information

MDPI

Article

A New ECG Data Processing Approach to Developing an Accurate Driving Fatigue Detection Framework with Heart Rate Variability Analysis and Ensemble Learning

Junartho Halomoan [1,*], Kalamullah Ramli [1,*], Dodi Sudiana [1], Teddy Surya Gunawan [2,3] and Muhammad Salman [1]

[1] Department of Electrical Engineering, Universitas Indonesia, Depok 16424, Indonesia; dodi.sudiana@ui.ac.id (D.S.); muhammad.salman@ui.ac.id (M.S.)
[2] Department of Electrical and Computer Engineering, International Islamic University Malaysia, Kuala Lumpur 53100, Malaysia; tsgunawan@iium.edu.my
[3] School of Electrical Engineering, Telkom University, Bandung 40257, Indonesia
* Correspondence: junartho.halomoan@ui.ac.id (J.H.); kalamullah.ramli@ui.ac.id (K.R.)

Abstract: More than 1.3 million people are killed in traffic accidents annually. Road traffic accidents are mostly caused by human error. Therefore, an accurate driving fatigue detection system is required for drivers. Most driving fatigue detection studies concentrated on improving feature engineering and classification methods. We propose a novel driving fatigue detection framework concentrating on the development of the preprocessing, feature extraction, and classification stages to improve the classification accuracy of fatigue states. The proposed driving fatigue detection framework measures fatigue using a two-electrode ECG. The resampling method and heart rate variability analysis were used to extract features from the ECG data, and an ensemble learning model was utilized to classify fatigue states. To achieve the best model performance, 40 possible scenarios were applied: a combination of 5 resampling scenarios, 2 feature extraction scenarios, and 4 classification model scenarios. It was discovered that the combination of a resampling method with a window duration of 300 s and an overlap of 270 s, 54 extracted features, and AdaBoost yielded an optimum accuracy of 98.82% for the training dataset and 81.82% for the testing dataset. Furthermore, the preprocessing resampling method had the greatest impact on the model's performance; it is a new approach presented in this study.

Keywords: fatigue detection; resampling; electrocardiogram; fatigue driving; heart rate variability analysis

check for updates

Citation: Halomoan, J.; Ramli, K.; Sudiana, D.; Gunawan, T.S.; Salman, M. A New ECG Data Processing Approach to Developing an Accurate Driving Fatigue Detection Framework with Heart Rate Variability Analysis and Ensemble Learning. *Information* **2023**, *14*, 210. https://doi.org/10.3390/info14040210

Academic Editors: Sami Azam, Zubaer Ibna Mannan and Kawsar Ahmad

Received: 7 February 2023
Revised: 15 March 2023
Accepted: 18 March 2023
Published: 30 March 2023

1. Introduction

Every year, over 1.3 million people are killed in road traffic accidents. Southeast Asia has the highest road traffic accident death rate, with over 400,000 people killed yearly [1]. Human error, traffic circumstances, road designs, vehicle conditions, and weather conditions contribute to road traffic accidents [2]. Human error is the most significant contributor to road traffic accidents [3,4]. According to [5], driving fatigue, an example of human error, is the leading cause of road traffic accidents. To improve safety while driving, the driver needs a warning system for driving fatigue detection.

In the past few years, research on driving fatigue detection has developed, as reviewed in [6]. One example of driving fatigue detection studies that use eye, mouth, and/or face features obtained through video, results in an accuracy of up to 99.59% [7]. However, driving fatigue detection using physical features can only work well in specific conditions because its accuracy depends on factors such as the lighting, the color of the driver's background, and the color of the driver's skin [6]. This method also requires several things, such as a large amount of data storage, a large number of images for data training, and a high level of processing to analyze the images.

Other studies on driving fatigue detection have used physiological signals such as electroencephalogram (EEG) with an accuracy of 97.19% [8] and a combination of physiological signals (ECG, EEG, EOG) with an accuracy of 97% [9], obtaining high accuracy results, above 97%. However, driving fatigue detection using EEG or a combination of physiological signal sensors requires many electrodes or sensors to be attached to the driver, which could be intrusive. Furthermore, many movements in real driving can create artifacts in the measured signals and affect the accuracy of driving fatigue detection.

Therefore, a driving fatigue detection measurement method that uses fewer sensors or electrodes to determine the actual biological condition of the driver's body and is not affected by environmental factors, requiring less data storage and less computation than image processing, is needed. We chose a heart rate-related sensor, specifically ECG, as the physiological measurement method for driving fatigue detection. However, several problems are encountered when using ECG for driving fatigue detection, as follows:

1. As shown in Table 1, a number of driving fatigue detection studies using ECG only from 2017 to 2022 achieved a low accuracy, up to 92.5% [10]. Most driving fatigue detection studies have combined a heart rate-related sensor with other physiological sensors to obtain a more accurate classification model. For example, [11] combined ECG, EEG, and driving behavior sensors, resulting in an accuracy of 95.4%. Using more sensors attached to the driver's body is impractical in real-world driving applications.
2. Most driving fatigue detection studies (Table 1) have concentrated on developing feature engineering and classification techniques. Very few studies have focused on developing preprocessing methods. In fact, the preprocessing stage plays the most important part in the classification problem [12–14]. Therefore, with the right preprocessing method, driving fatigue detection using ECG could likely increase the model's performance.
3. Most driving fatigue detection studies have focused on developing the classification stage using neural networks and deep learning models to improve model performance. Table 1 shows very good results for the method of [6], which uses this strategy. For example, Huang and Deng proposed a novel classification model for detecting driver fatigue in 2022. They used a combination of neural network models, resulting in a 97.9% accuracy [15]. However, these methods are not perfect. They require a large amount of data and many computational resources, and if the model is overtrained to minimize the error, it may become less generalized [16,17].

