Article

L-Tetrolet Pattern-Based Sleep Stage Classification Model Using Balanced EEG Datasets

Prabal Datta Barua [1,2], Ilknur Tuncer [3], Emrah Aydemir [4], Oliver Faust [5,*], Subrata Chakraborty [6,7], Vinithasree Subbhuraam [8], Turker Tuncer [9], Sengul Dogan [9] and U. Rajendra Acharya [10,11,12]

[1] School of Management & Enterprise, University of Southern Queensland, Darling Heights, QLD 4350, Australia
[2] Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia
[3] Elazig Governorship, Interior Ministry, Elazig 23119, Turkey
[4] Department of Management Information Systems, Management Faculty, Sakarya University, Sakarya 54050, Turkey
[5] School of Computing and Information Science, Anglia Ruskin University Cambridge Campus, Cambridge CB1 1PT, UK
[6] School of Science and Technology, Faculty of Science, Agriculture, Business and Law, University of New England, Armidale, NSW 2351, Australia
[7] Center for Advanced Modelling and Geospatial Information Systems, Faculty of Engineering and IT, University of Technology Sydney, Sydney, NSW 2007, Australia
[8] Egoscue Foundation, 12230 El Camino Real #110, San Diego, CA 92130, USA
[9] Department of Digital Forensics Engineering, Technology Faculty, Firat University, Elazig 23119, Turkey
[10] Ngee Ann Polytechnic, Department of Electronics and Computer Engineering, Singapore 599489, Singapore
[11] Department of Biomedical Engineering, School of Science and Technology, SUSS University, Singapore 599494, Singapore
[12] Department of Biomedical Informatics and Medical Engineering, Asia University, Taichung 41354, Taiwan
* Correspondence: oliver.faust@gmail.com

Abstract: Background: Sleep stage classification is a crucial process for the diagnosis of sleep or sleep-related diseases. Currently, this process is based on manual electroencephalogram (EEG) analysis, which is resource-intensive and error-prone. Various machine learning models have been recommended to standardize and automate the analysis process to address these problems. Materials and methods: The well-known cyclic alternating pattern (CAP) sleep dataset is used to train and test an L-tetrolet pattern-based sleep stage classification model in this research. By using this dataset, the following three cases are created, and they are: Insomnia, Normal, and Fused cases. For each of these cases, the machine learning model is tasked with identifying six sleep stages. The model is structured in terms of feature generation, feature selection, and classification. Feature generation is established with a new L-tetrolet (Tetris letter) function and multiple pooling decomposition for level creation. We fuse ReliefF and iterative neighborhood component analysis (INCA) feature selection using a threshold value. The hybrid and iterative feature selectors are named threshold selection-based ReliefF and INCA (TSRFINCA). The selected features are classified using a cubic support vector machine. Results: The presented L-tetrolet pattern and TSRFINCA-based sleep stage classification model yield 95.43%, 91.05%, and 92.31% accuracies for Insomnia, Normal dataset, and Fused cases, respectively. Conclusion: The recommended L-tetrolet pattern and TSRFINCA-based model push the envelope of current knowledge engineering by accurately classifying sleep stages even in the presence of sleep disorders.

Keywords: L-tetrolet pattern; sleep stage expert system; multiple pooling decomposition; insomnia; EEG signal classification

1. Introduction

People sleep an average of eight hours a day. This shows that almost one-third of human life is spent asleep [1–3]. Therefore, sleep quality plays an important role in our daily life. Today, people's sleep patterns are disrupted due to factors such as stress, intense work, and excessive use of multimedia devices [4–6]. Sleep disorders can negatively impact concentration, reducing task processing efficiency. Signals, such as electroencephalogram (EEG), electrocardiogram (ECG), and electrooculogram (EOG), are evaluated in people with sleep disorders. EEG signals are especially important for evaluating brain activity. EEG signals are also widely used in sleep scoring and the evaluation of sleep stages [7–10].

Two different standards are used for sleep scoring. They are the American Academy of Sleep Medicine (AASM) [11] and Rechtschaffen and Kales (R&K) [10]. The R&K standard was widely used from 1968 to 2007. Later, the sleep scoring guide was updated as the AASM standard [12].

A sleep cycle consists of the following six sleep phases: 1-W: wakefulness, 2–5-Stages (1–4): from light sleep to deep sleep, 6-REM: rapid eye movement. While the R&K standard accepts sleep stages according to this order, S3 and S4 are accepted as single stages in the AASM standard [13]. Manual identification of these stages is common during sleep disorders and sleep-related illness diagnoses [14]. This practice causes a high workload for human experts. Systems that automate sleep stage scoring are widely reported in the scientific literature [15–17]. These studies share the hypothesis that automated sleep stage classification can reduce the workload of human experts and ensure that errors due to environmental parameters are reduced [18–20]. However, automated sleep stage classification is difficult for machine learning and pattern recognition because sleep EEG datasets are heterogeneous.

We propose an L-tetrolet pattern-based sleep stage classification model that can extract transferable knowledge from heterogeneous EEG data. The popular cyclic alternating pattern (CAP) sleep EEG dataset was used to establish the sleep stage classification model. This dataset contains information from both insomniac and normal subjects, such as phase and sleep stages. Three cases were created to denote the general results of this dataset, and these cases consist of EEG signals of the insomniac subjects, normal subjects, and both insomniac and normal subjects, respectively. The proposed model could classify six sleep stages with an accuracy of 95.43%, 91.05%, and 92.31% for Insomnia, Normal dataset, and Fused cases, respectively.

Our main motivations were to propose a game-based feature extraction function and, by applying this function, present a new EEG signal classification model. To achieve that highly accurate learning model, a new L-tetrolet pattern and TSRFINCA-based sleep EEG signal classification model were created. The L-tetrolet pattern for textural feature extraction was inspired by the Tetris game. Statistical features were also extracted to enforce the presented feature generation method. A multilevel feature generation architecture was created using pooling functions to generate low-level and high-level features. The presented feature selector (TSRFINCA) incorporates three stages. In the first stage, a threshold point is determined, and feature selection is carried out by deploying this threshold point. ReliefF is applied to the selected features in the second stage, and the positive weighted feature is selected. In the last stage, iterative neighborhood component analysis (INCA) is applied to the selected features, and the most meaningful features are selected. The selected final features are utilized as the input of the cubic support vector machine (CSVM) classifier. To summarize, we proposed (i) a new game-based feature extractor, (ii) a new decomposition model by using four pooling techniques, and (iii) a hybrid high-performance feature selector. These methods have been used in a feature engineering model [21–23] to obtain high classification performance.

The novelties of our sleep stage classification model are given below as follows:

- L-tetrolet pattern: a new, Tetris-inspired, textural feature generation function;
- Statistical feature generator: created by fusing multiple pooling decomposers;
- TSRFINCA: a three-leveled hybrid and iterative feature selector.

Contributions:

- A new feature engineering model has been created by proposing new generation feature extraction, decomposition, and feature selection methods. The essential purpose of the proposed feature engineering model is to extract the most informative features from the used signals to obtain high classification performance with low time complexity.
- This research presents a highly accurate EEG classification model for sleep stage detection. By deploying the presented classification model, sleep stage classification results of the CAP sleep dataset are presented using three cases. Our proposal denotes general high classification performance since we applied this model to three different datasets.

The CAP Sleep Database on PhysioNet [24] is widely used in scientific work on sleep staging, and most of the published studies use the CAP database to establish the sleep phase [25–33]. Table 1 summarizes selected studies on sleep stage detection using different datasets.

Table 1. Literature review on sleep stage detection.

Studies	Method	Classifier	Dataset	Channels	The Results (%)
Abbasi et al. [34]	Convolutional Neural Network	Ensemble	Collected data	Multiple channels	Sensitivity: 78.44 Specificity: 96.49 Accuracy: 94.27
Li et al. [35]	Multi-Layer Convolutional Neural Networks	Auxiliary	SHHS dataset [36]	C3-A2, C4-A1, EOG	Accuracy: 85.12
Zaidi and Farooq [37]	Fourier Synchrosqueezed Transform Features	Support vector machine	DREAMS dataset	Cz-A1	Accuracy: 82.60
Sors et al. [38]	Deep Convolutional Neural Network	Convolutional Neural Network	The Sleep Heart Health Study dataset [39]	C4-A1, C3-A2	Accuracy: 87.00
Goshtasbi et al. [40]	Convolutional Neural Network	Softmax	SHHS dataset [36]	C4-A1, C3-A2	Accuracy: 81.30 Kappa: 74.00
Shahbakhti et al. [41]	Nonlinear Analysis	Linear discriminant analysis	DREAMS dataset [42]	Fp1, O1, and CZ or C3	Accuracy: 92.50 Sensitivity: 89.90 Specificity: 94.50
Zhao et al. [43]	SleepContextNet	Softmax	1. SHHS dataset [36] 2. CAP dataset [24,44]	C4-A1 and C3-A2	1. Accuracy: 86.40 Kappa: 81.00 2. Accuracy: 78.80 Kappa: 71.00
Eldele et al. [45]	Multi-Resolution Convolutional Neural Network, Adaptive Feature Recalibration	Softmax	SHHS dataset [36]	C4-A1	Accuracy: 84.20 Kappa: 78.00
Yang et al. [46]	One-Dimensional Convolutional Neural Network, Hidden Markov model	One-Dimensional Convolutional Neural Network, Hidden Markov model	DRM-SUB dataset [42]	Pz-Oz	Accuracy: 83.23 Kappa: 76.00

To support our novelty claims and to substantiate the key contributions, we have structured the manuscript as follows. The next section introduces the dataset used to design and test the sleep stage classification model. Section 3 outlines the processing methods that were used to implement and test the proposed sleep stage detection model. The model was evaluated with a set of experiments. Section 4 specifies these experiments and provides the corresponding results. The subsequent discussion section relates our results to the wider sleep research area. We also list limitations and future work before concluding the paper.

