained network will have a better translational invariance. For these reasons, this paper chooses the overlap method [38]. Firstly, each epoch belonging to the same class was combined, producing a long signal. Next, the long signal was processed by overlapping rectangular windows of a certain duration [39]. This procedure is depicted in Figure 1.



Figure 1. Data augmentation by overlap technique

2.3 SNNs

Traditional deep networks need hundreds of labeled data in each class to realize classification. Take a dataset with three labels, i.e., cars, planes, and birds, for example. If only trained by images in the three classes, the neural network cannot work effectively on a new class, e.g., trucks. Then, lots of truck images must be added to the dataset to retain the network. However, the addition and retaining are often time-consuming and costly [40]. Thus, SNNs have been developed to solve the classification of unbalanced data.

Every SNN consists of two identical neural networks, each of which can learn the hidden representation of an input vector [25]. The two networks are identical in that they share the same setup, including parameters and weights. The data belonging to the same class or two different classes are imported to the two networks. Then, the SNN produces two vectors that represent the two input data in lower dimensions. The distance between the two vectors is calculated by a distance measurement method. The greater the distance, the less similarity between the two input data. For this reason, a purely empirical threshold is determined for comparison. The distance between the two eigenvectors varies with the distance measurement methods, for each method has a unique equation. Therefore, the optimal threshold, that is, the threshold leading to the highest accuracy on the training set, depends on the specific method for distance measurement [41].

To implement all the above processes, the SNN needs to be trained through pairwise learning. Therefore, the cross-entropy loss function must be replaced with the comparative loss function [42]:

$$L(y, d) = \frac{1}{2}(y \times d + (1 - y) \times \max\{m - d, 0\}) \quad (1)$$

where, d is the distance between the two input eigenvectors; y is the binary output; m is the margin. If the input eigenvectors are dissimilar, they cannot contribute to the loss function, unless their distance is within the margin.

The SNN can work on different distance measurement methods. Nevertheless, not every method ensures the good performance of the network. Thus, it is very important to know which method does better in a specific scenario, and choose the most suitable method for distance measurement. For example, the performance of the Euclidean distance method decreases with the growing data size, while the cosine distance method increases with the size of the dataset. In addition, the threshold should be adjusted according to the selected method. For this purpose, the SNN must be pre-trained, and the most suitable distance measurement method and threshold must be selected. The architecture of the SNN is illustrated in Figure 2.

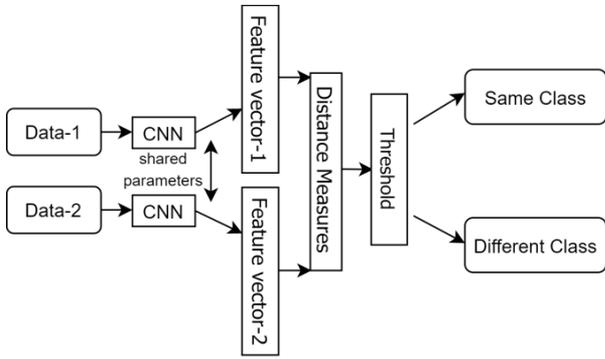


Figure 2. Architecture of SNN

This paper calls the Adam optimizer to iteratively update the network weights based on the training data. This optimization technique was selected to replace the traditional SNN training algorithm of stochastic gradient descent. As shown in Figure 3, the Adam optimizer uses two identical CNNs, which are responsible for acquiring the eigenvectors. Moreover, the comparative loss function was adopted to evaluate the ability of the SNN to differentiate between the two data.

The CNNs, a specific form of linear operation, are a basic neural network that employs convolution instead of matrix multiplication in at least one layer [43]. Each CNN consists of blocks that are added one after the other to learn complex features, and each block extracts the features from the data on the previous blocks. During the operation, the convolution layer (Layer C) learns simple features, and uses nonlinear activation functions to learn increasingly complex features. Then, the pooling layer (Layer P) brings the key information in the data to the foreground. Figure 3 shows the structure of a CNN in the SNN.