We propose a driving fatigue detection framework with several approaches to solving these problems. The presented approaches were experimented with in several scenarios. These scenarios were employed in three stages of the proposed driving fatigue detection framework, as shown in Figure 1: the preprocessing, feature extraction, and classification stages. Our main contributions are as follows:

1. We chose the single-lead ECG method for driving fatigue measurement due to its ease of use. This method is sufficient for measuring heart rate, and heart rate variability is correlated with driver fatigue [18,19]. The ECG recording configuration used for driving fatigue detection in this study is a modified lead-I ECG with two electrodes placed at the second intercostal space.
2. In the preprocessing stage, we applied three types of resampling methods—no resampling, resampling only, and resampling with overlapping windows—to obtain and gather more information from the ECG data. Five resampling scenarios were employed in the driving fatigue detection framework (Figure 1) to determine which resampling scenario had the greatest impact on the model's performance.
3. In the feature extraction stage, we applied new feature extraction methods that had not been used in previous driving fatigue detection studies (Table 1). These are Poincare plot analysis and multifractal detrended fluctuation analysis to extract nonlinear properties from ECG data. There are 2 scenarios for the feature extraction method

employed in the driving fatigue detection framework: 29 and 54 features are used to evaluate whether the nonlinear analysis method has an effect on the model's performance. A 29-feature scenario covers the properties of the time domain and frequency domain analysis, while a 54-feature scenario covers the properties of the time domain, frequency domain, and nonlinear analysis.

4. In the classification stage, we preferred to use an ensemble learning model to produce a better classification performance than an individual model. We employed four ensemble learning model scenarios (Figure 1), AdaBoost, bagging, gradient boosting, and random forest, to assess which method gave the best model performance. In the proposed driving fatigue detection framework, 40 possible scenarios were employed. A combination of five resampling scenarios, two feature extraction method scenarios, and four ensemble learning model scenarios were considered to determine which scenario produced the best model performance on both the training and testing datasets. In addition, we employed the cross-validation method to evaluate model generalizability and the hyperparameter optimization method to optimize the trained model in the proposed driving fatigue detection framework.

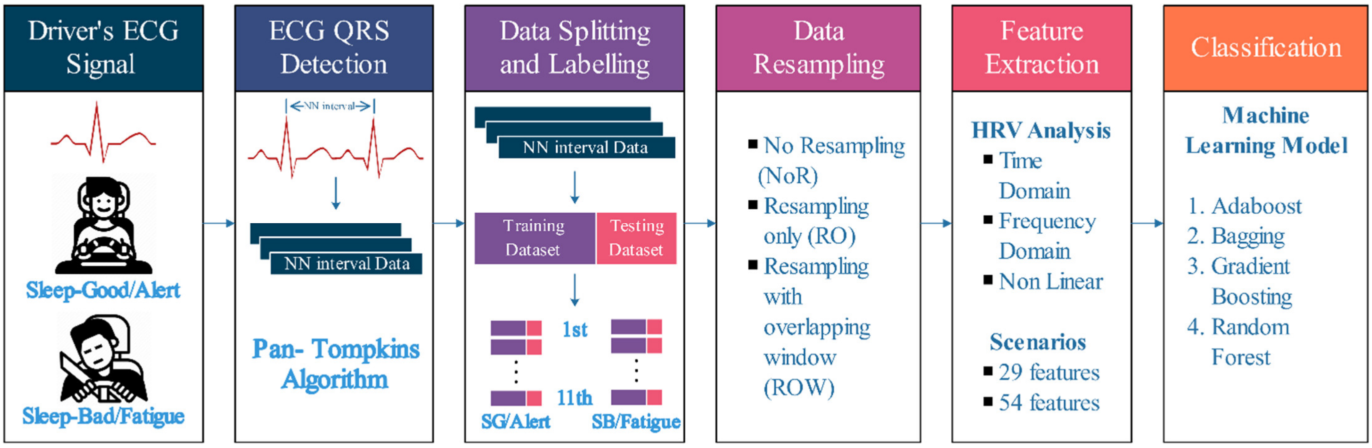


Figure 1. The proposed fatigue driving detection framework.

This paper consists of five sections. In Section 2, we describe the work associated with driving fatigue detection using ECG or other heart rate-related measurements. In Section 3, we describe the proposed driving fatigue detection framework model with several resampling, feature extraction, and ensemble learning methods. In Section 4, we present, analyze, and discuss the results. In Section 5, the results are summarized.

2. Related Works

There are three types of driving fatigue based on causal factors: passive task-related fatigue, active task-related fatigue, and sleep-related fatigue [19,20]. Passive task-related fatigue occurs when a driver experiences cognitive deterioration in specific settings, such as long-distance driving, tedious driving in low-traffic conditions, or continual noise. Active task-related fatigue occurs when a driver exhausts his or her cognitive function by performing a secondary task in the car, maneuvering during driving, or driving in heavy traffic. The relationship between sleep-related fatigue and a driver's sleep quality is significant. Therefore, sleep loss and disorders can induce fatigue [19].