2. Material and Method

2.1. Material

The CAP sleep stage dataset is a widely used benchmark dataset. The dataset consists of EEG recordings during the Non-REM (NREM) sleep phase. These data were obtained

from 108 polysomnographic patients registered at the Sleep Disorders Center of the Ospedale Maggiore of Parma in Italy [24]. The data were recorded as .edf files [47]. Sleep data comprises at least three EEG channels, two EOGs, submentalis muscle EMG, bilateral anterior tibial EMG, respiratory signals, and ECG. In total, 16 of the subjects were healthy, and 92 were pathological. Table 2 shows the neurological status and number of subjects [44]. The age range of the subjects is 14–82, and the average age is 45. In total, 61% of the subjects were men (66 people), and 38% were women (42 people).

Table 2. Neurological status and number of subjects.

The Neurological Status	F	M	Age: Min–Max (Average)	Number of Patients
No pathology (controls/normal)	9	7	23–42 (32.18)	16
Nocturnal frontal lobe epilepsy (NFLE)	19	21	14–67 (30.27)	40
REM behavior disorder (RBD)	3	19	58–82 (70.72)	22
Periodic leg movements (PLM)	3	7	40–62 (55.10)	10
Insomnia	5	4	47–82 (60.88)	9
Narcolepsy	3	2	18–44 (31.60)	5
Sleep-disordered breathing (SDB)	-	4	65–78(71.25)	4
Bruxism	-	2	23–34 (28.50)	2
Total number of pathologies	**33**	**59**	**14–82 (49.19)**	**92**

F: female, M: male.

The CAP Sleep Database has been downloaded from Physionet [48]. Expert neurologists labeled these sleep data according to Rechtschaffen & Kales (R&K) rules using the sleep stage (W = waking, S1–S4 = sleep stages, R = REM, MT = body movements), time, duration, signal type data in the tag files. Each label classifies a unique (non-overlapping) 30-s data window. Data start time, hypnogram start time, and frequency information are needed for labeling. This information was obtained from files with EDF extensions. Using the Matlab 2019b program, the .edf files were read, and all recorded channels were listed. Of these channels, only the F4-C4 channels were used. Due to the absence of F4-C4 (these channels are commonly used EEG channels [43,49,50]. Therefore, we used these channels.) channels, some of the normal recordings were ignored.

2.2. Method

This research presents a new, handcrafted feature-based EEG signal classification model. Feature creation, feature selection, and classification are the main phases of the presented model. The feature creation step incorporates both textural and statistical methods. Maximum pooling was used to create decomposed signals. By using the created decomposed signals, features have been extracted at both low and high levels. Specifically, we used absolute pooling, average pooling, and maximum absolute pooling. In the feature selection phase, a three leveled selector (TSRFINCA) was employed. In the classification phase, CSVM was deployed as a classifier. The general steps of this model are given below.

Step 0: Load EEG signals.

Step 1: Apply absolute average pooling, average pooling, and absolute maximum pooling to obtain M1 (absolute average pooling), M2 (average pooling), and M3 (absolute maximum pooling) signals. Herein, we used non-overlapping blocks with a length of two to create decompressed signals. In the M1 function, the absolute value of the used non-overlapping block was used as the decompressed signal. In M2 and M3, we used the maximum of the absolute block values and average values of the used non-overlapping blocks for decomposition. Equations (3)–(8) provide a mathematical definition of these functions.

Step 2: Extract 512 textural features from each signal (raw EEG signal and the generated M1, M2, and M3 signals). In this step, $4 \times 512 = 2048$ features have been generated.

Step 3: Generate 36 statistical features from each signal and textural features by using 18 statistical moments. The used 18 statistical moments have been applied to the raw signal and the generated textural features in Step 2.

Two main feature extraction methodologies, namely, e-textural and statistical feature extraction, were used for handcrafted feature extraction. By deploying our proposed L-tetrolet pattern, textural features were generated. Statistical features were extracted using statistical moments to enforce our feature generation phase.

Step 4: Apply maximum pooling to the EEG signal and update signal. This step defines the decomposition level.

Step 5: Repeat Steps 1–4 five times. Herein, a multilevel feature generator is created. By using handcrafted feature extractors, only low-level features have been generated. To create high-level features, a multilevel feature extraction model was created. Equation (1) provides a mathematical definition of the maximum pooling operator.

$$D = MaxP(EEG) \tag{1}$$

$$D(j) = max([EEG(i)\ EEG(i+1)]),\ j \in \left\{1, 2, \ldots, \left|\frac{L}{2}\right|\right\}, i \in \{1, 3, \ldots, L-1\} \tag{2}$$

Herein, *MaxP*(.) defines the maximum pooling function, *D* is decomposed signal, *L* is the length of the used EEG signal (*EEG*), and *max*(.) is maximum value finding function.

Step 6: Fuse the generated features.

Step 7: Summarize each feature individually.

Step 8: Determine the threshold point to eliminate redundant features.

Step 9: Apply ReliefF [51] to features and generate a weight for each feature.

Step 10: Choose positive ReliefF [51] weighted features.

Step 11: Apply INCA [52] to the positive weighted feature by selecting ReliefF in Step 10.

Step 12: Forward the selected features to the classifier.

The twelve steps detailed above define the proposed decision support model. Steps 1–6 represent the L-tetrolet feature generation. Steps 8–11 denote TSRFINCA feature selection, and Step 12 demonstrates the classification phase. Figure 1 shows the proposed L-tetrolet pattern-based sleep stage classification model flow diagram. The next sections introduce the individual model phases in detail.
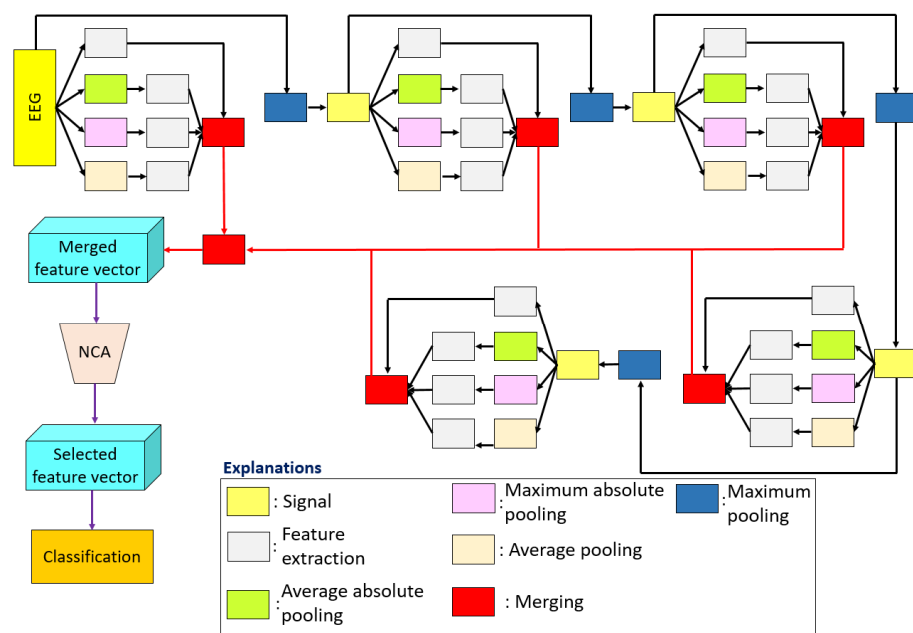

Figure 1. Snapshot of the proposed L-tetrolet and TSRFINCA based sleep stage classification model.

2.2.1. L-Tetrolet Pattern and Statistical Features Based Multileveled Feature Generation Method

Feature generation/extraction is the first phase of the proposed decision support method. Statistical and textural features were generated in this phase. Linear and nonlinear statistical moments were used to generate statistical features, and 18 statistical features were generated by using these moments. In the textural feature generation phase, we present a new microstructure that was inspired by the Tetris game. The letter 'L' (L-tetrolet) of the Tetris game was employed for pattern identification [53,54]. Therefore, the presented textural feature generation function is called an L-tetrolet pattern. The L-tetrolet pattern generates 512 features from a one-dimensional signal. Statistical features were also extracted from the generated textural features by deploying the 18 moments.

The primary objective of the presented feature generation model is to create low-level and high-level features. Therefore, a multileveled/multilayered method was employed to generate these features. A pooling-based decomposer was utilized as a decomposition method. By deploying four pooling functions – absolute average pooling, absolute maximum pooling, average pooling, and maximum pooling –, a five leveled feature generation method was created. The steps of the presented feature generation method are given below.

Step 1: Employ average, absolute maximum, and absolute average pooling to decompose the raw EEG signal into M1, M2, and M3. Here, 2 size non-overlapping blocks were used.

$$M1 = avp(EEG) \tag{3}$$

$$M2 = avp^{ab}(EEG) \tag{4}$$

$$M3 = max^{ab}(EEG) \tag{5}$$

$$avp(EEG) = M1(j) = \frac{EEG(i) + EEG(i+1)}{2}, \ i = \{1, 3, \ldots, Ln-1\}, j = \left\{1, 2, \ldots, \left|\frac{Ln}{2}\right|\right\} \tag{6}$$

$$avp^{ab}(EEG) = M2(j) = \frac{|EEG(i)| + |EEG(i+1)|}{2}, i = \{1, 3, \ldots, Ln-1\} \tag{7}$$

$$max^{ab}(EEG) = M3(j) = \begin{cases} |EEG(i)|, \ |EEG(i)| \geq |EEG(i+1)| \\ |EEG(i+1)|, \ |EEG(i)| < |EEG(i+1)| \end{cases}, \tag{8}$$
$$i = \{1, 3, \ldots, Ln-1\}$$

where $avp(.), avp^{ab}(.), maxp^{ab}(.)$ define average pooling, absolute average pooling, and absolute maximum pooling. EEG denotes the one-dimensional measurement signal, Ln represents the signal length. $|.|$ is absolute function.