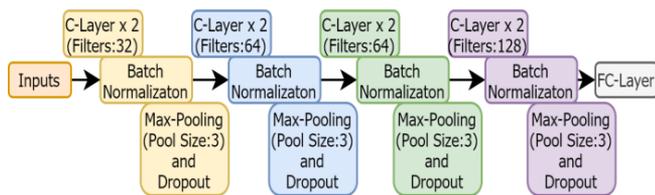


Figure 3. Structure of a CNN in our SNN

To realize a fair comparison with the SNN, the softmax function was added to the last layer of each CNN in the SNN, so as to perform the traditional classification.

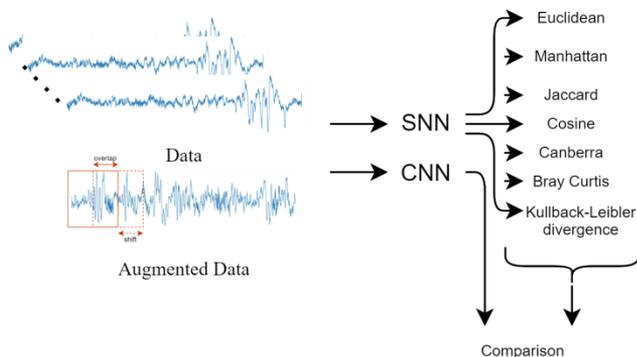


Figure 4. Flow of the proposed system

Figure 4 shows the flow of the proposed system, which

compares the CNN model with the SNN model created with each distance measurement method on augmented and non-augmented data. Firstly, seven different distance measurement methods were used separately in the SNN, using the sleep staging dataset, and the method with the best performance was determined. Then, the best-performing SNN model was compared with the CNN model on the same dataset.

2.4 Distance measures

2.4.1 Euclidean distance

In artificial intelligence, Euclidean distance is the most widely metric of the distance between two points [44]. Figure 5 depicts the calculation of Euclidean distance. Euclidean distance can be calculated by the Pythagorean theorem:

$$D(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2)$$

where, x and y are the Cartesian coordinates of each point.

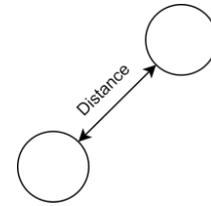


Figure 5. Euclidean distance between 2 points

2.4.2 Manhattan distance

As shown in Figure 6, Manhattan distance might perform worse than Euclidean distance, for the failure to give the shortest distance. But some scholars found this measure outperforming Euclidean distance [45]. Manhattan distance can be calculated without any diagonal movement:

$$D(x, y) = \sqrt{\sum_{i=1}^n |x_i - y_i|} \quad (3)$$

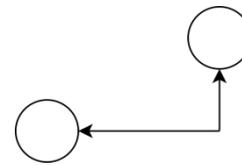


Figure 6. Manhattan distance between 2 points

2.4.3 Jaccard distance

The Jaccard distance statistically evaluates the similarity between two clusters. As shown in Figure 7, the intersection of the two sets can be identified by dividing with the total number of elements. If the two sets are the same, the intersection is 1; if the two sets have no common feature, the intersection is 0.

To calculate the Jaccard distance, it is necessary to subtract the Jaccard index from 1, for the distance is inversely proportional to similarity. The Jaccard distance between two points can be calculated by:

$$D(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|} \quad (4)$$

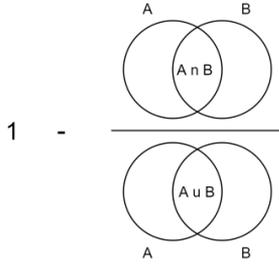


Figure 7. Jaccard distance between 2 samples

2.4.4 Cosine distance

The cosine angle between two vectors in a multidimensional space is a yardstick of the similarity between these vectors. If the two vectors have the same orientation, the cosine similarity is 1; if the two vectors have diametrically opposite orientations, the cosine similarity is -1. Note that cosine similarity only considers the direction of the vectors, without accounting their magnitude [46]. As shown in Figure 8, the cosine distance can be calculated by subtracting 1 from cosine similarity:

$$D(x, y) = 1 - \cos(\theta) = 1 - \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}} \quad (5)$$