There are two ways to measure fatigue while driving: based on condition data and based on performance data. Methods that measure fatigue based on condition data are divided into two types: subjective and objective measurement. Methods that measure fatigue based on performance data are also divided into two types: those that use vehicle data indicating driving behavior and those that use secondary task data [20]. The objective approach is a fatigue measurement method based on condition data using physiological and physical measurement methods on test subjects. In contrast, the subjective approach is a fatigue measurement method based on condition data using observer evaluation with a questionnaire, such as the Epworth Sleepiness Scale [21], Samn–Perelli Fatigue Scale [22], Stanford Sleepiness Scale [23], Karolinska Sleepiness Scale [24], and Chalder Fatigue Scale [25]. According to driving fatigue detection study reviews [6,26], most driving fatigue detection research uses the objective method to measure or detect a driver's fatigue; some use the subjective method as a comparison measure to determine the fatigue state's ground truth. In this paper, we chose a fatigue measurement-based condition data approach with an objective measurement method because it directly measures the actual condition of a driver. In addition, the subjective method is unnecessary because the drivers were accustomed to driving under two conditions, i.e., the well-rested condition and the sleep-deprived condition, to classify the two states.

Table 1. The literature on fatigue or drowsiness driving detection using ECG and other heart rate-related sensor measurements.

No.	Source	Number of Participants	Record. Time	Measurement	Features	Classification	Class	Accuracy
1	[27]	22	80 min	ECG and EEG	95	Support vector machine (SVM)	2	80.9

2	[28]	1st: 18;
		2nd: 24;
		3rd: 44
			90 min	EEG, ECG, EOG, steering behavior and lane positioning	54	Random forest	2	94.1

3	[29]	Unknown	5 min	ECG	4	SVM	2	83.9
4	[30]	6	67 min	ECG	12	SVM	2	AUC: 0.95
5	[10]	25	80 min	ECG	24	Ensemble logistic regression	2	92.5
6	[31]	6	60–120 min	ECG	Convolutional neural network (CNN) and recurrence plot		2	Accuracy: 70 Precision: 71 Recall: 85
7	[32]	25	Unknown	ECG	32	SVM	2	87.5
8	[33]	47	30 min	ECG signals and vehicle data	49	Random forest	2	91.2
9	[34]	23	33 min	driving behavior, reaction time and ECG	13	Eigenvalue of generation process of driving fatigue (GPDF)	3	72
10	[35]	45	45 min	BVP, respiration, skin conductance and skin temperature	73	CNN-LSTM	2	Recall: 82 Specificity: 71 Sensitivity: 93 AUC: 0.88
11	[11]	16	30 min	EEG, ECG, driving behavior	80	Majority voting classifier (kNN, LR, SVM)	2	95.4
12	[36]	16	Unknown	ECG	Multiple-objective genetic algorithm (MOGA) optimized deep multiple kernel learning support vector machine (D-MKL-SVM) + cross-correlation coefficient		2	AUC: 0.97
13	[37]	35	30 min	ECG, EEG, EOG	13	Artificial neural networks (ANNs)	2	83.5
14	[15]	9	>10 min	EDA, RESP, and PPG	15	ANN, backpropagation neural network, and cascade forward neural network	2	97.9
15	[38]	20	20 min	EEG and ECG	Product fuzzy convolutional network (PFCN)		2	94.19

Driving fatigue detection research using physical measurements (such as eye features, mouth features, face features, or combinations of physical features), as reviewed in [6], has produced very high accuracies, even up to 99.59% [7]. However, these measurement methods are limited by environmental conditions and the driver's appearance (such as skin color and the presence of a beard, a mustache, glasses, a hat, or tattoos) [39]. Other driving fatigue detection research using physiological or biological measurement methods, reviewed in [6], seems promising because it also results in very high accuracies, 97.19% with electroencephalogram (EEG) [8], 92.5% with electrocardiogram (ECG) [10], and 97.9% with multiple physiological measurements [15]. Moreover, these noninvasive methods address the physical measurement method's limitations but are also impractical because a few electrodes must be attached to the driver [39]. We chose ECG over EEG as a physiological measurement method because it requires fewer electrodes to be attached to the driver, making it easier to use. In addition, several driving fatigue detection studies have been conducted [10,27–30,32–34,40–42] using heart rate-related sensors such as ECG, photoplethysmography, blood volume pulse, and oximeters, which showed that heart rate variability has a relationship with the driver's fatigue status. Therefore, ECG is a suitable choice for driving fatigue detection.

Among the driving fatigue detection studies using heart rate-related sensors from 2017–2022, as shown in Table 1, 7 [10,11,15,28,33,36,38] of the 15 studies that have accuracies above 90% or area under the curve values of 0.97 are interesting to discuss. First, Babaeian et al. [10] proposed a driving fatigue detection framework using wavelet transform and ensemble logistic regression, achieving an accuracy of 92.5%. They focused on developing driving fatigue detection using a feature extraction method to extract more information and an ensemble machine learning model to obtain a better classification model. Second, Huang and Deng [15] proposed a driving fatigue detection framework using principal component analysis and the cascade forward neural network, achieving an accuracy of 97.9%. They made developments in the preprocessing stage by using principal component analysis to remove redundant information from the original data, using multiple biological sensors to obtain more information, and using an artificial neural network model to achieve better fatigue classification. They obtained the highest accuracy, as shown in Table 1. However, many sensors are attached to the driver, such as electrodermal activity, respiration, and photoplethysmography sensors, which might interfere with driving. Third, Mårtensson et al. [28] proposed a driving fatigue detection framework using many feature extraction methods and a random forest and achieved an accuracy of 94.1%. They made developments in feature engineering by using a combination of multiple biological sensors (EEG, ECG, and EOG) and driving performance sensors. They also used the random forest to obtain a better classification model. The weakness of this method is the same as that of [15]; having many electrodes attached to the driver can be intrusive.