Step 2: Generate features from the generated $M1$, $M2$, $M3$, and the raw one-dimensional signal (EEG). In this step, both statistical moments and the presented L-tetrolet pattern were used.

$$f^{st} = st(EEG) \tag{9}$$

$$f^T = L - tetrolet(EEG) \tag{10}$$

$$f^{stT} = st(L - tetrolet(EEG)) \tag{11}$$

In these Equations (see Equations (7) and (8)) statistical feature generation function ($st(.)$) and L-tetrolet pattern ($L - tetrolet(.)$) are defined. f^{st} represents 18 statistical features, f^T is 512 textural features and f^{stT} is the statistical features of the generated textural features. Table 3 lists the statistical moments that were used for feature extraction [55].

Here, the used 12th, 16th, 17th, and 18th moments extract nonlinear statistical features.

The presented L-tetrolet pattern was used to extract textural features. The steps of this function are detailed as follows:

Table 3. The statistical moments used for the generation of statistical features.

Num	Equation	Num	Equation
1	$\frac{1}{Ln}\sum\limits_{j=1}^{Ln} EEG(j)$	10	$max(EEG) - median(EEG)$
2	$\sqrt{\dfrac{\sum_{i=1}^{Ln}(EEG(i) - \frac{1}{Len}\sum_{j=1}^{Len} EEG(j))}{Ln - 1}}$	11	$\frac{1}{Ln}\sum\limits_{j=1}^{Ln}|EEG(j)|$
3	$max(EEG)$	12	$-\sum\limits_{j=1}^{Ln} log(prb(EEG(j)))^2$
4	$min(EEG)$	13	$max(|EEG|) - min(|EEG|)$
5	$median(EEG)$	14	$min(|EEG|)$
6	$\frac{1}{Ln}\left(\sum\limits_{i=1}^{Ln}(EEG(i) - \frac{1}{Ln}\sum\limits_{j=1}^{Ln} EEG(j))\right)^2$	15	$\sqrt{\dfrac{\sum_{i=1}^{Ln}(|EEG(i)| - \frac{1}{Ln}\sum_{j=1}^{Ln}|EEG(j)|)}{Ln - 1}}$
7	$\frac{1}{Ln}\sum\limits_{j=1}^{Ln} EEG(j)^2$	16	$-\sum\limits_{j=1}^{Ln} prb(EEG(j)) * log(prb(S(j)))$
8	$\frac{1}{Ln}\sum\limits_{i=1}^{Ln}\left|EEG(i) - \frac{1}{Ln}\sum\limits_{j=1}^{Ln} EEG(j)\right|$	17	$\sum\limits_{j=1}^{Ln} EEG(j)^2$
9	$max(EEG) - min(EEG)$	18	$-\sum\limits_{j=1}^{Ln} prb(EEG(j))^2 * log(prb(EEG(j)))^2$

where $prb(.)$ defines probability.

Step 2.1: Divide the used one-dimensional signals into overlapping blocks/windows (*blk*) with a size of 16.

$$blk = EEG(i + t - 1), \; i = \{1, 2, \ldots, Ln - 15\}, \; t = \{1, 2, \ldots, 16\} \tag{12}$$

Step 2.2: Create a matrix (*mtr*) with a size of 4×4 using the constructed block.

$$mtr(k, l) = blk(t), \; k = \{1, 2, 3, 4\}, \; l = \{1, 2, 3, 4\} \tag{13}$$

Figure 2 depicts the resulting 4×4 matrix.

Figure 2. The 4×4 matrix that was created for applying the proposed L-tetrolet pattern.

Step 2.3: Use two L-tetrolet based patterns by employing the 4×4 sized matrix. Figure 3 shows the L-tetrolet patterns that were used for feature generation.

Step 2.4: Extract bits using P1, P2, and binary feature generation function $S(.,.)$. These patterns (P1 and P2) are separately applied to the generated matrix. For P1 and P2, the used a, b, c, and d values are given in Equation (14) according to Figures 2 and 3.
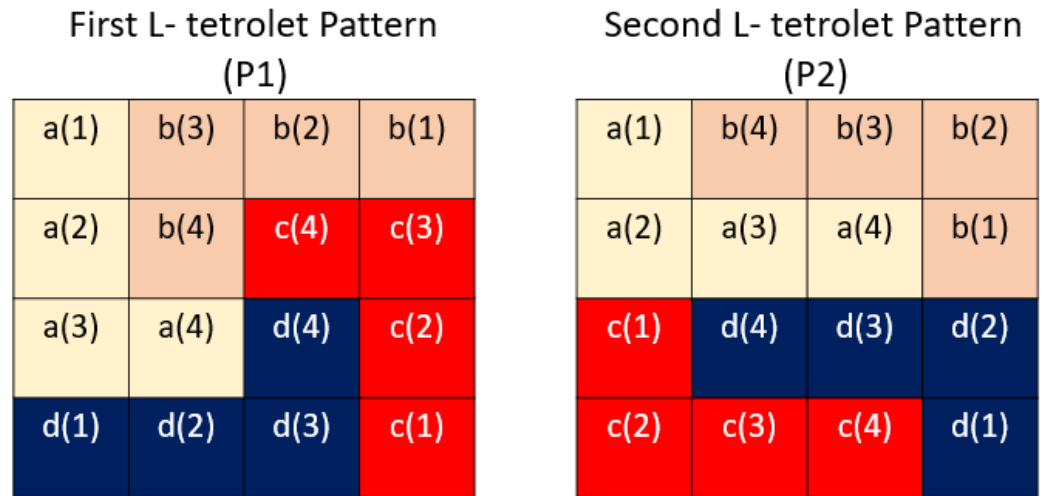
First L- tetrolet Pattern (P1)

a(1)	b(3)	b(2)	b(1)
a(2)	b(4)	c(4)	c(3)
a(3)	a(4)	d(4)	c(2)
d(1)	d(2)	d(3)	c(1)

Second L- tetrolet Pattern (P2)

a(1)	b(4)	b(3)	b(2)
a(2)	a(3)	a(4)	b(1)
c(1)	d(4)	d(3)	d(2)
c(2)	c(3)	c(4)	d(1)

Figure 3. The used L-tetrolet patterns. Each L-tetrolet is named using a letter (e.g., a, b, c, d), and these letters are shown using different colors. These patterns are called P1 and P2.

$$
\begin{vmatrix}
a^1(1) & a^2(1) \\
a^1(2) & a^2(2) \\
a^1(3) & a^2(3) \\
a^1(4) & a^2(4) \\
b^1(1) & b^2(1) \\
b^1(2) & b^2(2) \\
b^1(3) & b^2(3) \\
b^1(4) & b^2(4) \\
c^1(1) & c^2(1) \\
c^1(2) & c^2(2) \\
c^1(3) & c^2(3) \\
c^1(4) & c^2(4) \\
d^1(1) & d^2(1) \\
d^1(2) & d^2(2) \\
d^1(3) & d^2(3) \\
d^1(4) & d^2(4)
\end{vmatrix}
=
\begin{vmatrix}
V_1 & V_1 \\
V_5 & V_5 \\
V_9 & V_6 \\
V_{10} & V_7 \\
V_4 & V_8 \\
V_3 & V_4 \\
V_2 & V_3 \\
V_6 & V_2 \\
V_{16} & V_9 \\
V_{12} & V_{13} \\
V_8 & V_{14} \\
V_7 & V_{15} \\
V_{13} & V_{16} \\
V_{14} & V_{12} \\
V_{15} & V_{11} \\
V_{11} & V_{10}
\end{vmatrix}
\tag{14}
$$

Herein, a^1, b^1, c^1, d^1 are belonging to P1 pattern and a^2, b^2, c^2, d^2 are belonging to P2 pattern. By using these values, feature extraction process has been conducted. The bit generation phase has been given below.

$$
bit^t(k) = S\left(a^t(k), c^t(k)\right), \ k = \{1,2,3,4\}, \ t = \{1,2\}
\tag{15}
$$

$$
bit^t(k+4) = S\left(b^t(k), d^t(k)\right)
\tag{16}
$$

$$
S(par1, par2) = \begin{cases} 0, \ par1 - par2 < 0 \\ 1, \ par1 - par2 \geq 0 \end{cases}
\tag{17}
$$

where *par*1 and *par*2 are the first and second parameters of the binary feature generation (signum) function. Equations (15)–(17) were deployed to both P1 and P2, and eight bits were extracted from each pattern. The extracted bits are named bit^1 and bit^2 (they are shown using bit^t in Equations (15) and (16)). The length of each bit array is equal to eight. By deploying these bits, two novel signals were created for feature generation, and these signals were named the first map (*map*1) and the second map signal (*map*2), respectively. Binary to decimal conversion was used to create these signals, as shown in Equations (18) and (19).

Step 2.5: Create map signals employing the generated bits.

$$map1(i) = \sum_{k=1}^{8} bit^1(k) * 2^{k-1} \tag{18}$$

$$map2(i) = \sum_{k=1}^{8} bit^2(k) * 2^{k-1} \tag{19}$$

Step 2.6: Extract histograms of the $map1$ and $map2$ signals. Each histogram has $2^8 = 256$ values.

$$hist^1 = \delta(map1) \tag{20}$$

$$hist^2 = \delta(map2) \tag{21}$$

where $hist^1$ and $hist^2$ are histograms of the first and second map signals, respectively. $\delta(.)$ function is defined to extract histogram.

Step 2.7: Create a feature vector ($feat$) with a length of 512 by using $hist^1$ and $hist^2$.

$$feat(h) = hist^1(h), \ h = \{1, 2, \dots, 256\} \tag{22}$$

$$feat(h + 256) = hist^2(h) \tag{23}$$

Equations (19) and (20) define the feature concatenation process.

The given steps above (see Steps 2.1–2.7) are defined our proposed L-tetrolet pattern.

Step 3: Merge the generated textural, statistical, and statistical textural features of each signal. For a one-dimensional signal, 512 + 18 + 18 = 548 features were generated. In a level, the defined feature generation functions were applied to four signals (M1, M2, M3, and raw signal). Therefore, these functions generate 548 × 4 = 2192 features at each level.