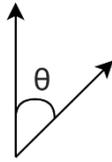


Figure 8. Cosine similarity between 2 samples

2.4.5 Canberra distance

The Canberra distance numerically measures the separation between two points in a vector space. If the coordinates of both samples are close to zero, the Canberra distance will be sensitive to tiny changes [47]. Mathematically, this distance measure can be defined as:

$$D(x, y) = \sum_{i=1}^n \frac{|x_i - y_i|}{|x_i| + |y_i|} \quad (6)$$

2.4.6 Bray-Curtis distance

Bray-Curtis distance is not technically a metric, as it does not provide the triangle inequality property. But it is a common way to measure the difference between samples. If the coordinates of both samples are close to zero, this measure is meaningless. Mathematically, this distance measure can be defined as:

$$D(x, y) = \frac{\sum_{i=1}^n |x_i - y_i|}{\sum_{i=1}^n (x_i + y_i)} \quad (7)$$

2.4.7 KLD

The KLD formulates the distance between two probability distributions. Like Bray-Curtis distance, the LKD is not a metric, because it does not satisfy the triangle inequality property. By the KLD, the ratios of the two distributions to

each other at each point are taken, and made equal to the sum of the logarithms of the ratios. If the two distributions are the same, the distance is 0; otherwise, the distance is a positive real number.

$$D(p \parallel q) = \sum_{i=1}^n p(x_i) \times (\log p(x_i) - \log q(x_i)) \quad (8)$$

where, $q(x)$ is the approximation; $p(x)$ is the true distribution.

3. EVALUATION

3.1 Confusion matrix

The Confusion matrix (Table 1) is a popular approach to evaluate model performance. The performance metrics like accuracy, specificity, and sensitivity can be calculated based on the number of samples assigned to the proper classes and the number of samples assigned to the incorrect classes.

Table 1. Confusion matrix

Actual Class	Predicted Class	
	Positive	Negative
	Pos. True Positive (TP) Neg. False Positive (FP)	False Negative (FN) True Negative (TN)

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (9)$$

$$Specificity = \frac{TN}{TN + FP} \quad (10)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (11)$$

Since the SNN is a binary classifier, its performance is generally evaluated by metrics like accuracy (9), specificity (10), and sensitivity (11).

3.2 Hold-out and cross validation methods

Hold-out and cross validation are two widely adopted techniques by researchers of automatic sleep staging. The hold-out technique separates the data into a training set and a test set. The performance of the target model is evaluated on the previously untrained test set, and trained on the training set. Normally, the training set and test set are split by the ratio of 80%:20%. Of course, this ratio varies with the data sizes.

Cross validation divides the original dataset into k groups. One of the groups is taken as the test set, and the other groups as training sets. Because training is done on several training and test sets, cross validation could predict the model performance on an unknown dataset. When the dataset is large, however, cross validation will involve many more computations, and thus consume much more time than the hold-out technique.

Considering the high computing and processing requirements of the SNN, the hold-out technique might be preferred. In this paper, the hold-out technique is chosen to process a total of 56,764 sleep stage data (Wake: 14984, NREM 1: 5581, NREM 2: 22676, NREM 3: 5197, NREM 4: 8326), each with a length of 3000 samples.

4. RESULTS AND DISCUSSION

As shown in Figure 2, SNNs were created, the input data were encoded, and the distance between the eigenvectors were measured, using seven different methods: Euclidean, Manhattan, Jaccard, Cosine, Canberra, Bray-Curtis, and KLD. Then, the optimal threshold was calculated empirically for each method, giving the highest classification accuracy. Specifically, numbers in the range of 0-1 were tested on the training dataset with 0.025 intervals, and the value giving the highest accuracy was accepted as the optimal threshold. The results obtained are shown in Table 2.

Table 2. Optimal thresholds

Distance measurement methods	Threshold values
Euclidean	0.55
Manhattan	0.525
Jaccard	0.5
Cosine	0.525
Canberra	0.6
Bray Curtis	0.6
KLD	0.425

According to the classification results (Table 3) on the data in five different classes (0: W, 1: N1, 2: N2, 3: N3, 4: N4), the SNN with Bray-Curtis achieved the best performance. The classification results of this SNN are reported in Table 4.