Fourth, Arefnezhad et al. [33] proposed a driving fatigue detection framework using multiple sensors and a random forest and achieved an accuracy of 91.2%. Their accuracy was the lowest among the top seven driving fatigue detection research accuracies, as shown in Table 1. They developed this method by using the fusion of ECG and vehicle data and extracted 49 features from it. Fifth, Du et al. [38] proposed a driving fatigue detection framework using a product fuzzy convolutional network and achieved an accuracy of 94.19%. They developed a new model based on deep learning for feature extraction and classification. Sixth, Gwak et al. [11] proposed a driving fatigue detection framework using hybrid sensing (EEG, ECG, and driving behavior sensors), extracted 80 features with a random forest, and achieved an accuracy of 95.4%. They had the same weakness as the methods of [28,38], which used many electrodes attached to the driver because EEG was used in the driving fatigue detection framework. Finally, Chui et al. [36] suggested a driving fatigue detection framework employing a cross-correlation coefficient for feature extraction and an MOGA-optimized D-MKL SVM for classification and achieved an area under the curve of 0.97. They used the Cyclic Alternating Pattern Sleep Database, which was implemented in neither real-world driving nor virtual driving. Their dataset did not reflect the fatigue status of the driver because the participants did not drive, and a number of participants had already been diagnosed with pathologies [43,44]. This dataset should only be used to investigate sleep-related pathologies.

It can be concluded generally from Table 1 that most driving fatigue detection studies use more physiological sensors to extract more information about the actual status of drivers. However, attaching more sensors to the driver might be more intrusive [6,45]. Therefore, it is important to use a few sensors that can reliably measure the driver's fatigue and, with the right signal processing methods, can enable high-accuracy driving fatigue detection.

In addition, most of the driving fatigue detection research described in Table 1 focused on improvements in preprocessing methods to remove artifacts from original data, feature extraction methods and classification models that had never been used in previous research, and the development of new deep learning model architectures. By evaluating several related works, it is clear that driving fatigue detection techniques can still be improved, such as the measurement method, the preprocessing method, the feature extraction method, the classification model, or a combination of these.

3. Materials and Methods

3.1. Dataset

As noted in the section on related work, we used an objective measurement method that has been widely used and validated in earlier driving fatigue detection research. We chose ECG due to its ease of use and proven relationship with heart rate variability and fatigue. The dataset used for the driving fatigue detection experiment was obtained from [18]. In the dataset, there were 11 healthy participants (10 men and a woman) aged 24–28 years. Each of them had a driving license and performed a simulated drive for at least 30 min in 2 driving conditions (alert and fatigued). Each of the conditions was tested on a particular day. For the alert driving condition, the drivers were instructed to have slept well for at least seven hours. To induce fatigue conditions, the drivers were instructed to ensure sleep deprivation by going to sleep late at night.

Several physiological signals were recorded during simulated driving: 64 channels for EEG, 2 for EOG, and 2 for ECG. The configuration used for ECG signal recording was a modified lead-I with two electrodes placed at the second intercostal space. The driver's physiological signals were recorded using a Biosemi Active Two System with a sampling rate of 512 Hz [18]. The sampling rate used in the experiments met the minimum requirement for heart rate variability analysis, which is 250 Hz [46]. All of them were collected into 68 channels of recorded data series. In this paper, we only used ECG signals for driving fatigue detection, with 2 of the 68 recorded signal channels labeled Sleep-Good and Sleep-Bad. The Sleep-Good labeled data indicate alert driving conditions, whereas the Sleep-Bad labeled data indicate fatigued driving conditions. Table 2 shows each participant's ECG data sample duration under the two conditions, measured in minutes and milliseconds.

3.2. Driving Fatigue Detection Framework

The initial stage of ECG data processing shown in Figure 1 is data acquisition, which is explained in the dataset section. The dataset used in this study was obtained from [18].

Heart rate variability is a measure of fluctuations in the interbeat interval calculated by extracting the beat-to-beat or RR interval from an ECG signal [47]. A robust and accurate QRS detection algorithm is needed to obtain valid heart rate variability data [48]. The Pan–Tompkins algorithm, which performs well in ECG beat segmentation [49], was employed in the second stage, the detection of QRS within the driving fatigue detection framework (Figure 1). The Pan–Tompkins approach consists of several steps, starting with the elimination of the DC offset, signal filtering with a digital bandpass filter and derivative filter, signal squaring, moving window integration, and identification of the QRS complex. The minimum sampling rate required to perform the Pan–Tompkins algorithm is 200 Hz [50], and the dataset used in this study meets its qualifications.

Table 2. The ECG data sample duration and total interval of each participant.