Step 4: Decompose the one-dimensional signal (*EEG*) by deploying the maximum pooling decomposer. This step defines signal updating.

Step 5: Repeat Steps 1–4 five times utilizing decomposed signal input. This constitutes the multilevel feature extractor.

Step 6: Merge generated features in each level and obtain 2192 × 5 = 10,960 features from a one-dimensional signal.

2.2.2. Threshold Selection Based Relieff and Iterative Neighborhood Component Analysis

A three-layered feature selection model was used in this phase, and these layers were threshold-based feature selection, positive ReliefF weighted features selection, and INCA selection processes. The primary objectives of this feature selector were the following:

- Present an effective feature selector;
- Use advantages of the three feature selection methods together;
- Select the most appropriate features automatically.

Figure 4 shows a block diagram of the proposed TSRFINCA selector.

The following steps introduce the TSRFINCA functionality:

Step 1: Normalize the generated features (X) individually.

$$X^{norm}(:, i) = \frac{X(:, i) - min(X^{norm}(:, i))}{max(X^{norm}(:, i)) - min(X^{norm}(:, i))}, \ i = \{1, 2, \dots, 10960\} \tag{24}$$

where X^{norm} represents normalized features by deploying min-max normalization.

Step 2: Deploy threshold-based feature selection. In this study, we used zero as threshold (β). The mathematical descriptions of this method are given below.

$$tpl(j) = \sum_{d=1}^{D} X^{norm}(d, j), j = \{1, 2, \dots, 10960\} \tag{25}$$

$$X^1(:, cnt) = X^{norm}(:, j), \ cnt = cnt + 1, \ if \ tpl(j) > \beta \qquad (26)$$

where tpl means summarization of the features, X^1 is the selected features in the first layer, and cnt is a counter.

Step 3: Employ ReliefF to X^1 and generate ReliefF weights (w^{RF}).

Step 4: Eliminate negative weighted features to obtain second layer features (X^2).

$$X^2(:, cnt) = X^1(:, j), \ cnt = cnt + 1, \ if \ w^{RF}(j) > 0 \qquad (27)$$

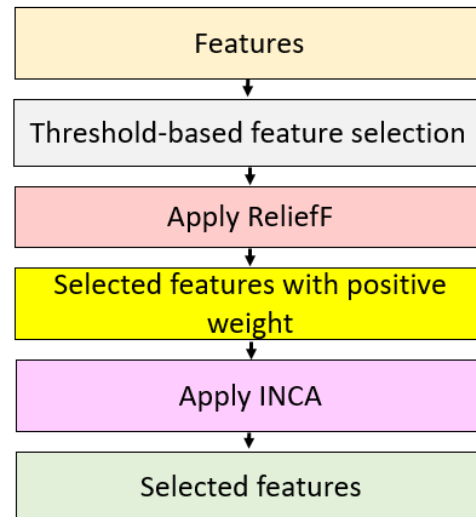Step 5: Apply INCA to X^2 and obtain the final features (X^3).

Figure 4. Block diagram of the TSRFINCA model.

INCA is an iterative selector, and it can select features of various sizes and hence it is applicable to a wide range of problem solutions. In this work, we progress now to the classification algorithm that was used for sleep stage detection.

2.2.3. Classification

Classification is the last phase of the presented sleep stage classification model. Here, we used a CSVM classification algorithm. The hyper-parameters of this classifier are given below as follows:

- Training and testing method: 10-fold cross-validation;
- Kernel: Third-degree polynomial order (Cubic);
- Box constraint level (C value): One;
- Multiclass method: One-vs-one.

3. Results

3.1. Experimental Setup

The CAP dataset was downloaded from Physionet to train and test the presented L-tetrolet pattern and TSRFINCA-based sleep stage classification model. This research focused on the sleep stages of insomniacs and normal subjects. The sleep stage datasets are generally heterogeneous. Therefore, high classification rates do not reflect the model performance. A balanced EEG dataset has been created to overcome this problem by randomly selecting EEG signals from each subject. By creating these three datasets, the following three cases were defined, and these are explained below:

Case 1: This dataset was collected from the insomnia subjects. It includes the following six classes: wake, stage 1, stage 2, stage 3, stage 4, and REM. This dataset contains 1356 EEG signals (each class has 226 EEG signals). F4-C4 channels have been used in this case.

Case 2: This case uses EEG signals from normal subjects. A homogenous dataset was created in this case. There are 1698 EEG signals in this dataset (each class has 283 EEG signals). F4-C4 channels have been used in this case.

Case 3: In this case, a merged dataset is used. This dataset was created by merging datasets of Cases 1–2. Therefore, it contains 3054 EEG signals (each class has 283 + 226 = 509 EEG signals). F4-C4 channels have been used in this case.

These three balanced datasets were used to define three distinct sleep stage identification tasks. The MATLAB (2020a) programming environment was used to calculate test results and implement the proposed decision support model. The used functions were named main, L-tetrolet pattern, statistical feature generator, TSRFINCA, and classification. In the main function, the EEG signals were read, and other functions were called in the main function to classify sleep stages. The proposed model was implemented on a basic desktop computer, and parallel programming or hardware acceleration was not used.

3.2. Results

The model quality was obtained by assessing the classification results according to the rules of 10-fold cross-validation. Here, six classification results were presented. Accuracy, F1-score, average precision, and geometric mean results were calculated. Table 4 lists the calculated results for each case.

Table 4. The calculated performance results of the presented L-tetrolet pattern and TSRFINCA model.

Case	Accuracy	F1-Score	Average Precision	Geometric Mean	Sensitivity	Specificity
Case 1	95.43%	95.42%	95.46%	95.36%	90.27	98.94
Case 2	91.05%	90.01%	90.08%	89.95%	86.22	97.17
Case 3	92.31%	92.29%	92.29%	92.23%	87.03	97.96

As can be seen from Table 4, the recommended method yielded 95.43%, 91.05%, and 92.31% classification accuracies for Case 1, Case 2, and Case 3, respectively. A 10-fold cross-validation was used to calculate these results. Table 5 details the fold-by-fold results.

Table 5. Fold-by-fold accuracies in % for the three cases.

Fold	Case 1	Case 2	Case 3
Fold-1	86.03	80.59	84.26
Fold-2	97.06	94.12	92.46
Fold-3	100.0	98.24	95.74
Fold-4	88.97	93.53	93.77
Fold-5	97.06	85.29	87.87
Fold-6	94.85	95.88	95.08
Fold-7	97.06	86.47	91.80
Fold-8	100.0	95.88	98.36
Fold-9	98.53	90.00	91.48
Fold-10	94.70	90.48	92.23
Overall	95.43	91.05	92.31

3.3. Computational Complexity Analysis

Computational complexity is a crucial property that determines the practicality of the proposed model. A lower computational complexity is more resource-efficient, which translates into less energy usage and lower cost. The presented model consists of three al-

gorithms. Therefore, the time complexities of these algorithms should be calculated [56,57]. Table 6 introduces these calculations in detail.

Table 6. The time complexity calculation of the presented model.

Phase	Steps	Computational Complexity
Feature generation	Pooling-based decomposition	$O(ndlognd)$
	Statistical feature generation	$O(ndlognd)$
	Textural feature generation (L-tetrolet pattern)	$O(ndlognd)$
	Statistical features extraction of the textural features	$O(ndlognd)$
TSRFINCA	Threshold feature selection	$O(kd)$
	ReliefF-based selection	$O(kd)$
	INCA	$O(Ik^3d)$
Classification	SVM	$O(k^3d)$
Total	$O(4ndlognd + 2kd + Ik^3d + k^3d) \cong O(ndlognd + Ik^3d)$	

In this table (see Table 6), the used coefficients are given as follows. n is the length of the signal, d defines the number of observations, k represents the time complexity coefficient of the used feature selection and classification models, and I defines the number of iterations for iterative feature selection.

Feature generation: We have used a multileveled feature generation in this study. In each level, a maximum pooling decomposer was used to halve the signal length. The used feature generation functions (L-tetrolet pattern and statistical feature generator) have low computational complexity ($O(n)$). Therefore, the time complexity of this phase is calculated as $O(ndlognd)$. Here, n is the size of the EEG signal, and d represents the number of EEG signals.

Feature selection: The TSRFINCA algorithm has three layers. The threshold-based feature selection model is a simple and basic model. Therefore, the time complexity of it is calculated as $O(kd)$. Here, k is defined as the number of features. In this phase, INCA is the most complex feature selector and $O(Ik^3d)$ is found as the computational complexity. Here, I is the number of iterations because it is an iterative feature selector, and in each iteration, the loss value is calculated using the SVM classifier.

Classification: A CSVM classifier, with a time complexity of $O(k^3d)$, was employed for classification.

As can be seen from the time complexity analysis (see Table 6), the proposed model has a low time burden. The deep learning models have an exponential time burden, but this model has a linear time burden. Therefore, there is no need to use extra hardware to implement our proposal. Furthermore, this model can extract features at low and high levels.