Table 3. Classification results with different distance measurement methods

	Sensitivity (%)	Specificity (%)	Accuracy (%)
Euclidean	84.83	81.16	82.89
Manhattan	84.25	77.96	80.79
Jaccard	83.43	80.72	82.02
Cosine	82.49	82.34	82.42
Canberra	85.37	80.97	83.03
Bray Curtis	86.02	81.35	83.52
KLD	71.15	62.32	65.57

Table 4. Binary classification results with Bray-Curtis

Stages	Sensitivity (%)	Specificity (%)	Accuracy (%)
0 vs 1	81.21	74.16	77.24
0 vs 2	86.13	95.86	90.42
0 vs 3	93.79	99.39	96.42
0 vs 4	90.05	90.57	90.31
1 vs 2	72.33	63.82	67.07
1 vs 3	88.23	96.10	91.80
1 vs 4	71.26	58.48	62.13
2 vs 3	87.07	78.75	82.38
2 vs 4	83.43	78.05	80.50
3 vs 4	93.95	98.49	96.11

Table 5. Results on augmented dataset

	Sensitivity (%)	Specificity (%)	Accuracy (%)
Euclidean	85.22	83.42	84.29
Manhattan	87.52	79.03	82.74
Jaccard	83.71	82.82	83.26
Cosine	85.65	84.25	84.94
Canberra	87.38	82.26	84.63
Bray Curtis	88.01	82.11	84.81
KLD	73.45	60.57	64.57

Table 6. Binary classification results with cosine distance

Stages	Sensitivity (%)	Specificity (%)	Accuracy (%)
0 vs 1	80.98	77.21	78.97
0 vs 2	84.29	96.53	89.48
0 vs 3	93.54	99.26	96.22
0 vs 4	90.06	92.36	91.18
1 vs 2	73.11	68.88	70.78
1 vs 3	87.65	97.45	91.99
1 vs 4	74.78	62.31	66.45
2 vs 3	85.23	79.62	82.18
2 vs 4	83.09	82.69	82.89
3 vs 4	94.38	98.71	96.44

As shown in Table 4, the SNN with Bray-Curtis obtained the best performance, when 0 and 3 stages were given together to the network. Then, the data were approximately doubled using the overlapping technique, and the classification results of the SNN with different distance measures are shown in Table 5. In this case, the best performance corresponded to the cosine distance measure. The classification results of the SNN with cosine distance are given in Table 6.

Finally, the results obtained by one of the identical parallel CNNs in our SNN were compared with those of the traditional classification method (subsection 2.3). The traditional method was evaluated on the same datasets as the SNN, an augmented set and a non-augmented set. The former set is about two times the size of the latter set. As shown in Table 7, the classification performance on augmented set was better than that on non-augmented set. Tables 8 and 9 provide the details on the binary classification results on augmented set.

Table 7. Classification results obtained using the CNN

	Sensitivity (%)	Specificity (%)	Accuracy (%)
Dataset	75.43	77.62	82.08
Aug. Dataset	78.89	79.32	83.8

Table 8. Binary classification results obtained using the CNN

Stages	Sensitivity (%)	Specificity (%)	Accuracy (%)
0 vs 1	94.59	30.55	77.56
0 vs 2	94.59	86.70	89.84
0 vs 3	94.59	89.61	93.31
0 vs 4	94.59	75.71	87.95
1 vs 2	30.55	86.70	75.84
1 vs 3	30.55	89.61	59.49
1 vs 4	30.55	75.71	57.62
2 vs 3	86.70	89.61	88.83
2 vs 4	86.70	75.71	83.80
3 vs 4	89.61	75.71	81.15