Participant	Sleep-Good (SG)/Alert			Sleep-Bad (SB)/Fatigue		
	ECG Data		Total NN Interval (T_{to}) in Msec	ECG Data		Total NN Interval (T_{to}) in Msec
	in Mins	in Msec		in Mins	in Msec	
1	30.05	1,803,000	1,800,267	30.05	1,803,000	1,800,945
2	30.05	1,803,000	1,800,827	30.05	1,803,000	1,801,166
3	30.05	1,803,000	1,801,157	30.05	1,803,000	1,800,932
4	53.88	3,232,750	3,229,156	30.05	1,803,000	1,801,279
5	51.53	3,091,500	3,088,254	31.45	1,887,250	1,884,670
6	30.86	1,851,500	1,849,644	30.05	1,803,000	1,800,605
7	40.13	2,407,750	2,404,552	30.05	1,803,000	1,800,367
8	44.52	2,671,250	2,668,485	33.74	2,024,250	2,022,111
9	35.1	2,106,250	2,103,198	30.05	1,803,000	1,800,723
10	36.1	2,166,250	2,163,819	30.05	1,803,000	1,800,733
11	23.59	1,415,482	1,413,316	30.05	1,803,000	1,800,578
Min	23.59	1,415,482	1,413,316	30.05	1,803,000	1,800,367
Max	53.88	3,232,750	3,229,156	33.74	2,024,250	2,022,111
St. Dev.	9.64	578,288	577,807	1.15	68,967	68,949
Average	36.90	2,213,794	2,211,152	30.51	1,830,773	1,828,555

After detecting the R waves in the ECG data, the RR interval between two adjacent QRSs could be measured. In this study, we prefer the term "NN interval" over "RR interval" because the NN interval is measured between two adjacent detected QRS complexes and excludes unreliable RR intervals [51]; it is measured in milliseconds. All the total NN intervals of each participant under the two different conditions are shown in Table 2. However, the duration of ECG recordings for each participant was not uniform. As shown in Table 2, the ECG data sample with the shortest duration was recorded for the 11th participant in the Sleep-Good condition, 23.59 min or 1,415,482 ms, and the total duration of the NN interval for the 11th participant in the Sleep-Good condition was 1,413,316 ms. To obtain a balanced set of data, we used the shortest duration of the NN interval as the duration reference for all participants in both the Sleep-Good and Sleep-Bad conditions. As illustrated in Figure 1, the following subsections further explain the third to sixth stages.

All of the stages illustrated in Figure 1 were programmed in Python 3 [52] with the packages, pandas [53], NumPy [54], and Scikit-learn [55], using a Spyder integrated development environment [56]. All experiments were performed on an AMD Ryzen 5 at 3.6 GHz with 16 GB of RAM and the Windows 10 operating system.

3.3. Data Splitting and Labelling

In most real-world machine learning applications where only a small amount of data are available, splitting the data is useful for evaluating the algorithm's performance. The most common method for splitting data is to divide them into two portions. The first portion of the data is used to train the algorithm and is referred to as the training dataset. The remaining data, referred to as the testing dataset, which could be considered "new data," are used to validate the algorithm and evaluate the model's ability to predict the future. The testing dataset must be independently and identically distributed [57]. There are no standards for data splitting ratios, as ratios may vary between studies. Most studies favor a ratio of 80% to 20%, respectively, for the training and testing datasets [58]. Two conditions must be met when splitting the dataset: the training set must be large enough to represent meaningful data, and the testing set must be sufficient to evaluate the model's performance [59].

We used heart rate variability analysis in the feature extraction stage of the proposed driving fatigue detection framework (Figure 1). The heart rate variability guidelines suggest that 5 min is the typical duration window for heart rate variability analysis [60]. For a proper data split, we separated 5 min of NN interval data from each participant's total NN interval duration for the testing dataset. The remainder of the NN interval was utilized for the training dataset. The ratio of the testing dataset is defined as:

$$R_{te} = \frac{T_{wte}}{T_{to}} \times 100\% \tag{1}$$

where T_{to} is the total NN interval duration for each participant for all training and testing datasets, measured in milliseconds, and T_{wte} is the NN interval duration for each participant used for the testing dataset, measured in milliseconds. As explained in the driving fatigue detection framework section, the total NN interval duration of the eleventh participant was used as the duration reference dataset for all participants. The total NN interval duration for all participants was extracted up to approximately 1,413,316 ms. Referring to Equation (1), the testing dataset's ratio was 22%, and the ratio of the training dataset was 78%.

As illustrated in Figure 1, the output of the second stage is a collection of NN interval data for all participants under the two conditions. In the third stage, the NN interval data were split into training and testing datasets at a ratio of 78% to 22%. Both datasets were labeled based on the dataset's parts and the participants' conditions. Sleep-Good (SG) was classified as an alert condition, while Sleep-Bad (SB) was classified as a fatigue condition.

3.4. Resampling Methods

In statistics, the resampling method repeatedly draws samples from the original data to obtain more information from a sample [61]. Several driving fatigue detection studies (Table 1) have employed the resampling method with the following window or epoch sizes: 5-minute window [27], 5-second sample [29], 2-minute epoch [31], 120-second window [30], 40-second window [33], 5-minute window [35], and unknown epoch size [28,37]. Most of these studies used resampling without an overlapping window method and did not investigate the effect of the resampling methods on their model performance. These findings motivated us to investigate more closely how the resampling method works and its effects on detecting driver fatigue.