4. Discussion

As stated in Sect. 3, the presented model has the following three fundamental phases: feature generation, TSRFINCA-based feature selection, and classification. The presented model uses four pooling methods to overcome the routing problem of pooling. For instance, the maximum pooling only routes peak values. We proposed a multiple pooling-based decomposition model to overcome the routing problem. Both textural and statistical feature generators have been utilized for feature extraction to create handcrafted features. By using these feature extractors and the proposed multiple pooling function, a multileveled feature extraction method has been presented to generate features at both low and high levels. The presented three-layered feature selection function—TSRFINCA—selected the top informative features from the used datasets. In this research, we have used three datasets. The presented TSRFINCA selects a variable-sized feature vector for each dataset. The sizes of the optimal feature sets were found to be 644, 711, and 188 for Case 1, Case 2, and Case 3, respectively. Figure 5 documents the feature selection process.
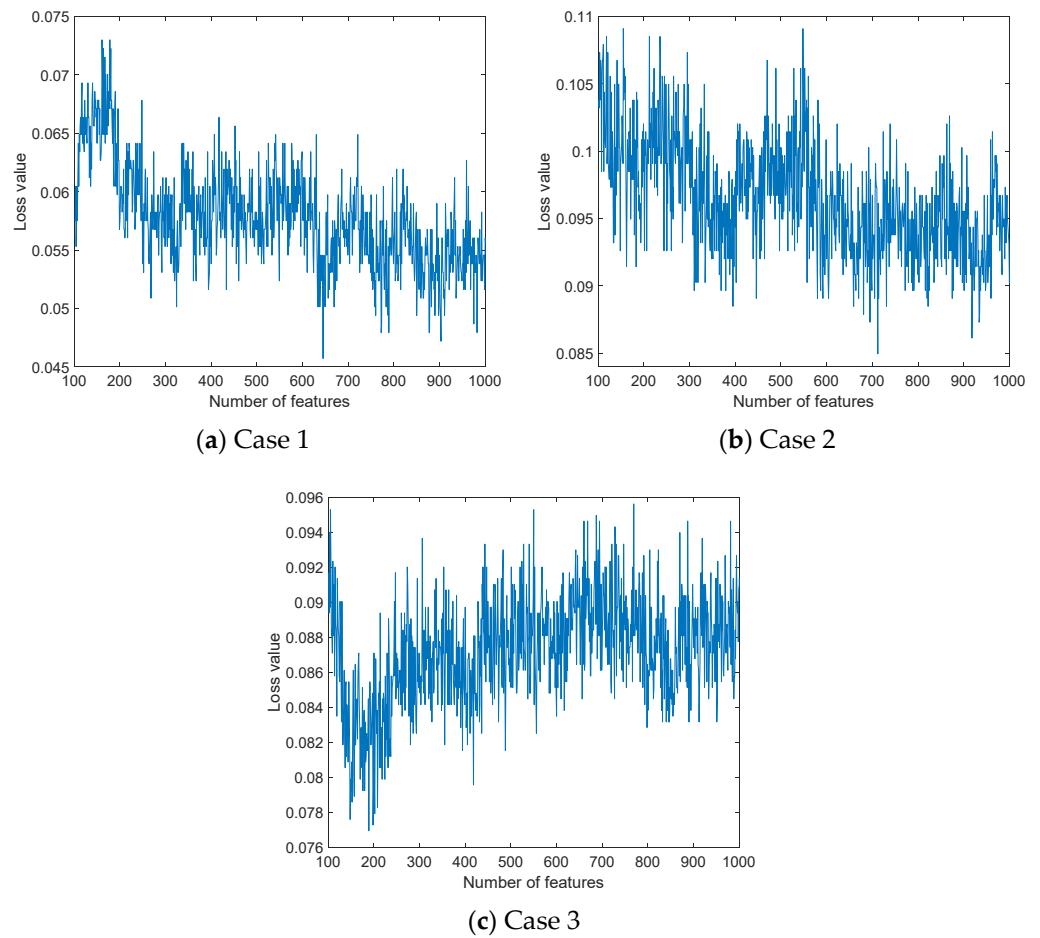
(**a**) Case 1

(**b**) Case 2

(**c**) Case 3

Figure 5. Feature selection processes of the cases.

In this figure (see Figure 5), a number of features and loss values are demonstrated. The proposed feature selector is an iterative feature selector and it calculates loss values of the 901 (initial value and end value of the loop are 100 and 1000, respectively; thus, $901 = 1000 - 100 + 1$ feature vectors have been evaluated for each dataset) feature vectors. The optimal feature vectors have been selected using minimum loss values. These optimal features were forwarded to the CSVM classifier. This classifier was utilized as both a loss value generator (calculating misclassification rates of the chosen 901 feature vectors) and a classifier. Figure 6 shows a confusion matrix for each case.
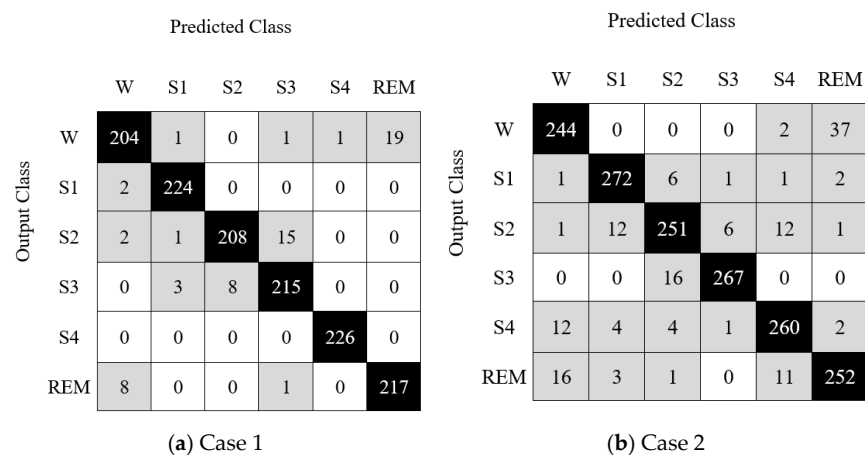
Predicted Class

		W	S1	S2	S3	S4	REM
Output Class	W	204	1	0	1	1	19
	S1	2	224	0	0	0	0
	S2	2	1	208	15	0	0
	S3	0	3	8	215	0	0
	S4	0	0	0	0	226	0
	REM	8	0	0	1	0	217

(**a**) Case 1

Predicted Class

		W	S1	S2	S3	S4	REM
Output Class	W	244	0	0	0	2	37
	S1	1	272	6	1	1	2
	S2	1	12	251	6	12	1
	S3	0	0	16	267	0	0
	S4	12	4	4	1	260	2
	REM	16	3	1	0	11	252

(**b**) Case 2

Figure 6. *Cont.*

Predicted Class

	W	S1	S2	S3	S4	REM
W	443	2	0	3	4	57
S1	4	489	11	1	2	2
S2	5	10	455	26	13	0
S3	0	1	27	480	0	1
S4	11	4	1	0	493	0
REM	32	6	2	0	10	459

Output Class

(**c**) Case 3

Figure 6. Confusion matrix for each case.

The confusion matrices in Figure 6 denote the case-specific results.

To select the optimal classifiers, features of Case 3 were tested on the shallow variable classifiers. These were decision tree (DT) [58], linear discriminant (LD) [59], Naïve Bayes (NB) [60], linear SVM (LSVM) [61], CSVM [62], quadratic SVM (QSVM) [62], k nearest neighbors (kNN) [63] and bagged tree (BT) [64]. Figure 7 introduces the accuracies achieved with the individual classifiers.
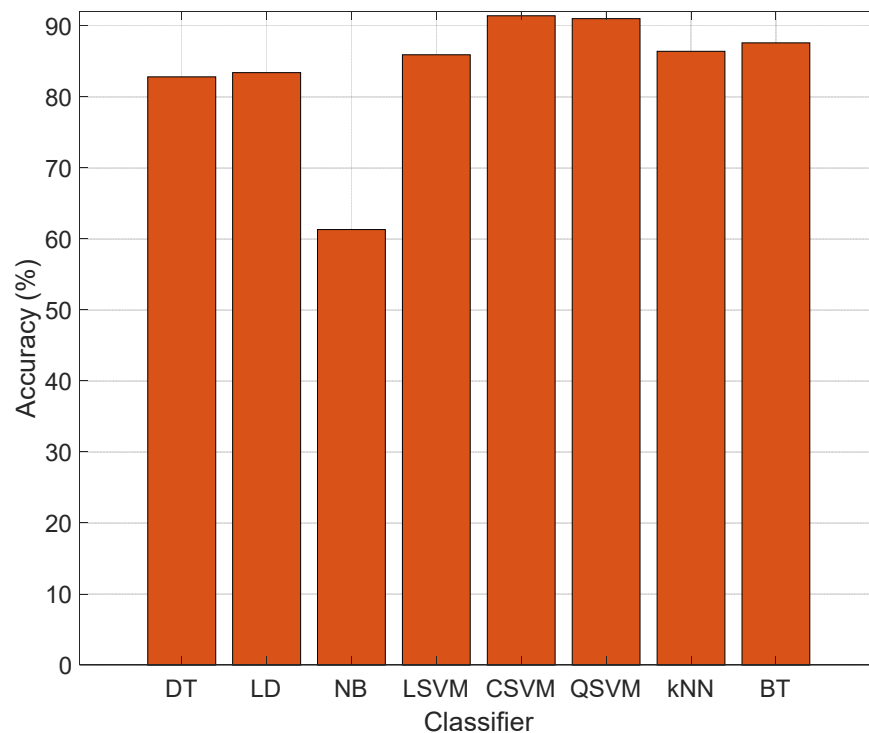

Figure 7. Classification accuracies of the classifier. Here, the presented L-tetrolet and maximum pooling-based feature generation method is applied to Case 3. The first and second layers of the TSRFINCA are applied to these features to eliminate the redundant feature, and NCA selected 1000 features for tests.

Figure 6 demonstrates that the best classifier is CSVM. Therefore, CSVM is selected as both an error generator and a classifier.

We have compared our model with other sleep stage classification methods. Table 7 lists the comparison results.

Table 7. The comparison results.