Table 9. Results obtained using the CNN on augmented dataset

Stages	Sensitivity (%)	Specificity (%)	Accuracy (%)
0 vs 1	91.43	51.04	80.74
0 vs 2	91.43	88.82	89.87
0 vs 3	91.43	88.44	90.68
0 vs 4	91.43	74.71	85.61
1 vs 2	51.04	88.82	81.44
1 vs 3	51.04	88.44	69.04
1 vs 4	51.04	74.71	65.17
2 vs 3	88.82	88.44	88.75
2 vs 4	88.82	74.71	85.09
3 vs 4	88.44	74.71	80.00

During the analysis of Tables 3 and 5, it should be noted that the SNNs with distance measures other than KLD and Manhattan achieved close results, because neural networks are stochastic algorithms. In other words, the similarity of the results cannot reveal the superiority between the techniques, but suggest that these techniques take advantage of stochasticity, such as the random initialization of weights. This means the same network can produce different results despite being trained on the same data.

Furthermore, although the SNN outperformed the CNN, the network did not perform better in the binary classification of all sleep stages. Compares Table 4 and 6 with Tables 8 and 9, it is evident that the CNN outshines the SNN in 0 vs 1, 1 vs 2, 2 vs 3, and 2 vs 4.

Table 10. Comparison between our SNN and previous state-of-the-art results

Study	Sensitivity (%)	Precision (%)	Accuracy (%)
This work	85.65	83.93	84.94
[48]	74	-	83
[49]	74	91	82
[50]	75.8	77.3	84.5
[13]	82.49	78.6	82
[22]	73.9	73.7	81.9
[35]	-	-	83.78

Table 10 compares the proposed SNN with the existing methods. The highest values are shown in bold font. Overall, it can be said that data augmentation improves the performance of both SNN and CNN.

In conclusion, the SNNs using binary classification methods are much better than traditional methods. Since EEG signals can be easily obtained from the forehead with a dry electrode, our method bodes well for developing low-cost and portable high-performance devices in the future. Meanwhile, the SNN system must be supported by the hold-out technique to solve the high requirements of random-access memory (RAM). If the memory issue can be solved in future, the results can be evaluated through the k-fold cross-validation. In addition, more approaches can be explored, and the system performance can be compared with the system proposed here.

5. CONCLUSIONS

Sleep quality varies greatly from person to person, making it impossible to obtain an equal number of balanced data from each stage of sleep. This paper mainly intends to solve the classification problem of unbalanced datasets in automatic sleep-staging systems, with the aid of the SNNs. The proposed SNN was compared with traditional classification methods. The comparison shows that the SNN outshined conventional methods with 84.94% accuracy, 84.25% specificity, and 85.65% sensitivity. This innovative approach for automatic sleep staging is promising for future studies.

REFERENCES

[1] Brain Basics: Understanding Sleep. NIH Publication, 2014: pp. 6-3440.
 [2] Horne, J. (1978). A review of the biological effects of total sleep deprivation in man. *Biological Psychology*, 17(1-2): 55-102. [https://doi.org/10.1016/0301-0511\(78\)90042-x](https://doi.org/10.1016/0301-0511(78)90042-x)