References [62,63] stated that an optimal ensemble learning method depends on the diversity of each learner. Diversity can be enhanced by dividing the original dataset into smaller subsets of data. We hypothesized that the resampling method could be used to enhance diversity by dividing the dataset into smaller subsets of data. Moreover, resampling with an overlapping window method yields more subsets of data than resampling without an overlapping window method. As a result, resampling with an overlapping window method would enhance diversity more than resampling without an overlapping window method and would impact the driving fatigue detection accuracy. We applied three resampling methods in the proposed driving fatigue detection framework: no resampling, resampling only, and resampling with an overlapping window. No resampling (NoR) is a method of putting all NN interval data in a dataset into a single window. Resampling only (RO), which is another name for "resampling without an overlapping window," is a method of dividing all NN interval data in a dataset into multiple windows. Resampling with overlapping windows (ROW) divides all NN interval data in a dataset into multiple windows with an overlap between two neighboring windows. These methods are illustrated in Figure 2, Figure 3 and Figure 4, respectively.

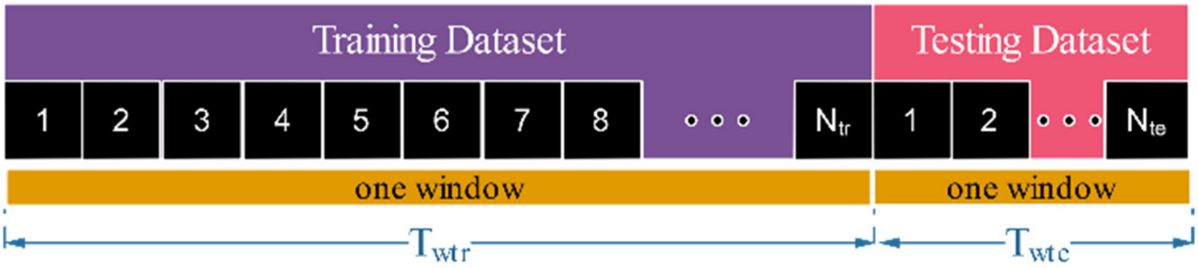


Figure 2. No resampling (NoR).

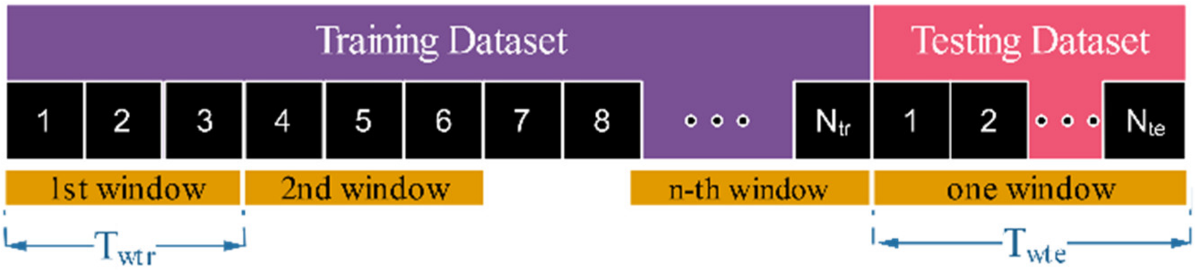


Figure 3. Resampling only (RO).

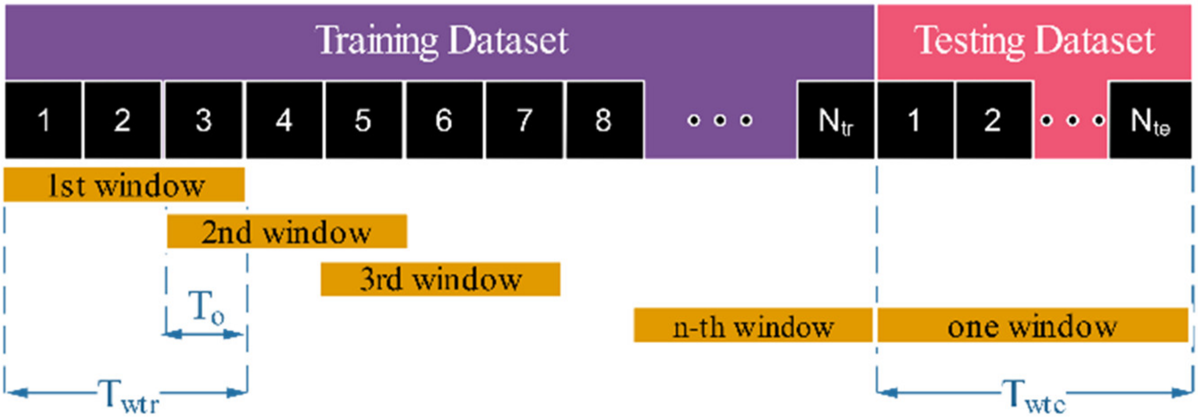


Figure 4. Resampling with overlapping windows (ROW).