Study	Dataset	Accuracy Result (%)
Bajaj and Pachori [65]	Sleep-EDF dataset [24,66]	88.47 (Pz-Oz)
Hassan et al. [67]	Sleep-EDF database [24,66]	90.69 (Pz-Oz)
Jiang et al. [68]	1. Sleep-EDF database [24,66] 2. Sleep-EDF Expanded database [24]	89.40 (Fpz-Cz) 88.30 (Pz-Oz)
Kanwal et al. [69]	Sleep-EDF database [24,66]	93.00 (Pz-Oz, PFz-Cz, EOG)
Basha et al. [70]	Sleep-EDF database [24,66]	90.20 (PFz-Cz)
Jadhav et al. [71]	Sleep-EDF Expanded database [24]	85.07 (PFz-Cz) 82.92 (Pz-Oz)
Michielli et al. [72]	Sleep-EDF database [24,66]	90.80 (Pz-Oz)
Huang et al. [73]	Sleep-EDF Expanded database [24]	84.60 (Fpz-Cz) 82.30 (Pz-Oz)
Kim et al. [74]	CAP Sleep Database on PhysioNet [24]	73.60 (unspecified)
Shanin et al. [75]	Collected data	92.00 (C3-C4)
Karimzadeh et al. [76]	Sleep-EDF dataset [24,66]	88.97 (Pz-Oz)
Seifpour et al. [77]	Sleep-EDF dataset [24,66]	90.60 (Fpz-Cz) 88.60 (Pz-Oz)
Sharma et al. [3]	Sleep-EDF dataset [24,66]	91.50 (Pz-Oz)
Zhou et al. [78]	1. Sleep-EDF database [24,66] 2. Sleep-EDF Expanded database [24]	1. 91.80 (Fpz-Cz) 2. 85.30 (Pz-Oz)
Zhang et al. [79]	1. UCD dataset [24] 2. MIT-BIH polysomnographic database [24]	1. 88.40 (C3-A2 + C4-A1) 2. 87.60 (C3-A2 + C4-A1)
Liu et al. [80]	Sleep-EDF Expanded database [24]	84.44 (Fpz-Cz + Pz-Oz)
Cai et al. method [81]	Sleep-EDF database [24,66]	87.21 (Fpz-Cz)
Loh et al. [82]	CAP Sleep Database [24,44]	90.46 (C4-A1/C3-A2)
Sharma et al. [49]	CAP Sleep Database [24,44]	85.10 (F4-C4 + C4-A1)
Dhok et al. [83]	CAP Sleep Database [24,44]	87.45 (C4-C1/C3-A2)
Sharma et al. [84]	CAP Sleep Database [24,44]	83.30 (C4-A1 + F4-C4)
The proposed method	CAP Sleep Database on PhysioNet [24]	Case1: 95.43 (F4-C4) Case2: 91.05 (F4-C4) Case3: 92.31 (F4-C4)

Table 7 shows the success of the presented L-tetrolet pattern and TSRFINCA-based model. Moreover, prior presented models generally used a single dataset, but we tested our model on three balanced/homogenous datasets. Our proposal attained over 90% classification accuracy for all cases. These findings clearly demonstrate our success. The advantages of this model are the following:

- A new game-inspired feature generation model is presented, and the effectiveness of this approach is established through EEG-based sleep stage classification;
- To overcome the routing problem of the pooling method, a multiple pooling decomposer-based feature generation strategy was used;
- A three-layered feature selector is presented;
- By applying these methods and CSVM, a highly accurate sleep stage classification model is presented;
- The recommended model outperformed;
- The proposed model can be applied to a computer with basic system configurations.

    The drawbacks of this research are the following:

- The presented TSRFINCA is a hybrid and iterative feature selector, but the computational complexity is high. Moreover, we have used a shallow classifier. In this work, deep classifiers can be used to increase the classification ability, or a metaheuristic optimization model can be used to tune the hyperparameters of the used classifier;
- The datasets used are small. Therefore, when we used one dataset for training and the other datasets for testing, we achieved a classification accuracy of about 50%. Since these EEG signals have sick subjects (each case defines a disorder).

Diagnoses of sleep or sleep-related diseases are time-consuming because the diagnostic pathway relies on manual signal analysis. A new EEG-based sleep stage detection/monitoring system can be developed soon to help medical professionals with diagnosis. Figure 8 denotes the intended intelligent monitoring system.
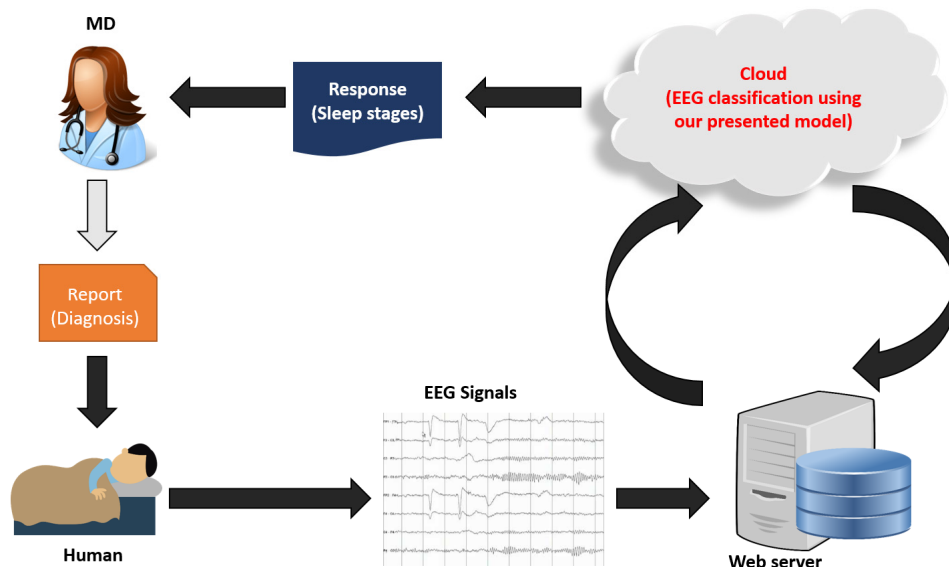

Figure 8. The intended automated sleep stage classification and monitoring model.

This research presents a new game-based feature generation function. The L-tetrolet pattern is inspired by the Tetris game. Other game-based feature generation or decomposition models can be presented in future studies, and the recommended model can be applied to other one-dimensional signals to solve classification problems. In the future, we plan to develop a game-based deep learning model for one-dimensional signal classification, which might replace or augment recurrent neural networks.

5. Conclusions

In this research, we propose a new feature engineering model. The essential goal of that model is to extract the most significant features from EEG signals. The model is based on a new game-based feature extraction function, named L-tetrolet, which extracts textural feature information. To generate high-level features, a multileveled feature extraction structure is presented using a combination of four pooling techniques. This approach fuses hybrid approximation and the advantages of pooling techniques. In the feature selection phase, a three-layered hybrid feature selector has been used, and the selected features have been classified using a shallow classifier. Using PhysioNet, three different sleep EEG datasets were created, each containing six groups. Our proposed L-tetrolet-based model attained >90% overall classification accuracy on these datasets. Moreover, our proposal reached 95.43% classification accuracy in Case 1. These results were compared to other recent models, showing that our model outperforms all the previous methods used for sleep stage detection based on signals from the CAP database. These findings demonstrated that our model achieved satisfactory classification performance and time complexity for solving sleep stage classification problems using EEG signals.

In the future, we plan to accomplish the following:

-   Propose new game-based feature extraction functions;
-   Purpose self-organized feature engineering models;
-   Propose a new generation of pooling/decomposition methods by using quantum computing and superposition;
-   Develop a new sleep stage classification application, which will be used in medical centers.

References

1. Santaji, S.; Desai, V. Analysis of EEG Signal to Classify Sleep Stages Using Machine Learning. *Sleep Vigil.* **2020**, *4*, 145–152. [CrossRef]
2. Taran, S.; Sharma, P.C.; Bajaj, V. Automatic sleep stages classification using optimize flexible analytic wavelet transform. *Knowl.-Based Syst.* **2020**, *192*, 105367. [CrossRef]
3. Sharma, M.; Goyal, D.; Achuth, P.; Acharya, U.R. An accurate sleep stages classification system using a new class of optimally time-frequency localized three-band wavelet filter bank. *Comput. Biol. Med.* **2018**, *98*, 58–75. [CrossRef]
4. Urtnasan, E.; Park, J.-U.; Joo, E.Y.; Lee, K.-J. Deep Convolutional Recurrent Model for Automatic Scoring Sleep Stages Based on Single-Lead ECG Signal. *Diagnostics* **2022**, *12*, 1235. [CrossRef]
5. Ahmadi, A.; Bazregarzadeh, H.; Kazemi, K. Automated detection of driver fatigue from electroencephalography through wavelet-based connectivity. *Biocybern. Biomed. Eng.* **2021**, *41*, 316–332. [CrossRef]
6. Xu, S.; Faust, O.; Silvia, S.; Chakraborty, S.; Barua, P.D.; Loh, H.W.; Elphick, H.; Molinari, F.; Acharya, U.R. A review of automated sleep disorder detection. *Comput. Biol. Med.* **2022**, *150*, 106100. [CrossRef] [PubMed]
7. Cai, Q.; An, J.; Gao, Z. A multiplex visibility graph motif-based convolutional neural network for characterizing sleep stages using EEG signals. *Brain Sci. Adv.* **2020**, *6*, 355–363. [CrossRef]
8. Aboalayon, K.A.; Ocbagabir, H.T.; Faezipour, M. Efficient sleep stage classification based on EEG signals. In Proceedings of the IEEE Long Island Systems, Applications and Technology (LISAT) Conference, Farmingdale, NY, USA, 2 May 2014; pp. 1–6.
9. Hassan, A.R.; Subasi, A. A decision support system for automated identification of sleep stages from single-channel EEG signals. *Knowl.-Based Syst.* **2017**, *128*, 115–124. [CrossRef]
10. Malhotra, R.K.; Avidan, A.Y. Sleep stages and scoring technique. In *Atlas of Sleep Medicine*; Elsevier: Amsterdam, The Netherlands, 2013; pp. 77–99; ISBN 9781455712687.
11. Berry, R.B.; Brooks, R.; Gamaldo, C.E.; Harding, S.M.; Marcus, C.; Vaughn, B.V. The AASM manual for the scoring of sleep and associated events. *Rules Terminol. Tech. Specif. Darien Ill. Am. Acad. Sleep Med.* **2012**, *176*, 2012.
12. Moser, D.; Anderer, P.; Gruber, G.; Parapatics, S.; Loretz, E.; Boeck, M.; Kloesch, G.; Heller, E.; Schmidt, A.; Danker-Hopfe, H. Sleep classification according to AASM and Rechtschaffen & Kales: Effects on sleep scoring parameters. *Sleep* **2009**, *32*, 139–149. [PubMed]
13. Fraiwan, L.; Lweesy, K.; Khasawneh, N.; Wenz, H.; Dickhaus, H. Automated sleep stage identification system based on time–frequency analysis of a single EEG channel and random forest classifier. *Comput. Methods Programs Biomed.* **2012**, *108*, 10–19. [CrossRef]
14. Loh, H.W.; Ooi, C.P.; Seoni, S.; Barua, P.D.; Molinari, F.; Acharya, U.R. Application of Explainable Artificial Intelligence for Healthcare: A Systematic Review of the Last Decade (2011–2022). *Comput. Methods Programs Biomed.* **2022**, *226*, 107161. [CrossRef]
15. Ebrahimi, F.; Mikaeili, M.; Estrada, E.; Nazeran, H. Automatic sleep stage classification based on EEG signals by using neural networks and wavelet packet coefficients. In Proceedings of the 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vancouver, BC, Canada, 20–25 August 2008; pp. 1151–1154.
16. Tzimourta, K.D.; Tsilimbaris, A.; Tzioukalia, K.; Tzallas, A.T.; Tsipouras, M.G.; Astrakas, L.G.; Giannakeas, N. EEG-based automatic sleep stage classification. *Biomed. J.* **2018**, *1*, 6.
17. Sun, C.; Fan, J.; Chen, C.; Li, W.; Chen, W. A two-stage neural network for sleep stage classification based on feature learning, sequence learning, and data augmentation. *IEEE Access* **2019**, *7*, 109386–109397. [CrossRef]
18. Alickovic, E.; Subasi, A. Ensemble SVM method for automatic sleep stage classification. *IEEE Trans. Instrum. Meas.* **2018**, *67*, 1258–1265. [CrossRef]
19. Faust, O.; Razaghi, H.; Barika, R.; Ciaccio, E.J.; Acharya, U.R. A review of automated sleep stage scoring based on physiological signals for the new millennia. *Comput. Methods Programs Biomed.* **2019**, *176*, 81–91. [CrossRef] [PubMed]