[3] Bandyopadhyay, A., Sigua, N.L. (2019). What is sleep deprivation? *American Journal of Respiratory and Critical Care Medicine*, 199(6): 11-12. <https://doi.org/10.1164/rccm.1996P11>
 [4] Köktürk, O. (2010). Diagnostic methods and polysomnography in sleep respiratory disorders. *Respiratory system and diseases*. Özlü T, Metintaş M, Karadağ M, Kaya A (Eds). Istanbul: Istanbul Medicine Bookstore, 2109-2125.
 [5] Berry, R. (2013). The AASM manual for the scoring of sleep and associated events: Rules, terminology and technical specifications. version 2.0. 2 Darien. Illinois American Academy of Sleep Medicine.
 [6] Hori, T., Sugita, Y., Koga, E., Shirakawa, S., Inoue, K., Uchida, S., Kuwahara, H., Kousaka, M., Kobayashi, T., Tsuji, Y., Terashima, M., Fukuda, K., Fukuda, N. (2001). Proposed supplements and amendments to ‘A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects’, the Rechtschaffen & Kales (1968) standard. *Psychiatry and Clinical Neurosciences*, 55(3): 305-310. <https://doi.org/10.1046/j.1440-1819.2001.00810.x>
 [7] Berry, R.B., Budhiraja, R., Gottlieb, D.J., Gozal, D., Iber, C., Kapur, V.K., Marcus, C.L., Mehra, R., Parthasarathy, S., Quan, S.F., Redline, S., Strohl, K.P., Davidson Ward, S.L., Tangredi, M.M. (2012). Rules for scoring respiratory events in sleep: Update of the 2007 AASM manual for the scoring of sleep and associated events: deliberations of the sleep apnea definitions task force of the American Academy of Sleep Medicine. *Journal of Clinical Sleep Medicine*, 8(5): 597-619. <https://doi.org/10.5664/jcsm.2172>
 [8] Rosenberg, R.S., Van Hout, S. (2014). The American Academy of Sleep Medicine inter-scorer reliability program: Respiratory events. *Journal of Clinical Sleep Medicine*, 10(4): 447-454. <https://doi.org/10.5664/jcsm.3630>
 [9] Biswal, S., Kulas, J., Sun, H., Goparaju, B., Westover, M.B., Bianchi, M.T., Sun, J. (2017). SLEEPNET: Automated sleep staging system via deep learning. *arXiv preprint arXiv:1707.08262*.
 [10] Sors, A., Bonnet, S., Mirek, S., Vercueil, L., Payen, J. (2018). A convolutional neural network for sleep stage scoring from raw single-channel EEG. *Biomedical Signal Processing and Control*, 42: 107-114. <https://doi.org/10.1016/j.bspc.2017.12.001>
 [11] Andreotti, F., Phan, H., De Vos, M. (2018). Visualising convolutional neural network decisions in automatic sleep scoring. *CEUR Workshop Proceedings*.
 [12] Huang, W., Guo, B., Shen, Y., Tang, X., Zhang, T., Li, D., Jiang, Z. (2020). Sleep staging algorithm based on multichannel data adding and multifeature screening. *Computer Methods and Programs in Biomedicine*, 187: 105253. <https://doi.org/10.1016/j.cmpb.2019.105253>
 [13] Supratak, A., Dong, G., Wu, C., Guo, Y. (2017). DeepSleepNet: A model for automatic sleep stage scoring based on raw single-channel EEG. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(11): 1998-2008. <https://doi.org/10.1109/TNSRE.2017.2721116>
 [14] Kıymık, M.K., Güler, I., Dizibüyük, A., Akin, M. (2005). Comparison of STFT and wavelet transform methods in determining epileptic seizure activity in EEG signals for