In resampling methods, two parameters are described: T_w and T_o. The parameter T_w is the duration of a window, whereas the parameter T_o is the duration of the overlap between two neighboring windows. According to heart rate variability guidelines [60], the typical duration of a window for heart rate variability analysis is 5 min, so we set the T_w duration to 5 min or 300 s. However, there was no reference or standard for the duration of the overlapping window. We used 3 settings for the duration of the overlap (T_o), 210 s, 240 s, and 270 s, to assess whether there were significant differences in the model's performance. These values were suggested because we hypothesized that a longer duration of the overlap (T_o) would improve the model's performance, but the overlap duration (T_o) must not be longer than the window duration (T_w). Table 3 summarizes the various resampling methods discussed in this section into five possible resampling scenarios within the proposed driving fatigue detection framework.

Table 3. List of resampling scenarios used in the proposed driving fatigue detection framework.

Resampling Method	Training Dataset (78%)		Testing Dataset (22%)		Term
	Duration	Number of Windows (N_{tr})	Duration	Number of Windows (N_{te})	
No resampling	$T_{wtr} \approx$ 18 min or 1080 s	1	$T_{wte} \approx$ 300 s	1	NoR
Resampling only	$T_{wtr} \approx$ 300 s $T_o \approx$ 0 s	3			RO
Resampling with overlapping windows	$T_{wtr} \approx$ 300 s $T_o \approx$ 210 s	9			ROW210
	$T_{wtr} \approx$ 300 s $T_o \approx$ 240 s	14			ROW240
	$T_{wtr} \approx$ 300 s $T_o \approx$ 270 s	27			ROW270

Due to the various window durations and overlapping window durations shown in Table 3, each resampling scenario produces a distinct number of windows. The total NN interval duration of each participant for the testing dataset was set to approximately 5 min, and the total NN interval duration of each participant for all training and testing datasets was set to approximately 1,413,316 ms or 23.5 min. The number of windows in the training dataset can be calculated as follows:

$$N_{tr} = \left\lfloor \frac{(T_{to} - T_{wte}) - T_{wtr}}{T_{wtr} - T_o} \right\rfloor + 1\;,\; T_{wtr} > T_o,\; T_o \geq 0 \tag{2}$$

where T_{to} is the total NN interval duration of each participant for all training and testing datasets, measured in milliseconds, T_{wte} is the total NN interval duration of each participant for the testing dataset, measured in milliseconds, T_{wtr} is the window duration for the training dataset used for heart rate variability analysis, measured in milliseconds, and T_o is the overlap duration between two consecutive windows, measured in milliseconds. Using (2), the number of windows in the training dataset can be calculated and is shown in Table 3.

3.5. Feature Extraction

After dividing the NN interval data into multiple windows using the resampling method, we extracted the information from each window. The feature extraction method was used to interpret the physiological condition of the driver so we could distinguish the driver's condition between one event and another. The most well-known method used for extracting features from NN interval data is heart rate variability analysis, which was first introduced in the guidelines of [60], which analyzed the variations between consecutive heartbeats and RR intervals. This method has been commonly used and tested in previous driving fatigue detection research [15,27–30,32,33,35,37].

In [60], two measures are used to analyze heart rate variability: the time domain and frequency domain. For the time domain measurements, we used statistical and geometrical analysis methods. Both methods were used to analyze the oscillation of NN interval data. Table 4 shows 20 features from the time domain applied in the driving fatigue detection framework shown in Figure 1. In the frequency domain, the power spectral density (PSD) feature was extracted from NN interval data to estimate the power distribution; this is also called spectral analysis. There are two approaches to calculating PSD: parametric and nonparametric. We chose the nonparametric approach because it features a simpler algorithm and less computation than parametric [64]. The PSD estimator method used in the driving fatigue detection experiment was Welch's method. The estimation of PSD was analyzed into five frequency bands: ultralow frequency (ULF)—under 0.003 Hz, very low frequency (VLF)—between 0.003 Hz and 0.04 Hz, low frequency (LF)—between 0.04 Hz and 0.15 Hz, high frequency (HF)—between 0.15 Hz and 0.4 Hz, and very high frequency (VHF) [60]. Table 4 shows nine features extracted from the frequency domain and applied in the driving fatigue detection framework shown in Figure 1. The total number of features extracted from the time domain and frequency domain analysis is 29.

Reference [65] proved that feature extraction methods using the frequency domain and nonlinear approach could distinguish different psychological states better than the frequency domain approach alone. We used two nonlinear measurement methods to extract nonlinear features from NN intervals: Poincare plot analysis (PPA) and multifractal detrended fluctuation analysis (MF-DFA). These methods were not used in the driving fatigue detection research presented in Table 1.

Table 4. List of extracted features of NN interval data in time and frequency domain analysis.