20. Acharya, U.R.; Bhat, S.; Faust, O.; Adeli, H.; Chua, E.C.-P.; Lim, W.J.E.; Koh, J.E.W. Nonlinear dynamics measures for automated EEG-based sleep stage detection. *Eur. Neurol.* **2015**, *74*, 268–287. [CrossRef] [PubMed]

21. Baygin, M.; Yaman, O.; Tuncer, T.; Dogan, S.; Barua, P.D.; Acharya, U.R. Automated accurate schizophrenia detection system using Collatz pattern technique with EEG signals. *Biomed. Signal Process. Control* **2021**, *70*, 102936. [CrossRef]

22. Barua, P.D.; Dogan, S.; Tuncer, T.; Baygin, M.; Acharya, U.R. Novel automated PD detection system using aspirin pattern with EEG signals. *Comput. Biol. Med.* **2021**, *137*, 104841. [CrossRef]

23. Kobat, M.A.; Kivrak, T.; Barua, P.D.; Tuncer, T.; Dogan, S.; Tan, R.-S.; Ciaccio, E.J.; Acharya, U.R. Automated COVID-19 and Heart Failure Detection Using DNA Pattern Technique with Cough Sounds. *Diagnostics* **2021**, *11*, 1962. [CrossRef] [PubMed]

24. 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. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation* **2000**, *101*, e215–e220. [CrossRef] [PubMed]

25. Mariani, S.; Bianchi, A.M.; Manfredini, E.; Rosso, V.; Mendez, M.O.; Parrino, L.; Matteucci, M.; Grassi, A.; Cerutti, S.; Terzano, M.G. Automatic detection of A phases of the Cyclic Alternating Pattern during sleep. In Proceedings of the 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology, Buenos Aires, Argentina, 31 August–4 September 2010; pp. 5085–5088.

26. Mariani, S.; Grassi, A.; Mendez, M.O.; Milioli, G.; Parrino, L.; Terzano, M.G.; Bianchi, A.M. EEG segmentation for improving automatic CAP detection. *Clin. Neurophysiol.* **2013**, *124*, 1815–1823. [CrossRef]

27. Machado, F.; Teixeira, C.; Santos, C.; Bento, C.; Sales, F.; Dourado, A. A-phases subtype detection using different classification methods. In Proceedings of the 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Orlando, FL, USA, 16–20 August 2016; pp. 1026–1029.

28. Mostafa, S.S.; Mendonça, F.; Ravelo-García, A.; Morgado-Dias, F. Combination of deep and shallow networks for cyclic alternating patterns detection. In Proceedings of the 2018 13th APCA International Conference on Automatic Control and Soft Computing (CONTROLO), Ponta Delgada, Portugal, 4–6 June 2018; pp. 98–103.

29. Hartmann, S.; Baumert, M. Automatic a-phase detection of cyclic alternating patterns in sleep using dynamic temporal information. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2019**, *27*, 1695–1703. [CrossRef]

30. Hartmann, S.; Baumert, M. Improved A-phase Detection of Cyclic Alternating Pattern Using Deep Learning. In Proceedings of the 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin, Germany, 23–27 July 2019; pp. 1842–1845.

31. Mendonça, F.; Mostafa, S.S.; Morgado-Dias, F.; Juliá-Serdá, G.; Ravelo-García, A.G. A Method for Sleep Quality Analysis Based on CNN Ensemble With Implementation in a Portable Wireless Device. *IEEE Access* **2020**, *8*, 158523–158537. [CrossRef]

32. Dimitriadis, S.I.; Salis, C.I.; Liparas, D. A Sleep Disorder Detection Model based on EEG Cross-Frequency Coupling and Random Forest. *medRxiv* **2020**, *18*. [CrossRef]

33. Arce-Santana, E.R.; Alba, A.; Mendez, M.O.; Arce-Guevara, V. A-phase classification using convolutional neural networks. *Med. Biol. Eng. Comput.* **2020**, *58*, 1003–1014. [CrossRef]

34. Abbasi, S.F.; Jamil, H.; Chen, W. EEG-based neonatal sleep stage classification using ensemble learning. *Comput. Mater. Contin* **2022**, *70*, 4619–4633.

35. Li, C.; Qi, Y.; Ding, X.; Zhao, J.; Sang, T.; Lee, M. A Deep Learning Method Approach for Sleep Stage Classification with EEG Spectrogram. *Int. J. Environ. Res. Public Health* **2022**, *19*, 6322. [CrossRef] [PubMed]

36. Zhang, G.-Q.; Cui, L.; Mueller, R.; Tao, S.; Kim, M.; Rueschman, M.; Mariani, S.; Mobley, D.; Redline, S. The National Sleep Research Resource: Towards a sleep data commons. *J. Am. Med. Inform. Assoc.* **2018**, *25*, 1351–1358. [CrossRef]

37. Zaidi, T.F.; Farooq, O. EEG Sub-bands based Sleep Stages Classification using Fourier Synchrosqueezed Transform Features. *Expert Syst. Appl.* **2022**, *212*, 118752. [CrossRef]

38. Sors, A.; Bonnet, S.; Mirek, S.; Vercueil, L.; Payen, J.-F. A convolutional neural network for sleep stage scoring from raw single-channel EEG. *Biomed. Signal Process. Control* **2018**, *42*, 107–114. [CrossRef]

39. Quan, S.F.; Howard, B.V.; Iber, C.; Kiley, J.P.; Nieto, F.J.; O'Connor, G.T.; Rapoport, D.M.; Redline, S.; Robbins, J.; Samet, J.M. The sleep heart health study: Design, rationale, and methods. *Sleep* **1997**, *20*, 1077–1085.

40. Goshtasbi, N.; Boostani, R.; Sanei, S. SleepFCN: A Fully Convolutional Deep Learning Framework for Sleep Stage Classification Using Single-Channel Electroencephalograms. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2022**, *30*, 2088–2096. [CrossRef] [PubMed]

41. Shahbakhti, M.; Beiramvand, M.; Eigirdas, T.; Solé-Casals, J.; Wierzchon, M.; Broniec-Wójcik, A.; Augustyniak, P.; Marozas, V. Discrimination of Wakefulness from Sleep Stage I Using Nonlinear Features of a Single Frontal EEG Channel. *IEEE Sens. J.* **2022**, *22*, 6975–6984. [CrossRef]

42. Devuyst, S.; Dutoit, T.; Kerkhofs, M. *The DREAMS Databases and Assessment Algorithm*; Zenodo: Geneva, Switzerland, 2005.

43. Zhao, C.; Li, J.; Guo, Y. SleepContextNet: A temporal context network for automatic sleep staging based single-channel EEG. *Comput. Methods Programs Biomed.* **2022**, *220*, 106806. [CrossRef]

44. Terzano, M.G.; Parrino, L.; Sherieri, A.; Chervin, R.; Chokroverty, S.; Guilleminault, C.; Hirshkowitz, M.; Mahowald, M.; Moldofsky, H.; Rosa, A. Atlas, rules, and recording techniques for the scoring of cyclic alternating pattern (CAP) in human sleep. *Sleep Med.* **2001**, *2*, 537–553. [CrossRef]

45. Eldele, E.; Chen, Z.; Liu, C.; Wu, M.; Kwoh, C.-K.; Li, X.; Guan, C. An attention-based deep learning approach for sleep stage classification with single-channel eeg. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2021**, *29*, 809–818. [CrossRef]

46. Yang, B.; Zhu, X.; Liu, Y.; Liu, H. A single-channel EEG based automatic sleep stage classification method leveraging deep one-dimensional convolutional neural network and hidden Markov model. *Biomed. Signal Process. Control* **2021**, *68*, 102581. [CrossRef]

47. Kemp, B.; Värri, A.; Rosa, A.C.; Nielsen, K.D.; Gade, J. A simple format for exchange of digitized polygraphic recordings. *Electroencephalogr. Clin. Neurophysiol.* **1992**, *82*, 391–393. [CrossRef]

48. Physionet. CAP Sleep Database. 2012. Available online: https://physionet.org/content/capslpdb/1.0.0 (accessed on 27 August 2020).