- real-time application. *Computers in Biology and Medicine*, 35(7): 603-616. <https://doi.org/10.1016/j.compbiomed.2004.05.001>
- [15] Radha, M., Garcia-Molina, G., Poel, M., Tononi, G. (2014). Comparison of feature and classifier algorithms for online automatic sleep staging based on a single EEG signal. 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 1867-1880. <https://doi.org/10.1109/EMBC.2014.6943976>
- [16] Liao, Y., Zhang, M., Wang, Z., Xie, X. (2020). Tri-FeatureNet: An adversarial learning-based invariant feature extraction for sleep staging using single-channel EEG. 2020 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1-5. <https://doi.org/10.1109/ISCAS45731.2020.9180501>
- [17] Tabar, Y.R., Mikkelsen, K.B., Rank, M.L., Hemmsen, M.C., Kidmose, P. (2021). Investigation of low dimensional feature spaces for automatic sleep staging. *Computer Methods and Programs in Biomedicine*, 205: 106091. <https://doi.org/10.1016/j.cmpb.2021.106091>
- [18] Hassan, A.R., Bhuiyan, M.I.H. (2016). A decision support system for automatic sleep staging from EEG signals using tunable Q-factor wavelet transform and spectral features. *Journal of Neuroscience Methods*, 271: 107-118. <https://doi.org/10.1016/j.jneumeth.2016.07.012>
- [19] Yi, L., Fan, Y.L., Li, G., Tong, Q.Y. (2009). Sleep stage classification based on EEG Hilbert-Huang transform. in 2009 4th IEEE Conference on Industrial Electronics and Applications. 2009. <https://doi.org/10.1109/ICIEA.2009.5138842>
- [20] Guo, C., Lu, F., Liu, S., Xu, W. (2015). Sleep EEG staging based on Hilbert-Huang transform and sample Entropy. 2015 International Conference on Computational Intelligence and Communication Networks (CICN), pp. 442-445. <https://doi.org/10.1109/CICN.2015.92>
- [21] Phan, H., Chen, O.Y., Tran, M.C., Koch, P., Mertins, A., De Vos, M. (2021). XSleepNet: Multi-view sequential model for automatic sleep staging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. <https://doi.org/10.1109/TPAMI.2021.3070057>
- [22] Phan, H., Andreotti, F., Cooray, N., Chen, O.Y., De Vos, M. (2018). Joint classification and prediction CNN framework for automatic sleep stage classification. *IEEE Transactions on Biomedical Engineering*, 66(5): 1285-1296. <https://doi.org/10.1109/TBME.2018.2872652>
- [23] Nasiri, S., Clifford, G.D. (2020). Attentive adversarial network for large-scale sleep staging. *Machine Learning for Healthcare Conference*. PMLR.
- [24] Zhang, C., Liu, W., Ma, H., Fu, H. (2016). Siamese neural network based gait recognition for human identification. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2832-2836. <https://doi.org/10.1109/ICASSP.2016.7472194>
- [25] Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R. (1993). Signature verification using a "siamese" time delay neural network. *Proceedings of the 6th International Conference on Neural Information Processing Systems*, pp. 737-744.
- [26] Chopra, S., Hadsell, R., LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), pp. 539-546. <https://doi.org/10.1109/CVPR.2005.202>
- [27] Taigman, Y., Yang, M., Ranzato, M., Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1701-1708. <https://doi.org/10.1109/CVPR.2014.220>
- [28] Lian, Z., Li, Y., Tao, J., Huang, J. (2018). Speech emotion recognition via contrastive loss under siamese networks. *Proceedings of the Joint Workshop of the 4th Workshop on Affective Social Multimedia Computing and first Multi-Modal Affective Computing of Large-Scale Multimedia Data*.
- [29] Swati, Gupta, G., Yadav, M., Sharma, M., Vig, L. (2017). Siamese networks for chromosome classification. 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 72-81. <https://doi.org/10.1109/ICCVW.2017.17>
- [30] Zou, Y., Li, J., Chen, X., Lan, R. (2018). Learning Siamese networks for laser vision seam tracking. *J Opt Soc Am A Opt Image Sci Vis*, 35(11): 1805-1813. <https://doi.org/10.1364/JOSAA.35.001805>
- [31] Zeng, X., Chen, H., Luo, Y., Ye, W. (2019). Automated diabetic retinopathy detection based on binocular siamese-like convolutional neural network. *IEEE Access*, 7: 30744-30753. <https://doi.org/10.1109/ACCESS.2019.2903171>
- [32] Goldberger, A.L., Amaral, L.A., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.K., Stanley, H.E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23): e215-e220. <https://doi.org/10.1161/01.cir.101.23.e215>
- [33] Boostani, R., Karimzadeh, F., Nami, M. (2017). A comparative review on sleep stage classification methods in patients and healthy individuals. *Computer Methods and Programs in Biomedicine*, 140: 77-91. <https://doi.org/10.1016/j.cmpb.2016.12.004>
- [34] Wolpert, E.A. (1968). A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects. *Arch Gen Psychiatry*, 20(2): 246-247. <https://doi.org/10.1001/archpsyc.1969.01740140118016>
- [35] Fu, M., Wang, Y., Chen, X., Li, J., Xu, F., Liu, X., Hou, F. (2021). Deep learning in automatic sleep staging with a single channel electroencephalography. *Frontiers in Physiology*, 12: 179. <https://doi.org/10.3389/fphys.2021.628502>
- [36] Chen, H., Hu, N., Cheng, Z., Zhang, L., Zhang, Y. (2019). A deep convolutional neural network based fusion method of two-direction vibration signal data for health state identification of planetary gearboxes. *Measurement*, 146: 268-278. <https://doi.org/10.1016/j.measurement.2019.04.093>
- [37] Tang, S., Yuan, S., Zhu, Y. (2020). Data preprocessing techniques in convolutional neural network based on fault diagnosis towards rotating machinery. *IEEE Access*, 8: 149487-149496. <https://doi.org/10.1109/ACCESS.2020.3012182>
- [38] Hendriks, J., Dumond, P. (2021). Exploring the relationship between preprocessing and hyperparameter