49. Sharma, M.; Tiwari, J.; Acharya, U.R. Automatic sleep-stage scoring in healthy and sleep disorder patients using optimal wavelet filter bank technique with EEG signals. *Int. J. Environ. Res. Public Health* **2021**, *18*, 3087. [CrossRef]

50. Lai, D.; Heyat, M.B.B.; Khan, F.I.; Zhang, Y. Prognosis of sleep bruxism using power spectral density approach applied on EEG signal of both EMG1-EMG2 and ECG1-ECG2 channels. *IEEE Access* **2019**, *7*, 82553–82562. [CrossRef]

51. Urbanowicz, R.J.; Meeker, M.; La Cava, W.; Olson, R.S.; Moore, J.H. Relief-based feature selection: Introduction and review. *J. Biomed. Inform.* **2018**, *85*, 189–203. [CrossRef] [PubMed]

52. Tuncer, T.; Dogan, S.; Özyurt, F.; Belhaouari, S.B.; Bensmail, H. Novel Multi Center and Threshold Ternary Pattern Based Method for Disease Detection Method Using Voice. *IEEE Access* **2020**, *8*, 84532–84540. [CrossRef]

53. Patsis, G.; Sahli, H.; Verhelst, W.; Troyer, O.D. Evaluation of attention levels in a tetris game using a brain computer interface. In Proceedings of the International Conference on User Modeling, Adaptation, and Personalization, Rome, Italy, 10–14 June 2013; pp. 127–138.

54. Krommweh, J. Tetrolet transform: A new adaptive Haar wavelet algorithm for sparse image representation. *J. Vis. Commun. Image Represent.* **2010**, *21*, 364–374. [CrossRef]

55. Kuncan, F.; Kaya, Y.; Kuncan, M. Sensör işaretlerinden cinsiyet tanıma için yerel ikili örüntüler tabanlı yeni yaklaşımlar. *J. Fac. Eng. Archit. Gazi Univ.* **2019**, *34*, 2173–2185. [CrossRef]

56. Chivers, I.; Sleightholme, J. An introduction to Algorithms and the Big O Notation. In *Introduction to Programming with Fortran*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 359–364.

57. Rubinstein-Salzedo, S. Big o notation and algorithm efficiency. In *Cryptography*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 75–83.

58. Safavian, S.R.; Landgrebe, D. A survey of decision tree classifier methodology. *IEEE Trans. Syst. Man Cybern.* **1991**, *21*, 660–674. [CrossRef]

59. Kim, K.S.; Choi, H.H.; Moon, C.S.; Mun, C.W. Comparison of k-nearest neighbor, quadratic discriminant and linear discriminant analysis in classification of electromyogram signals based on the wrist-motion directions. *Curr. Appl. Phys.* **2011**, *11*, 740–745. [CrossRef]

60. Rish, I. An empirical study of the naive Bayes classifier. In Proceedings of the IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, Seattle, WA, USA, 4–10 August 2001; pp. 41–46.

61. Chang, Y.-W.; Lin, C.-J. Feature ranking using linear SVM. In Proceedings of the Causation and Prediction Challenge, Hong-Kong, China, 15 December 2007–30 April 2008; pp. 53–64.

62. Jain, U.; Nathani, K.; Ruban, N.; Raj, A.N.J.; Zhuang, Z.; Mahesh, V.G. Cubic SVM classifier based feature extraction and emotion detection from speech signals. In Proceedings of the 2018 International Conference on Sensor Networks and Signal Processing (SNSP), Xi'an, China, 28–31 October 2018; pp. 386–391.

63. Horton, P.; Nakai, K. Better Prediction of Protein Cellular Localization Sites with the it k Nearest Neighbors Classifier. In Proceedings of the Ismb, Halkidiki, Greece, 12–15 June 1997; pp. 147–152.

64. Widasari, E.R.; Tanno, K.; Tamura, H. Automatic Sleep Disorders Classification Using Ensemble of Bagged Tree Based on Sleep Quality Features. *Electronics* **2020**, *9*, 512. [CrossRef]

65. Bajaj, V.; Pachori, R.B. Automatic classification of sleep stages based on the time-frequency image of EEG signals. *Comput. Methods Programs Biomed.* **2013**, *112*, 320–328. [CrossRef]

66. Kemp, B.; Zwinderman, A.H.; Tuk, B.; Kamphuisen, H.A.; Oberye, J.J. Analysis of a sleep-dependent neuronal feedback loop: The slow-wave microcontinuity of the EEG. *IEEE Trans. Biomed. Eng.* **2000**, *47*, 1185–1194. [CrossRef]

67. Hassan, A.R.; Bhuiyan, M.I.H. Computer-aided sleep staging using complete ensemble empirical mode decomposition with adaptive noise and bootstrap aggregating. *Biomed. Signal Process. Control* **2016**, *24*, 1–10. [CrossRef]

68. Jiang, D.; Lu, Y.-n.; Yu, M.; Yuanyuan, W. Robust sleep stage classification with single-channel EEG signals using multimodal decomposition and HMM-based refinement. *Expert Syst. Appl.* **2019**, *121*, 188–203. [CrossRef]

69. Kanwal, S.; Uzair, M.; Ullah, H.; Khan, S.D.; Ullah, M.; Cheikh, F.A. An image based prediction model for sleep stage identification. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 1366–1370.

70. Basha, A.J.; Balaji, B.S.; Poornima, S.; Prathilothamai, M.; Venkatachalam, K. Support vector machine and simple recurrent network based automatic sleep stage classification of fuzzy kernel. *J. Ambient Intell. Humaniz. Comput.* **2020**, 7191860. [CrossRef]

71. Jadhav, P.; Rajguru, G.; Datta, D.; Mukhopadhyay, S. Automatic sleep stage classification using time–frequency images of CWT and transfer learning using convolution neural network. *Biocybern. Biomed. Eng.* **2020**, *40*, 494–504. [CrossRef]

72. Michielli, N.; Acharya, U.R.; Molinari, F. Cascaded LSTM recurrent neural network for automated sleep stage classification using single-channel EEG signals. *Comput. Biol. Med.* **2019**, *106*, 71–81. [CrossRef]

73. Huang, J.; Ren, L.; Zhou, X.; Yan, K. An improved neural network based on SENet for sleep stage classification. *IEEE J. Biomed. Health Inform.* **2022**, *26*, 4948–4956. [CrossRef]

74. Kim, J.; Lee, J.; Shin, M. Sleep stage classification based on noise-reduced fractal property of heart rate variability. *Procedia Comput. Sci.* **2017**, *116*, 435–440. [CrossRef]

75. Shahin, M.; Ahmed, B.; Hamida, S.T.-B.; Mulaffer, F.L.; Glos, M.; Penzel, T. Deep learning and insomnia: Assisting clinicians with their diagnosis. *IEEE J. Biomed. Health Inform.* **2017**, *21*, 1546–1553. [CrossRef]

76. Karimzadeh, F.; Boostani, R.; Seraj, E.; Sameni, R. A distributed classification procedure for automatic sleep stage scoring based on instantaneous electroencephalogram phase and envelope features. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2017**, *26*, 362–370. [CrossRef]

77. Seifpour, S.; Niknazar, H.; Mikaeili, M.; Nasrabadi, A.M. A new automatic sleep staging system based on statistical behavior of local extrema using single channel EEG signal. *Expert Syst. Appl.* **2018**, *104*, 277–293. [CrossRef]

78. Zhou, J.; Wang, G.; Liu, J.; Wu, D.; Xu, W.; Wang, Z.; Ye, J.; Xia, M.; Hu, Y.; Tian, Y. Automatic Sleep Stage Classification With Single Channel EEG Signal Based on Two-Layer Stacked Ensemble Model. *IEEE Access* **2020**, *8*, 57283–57297. [CrossRef]

79. Zhang, J.; Yao, R.; Ge, W.; Gao, J. Orthogonal convolutional neural networks for automatic sleep stage classification based on single-channel EEG. *Comput. Methods Programs Biomed.* **2020**, *183*, 105089. [CrossRef] [PubMed]

80. Liu, G.-R.; Lo, Y.-L.; Malik, J.; Sheu, Y.-C.; Wu, H.-T. Diffuse to fuse EEG spectra–Intrinsic geometry of sleep dynamics for classification. *Biomed. Signal Process. Control* **2020**, *55*, 101576. [CrossRef]

81. Cai, Q.; Gao, Z.; An, J.; Gao, S.; Grebogi, C. A Graph-Temporal fused dual-input Convolutional Neural Network for Detecting Sleep Stages from EEG Signals. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *68*, 777–781. [CrossRef]

82. Loh, H.W.; Ooi, C.P.; Dhok, S.G.; Sharma, M.; Bhurane, A.A.; Acharya, U.R. Automated detection of cyclic alternating pattern and classification of sleep stages using deep neural network. *Appl. Intell.* **2021**, *52*, 2903–2917. [CrossRef]

83. Dhok, S.; Pimpalkhute, V.; Chandurkar, A.; Bhurane, A.A.; Sharma, M.; Acharya, U.R. Automated phase classification in cyclic alternating patterns in sleep stages using Wigner–Ville distribution based features. *Comput. Biol. Med.* **2020**, *119*, 103691. [CrossRef]

84. Sharma, M.; Patel, V.; Tiwari, J.; Acharya, U.R. Automated characterization of cyclic alternating pattern using wavelet-based features and ensemble learning techniques with eeg signals. *Diagnostics* **2021**, *11*, 1380. [CrossRef]