- tuning for vibration-based machine fault diagnosis using CNNs. *Vibration*, 4(2): 284-309. <https://doi.org/10.3390/vibration4020019>
- [39] Mousavi, Z., Rezaei, T.Y., Sheykhivand, S., Farzamina, A., Razavie, S.N. (2019). Deep convolutional neural network for classification of sleep stages from single-channel EEG signals. *Journal of Neuroscience Methods*, 324: 108312.
- [40] Li, F.F., Fergus, R., Perona, P. (2006). One-Shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4): 594-611. <https://doi.org/10.1109/TPAMI.2006.79>
- [41] Zinzuvadiya, M., Dhameliya, V., Vaghela, S., Patki, S., Nanavati, N., Bhavsar, A. (2020). Co-detection in images using saliency and siamese networks. In: Chaudhuri B., Nakagawa M., Khanna P., Kumar S. (eds) *Proceedings of 3rd International Conference on Computer Vision and Image Processing. Advances in Intelligent Systems and Computing*, vol 1024. Springer, Singapore. https://doi.org/10.1007/978-981-32-9291-8_28
- [42] De Baets, L., C., Develder, T., Dhaene, Deschrijver, D. (2019). Detection of unidentified appliances in non-intrusive load monitoring using siamese neural networks. *International Journal of Electrical Power & Energy Systems*, 104: 645-653. <https://doi.org/10.1016/j.ijepes.2018.07.026>
- [43] Goodfellow, I., Benhio, Y., Courville, A. (2016). *Deep Learning*. Vol. 1. MIT Press Cambridge.
- [44] Viriyavisuthisakul, S., Sanguansat, P., Charnkeitkong, P., Haruechaiyasak, C. (2015). A comparison of similarity measures for online social media Thai text classification. 2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp. 1-6. <https://doi.org/10.1109/ECTICon.2015.7207106>
- [45] Strauss, T., von Maltitz, M.J. (2017). Generalising Ward's method for use with Manhattan distances. *PloS One*, 12(1): e0168288. <https://doi.org/10.1371/journal.pone.0168288>
- [46] Liu, D., Chen, X., Peng, D. (2019). Some cosine similarity measures and distance measures between q -rung orthopair fuzzy sets. *International Journal of Intelligent Systems*, 34(7): 1572-1587. <https://doi.org/10.1002/int.22108>
- [47] Kumar, V., Chhabra, J.K., Kumar, D. (2014). Performance evaluation of distance metrics in the clustering algorithms. *INFOCOMP Journal of Computer Science*, 13(1): 38-52.
- [48] Fraiwan, L., Lweesy, K., Khasawneh, N., Wenz, H., Dickhaus, H. (2012). Automated sleep stage identification system based on time-frequency analysis of a single EEG channel and random forest classifier. *Computer Methods and Programs in Biomedicine*, 108(1): 10-19. <https://doi.org/10.1016/j.cmpb.2011.11.005>
- [49] Tsinalis, O., Matthews, P.M., Guo, Y., Zafeiriou, S. (2016). Automatic sleep stage scoring with single-channel EEG using convolutional neural networks. *arXiv preprint arXiv:1610.01683*.
- [50] Wei, L., Lin, Y., Wang, J., Ma, Y. (2017). Time-frequency convolutional neural network for automatic sleep stage classification based on single-channel EEG. 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI), pp. 88-95. <https://doi.org/10.1109/ICTAI.2017.00025>