No.		Type of Analysis	Feature Name	Feature Description
1	Time Domain	Statistical analysis [47,66]	MeanNN	Mean of the NN intervals of time series data
2			SDNN	Standard deviation of the NN intervals of time series data
3			SDANN	Standard deviation of the average NN intervals of each 5-minute segment of time series data
4			SDNNI	Mean of the standard deviations of NN intervals of each 5-minute segment of time series data
5			RMSSD	Square root of the mean of the sum of successive differences between adjacent NN intervals
6			SDSD	Standard deviation of the successive differences between NN intervals of time series data
7			CVNN	Ratio of SDNN to MeanNN
8			CVSD	Ratio of RMSSD and MeanNN
9			MedianNN	Median of the absolute values of the successive differences between NN intervals of time series data
10			MadNN	Median absolute deviation of the NN intervals of time series data
11			HCVNN	Ratio of MadNN to Median
12			IQRNN	Interquartile range (IQR) of the NN intervals
13			Prc20NN	The 20th percentile of the NN intervals
14			Prc80NN	The 80th percentile of the NN intervals
15			pNN50	The proportion of NN intervals greater than 50 ms out of the total number of NN intervals of time series data
16			pNN20	The proportion of NN intervals greater than 20 ms out of the total number of NN intervals of time series data
17			MinNN	Minimum of the NN intervals of time series data
18			MaxNN	Maximum of the NN intervals of time series data
19		Geometrical analysis [47,66]	TINN	Width of the baseline of the distribution of the NN interval obtained by triangular interpolation
20			HTI	HRV triangular index
21	Frequency Domain	Spectral analysis [47,66]	ULF	Power in the ultralow frequency range
22			VLF	Power in the very low-frequency range
23			LF	Power in the low-frequency range
24			HF	Power in the high-frequency range
25			VHF	Power in the very high-frequency range
26			LFHF	Ratio of LF to HF
27			LFn	Normalized power in the low-frequency range
28			HFn	Normalized power in the high-frequency range
29			LnHF	Natural logarithm of power in the high frequency range

Several studies have used PPA to analyze athlete fatigue [67], analyze driver fatigue [68], and evaluate driver fatigue [69]. In a scatter diagram, PPA shows each RR interval data point qualitatively as a function of the previous RR interval data [70]. As in [71], PPA can be calculated quantitatively with the parameters shown in Table 5.

The autonomic nervous system is divided into two systems: the sympathetic nervous system and the parasympathetic nervous system. Driver fatigue is closely related to the sympathetic and parasympathetic systems [72,73]. In PPA, parameter SD1 reflects parasympathetic activity, whereas parameter SD2 reflects both sympathetic and parasympathetic activity [71]. Accordingly, the PPA method can be used to extract nonlinear features and analyze driving fatigue.

MF-DFA has been used in a number of studies, such as to assess fatigue using EMG [74], to evaluate the fractal features of each individual using EEG [75], to study the fatigue of a runner using ECG [76], and to analyze driving fatigue stages using EEG [77]. In this study, we extracted the fractal characteristics of each participant's ECG under the two conditions using MF-DFA. Table 5 shows the features extracted using MF-DFA, and further details about MF-DFA can be found in [78]. As shown in Tables 4 and 5, the total number of features extracted from the time domain, frequency domain, and nonlinear analysis is 54.

In the proposed driving fatigue detection framework (Figure 1), 2 scenarios of feature extraction methods were employed, with 29 and 54 features, to evaluate the performance of feature extraction methods in detecting the fatigue state.

Table 5. List of extracted features of NN interval data in nonlinear analysis.

No	Type of Analysis	Feature Name	Feature Description
1	Poincare analysis [66,71,79]	SD1	Standard deviation perpendicular to the line of identity
2		SD2	Standard deviation along the identity line
3		SD1/SD2	Ratio of SD1 to SD2
4		S	Area of the ellipse described by SD1 and SD2
5		CSI	Cardiac Sympathetic Index
6		CVI	Cardiac Vagal Index
7		CSI modified	Modified CSI
8	Detrended fluctuation analysis (DFA) [66,78,80]	DFA $\alpha1$	Detrended fluctuation analysis
9		MFDFA $\alpha1$—Width	Multifractal DFA $\alpha1$—width parameter
10		MFDFA $\alpha1$—Peak	Multifractal DFA $\alpha1$—peak parameter
11		MFDFA $\alpha1$—Mean	Multifractal DFA $\alpha1$—mean parameter
12		MFDFA $\alpha1$—Max	Multifractal DFA $\alpha1$—maximum parameter
13		MFDFA $\alpha1$—Delta	Multifractal DFA $\alpha1$—delta parameter
14		MFDFA $\alpha1$—Asymmetry	Multifractal DFA $\alpha1$—asymmetry parameter
15		MMFDFA $\alpha1$—Fluctuation	Multifractal DFA $\alpha1$—fluctuation parameter
16		MFDFA $\alpha1$—Increment	Multifractal DFA—increment parameter
17		DFA $\alpha2$	Detrended fluctuation analysis
18		MFDFA $\alpha2$—Width	Multifractal DFA $\alpha2$—width parameter
19		MFDFA $\alpha2$—Peak	Multifractal DFA $\alpha2$—peak parameter
20		MFDFA $\alpha2$—Mean	Multifractal DFA $\alpha2$—mean parameter
21		MFDFA $\alpha2$—Max	Multifractal DFA $\alpha2$—maximum parameter
22		MFDFA $\alpha2$—Delta	Multifractal DFA $\alpha2$—delta parameter
23		MFDFA $\alpha2$—Asymmetry	Multifractal DFA $\alpha2$—asymmetry parameter
24		MFDFA $\alpha2$—Fluctuation	Multifractal DFA $\alpha2$—fluctuation parameter
25		MFDFA $\alpha2$—Increment	Multifractal DFA $\alpha2$—increment parameter

3.6. Classification Model

References [6,26] are detailed surveys of classification algorithms used for fatigue or drowsiness detection. Support vector machine (SVM), neural network, and convolutional neural network classifiers are the most implemented methods in driving fatigue detection because they have better accuracy than other classifiers, such as K-nearest neighbors, naive Bayes, and decision trees. Nonetheless, the SVM classifier has weaknesses in parameter selection when determining the optimum values and an excessive processing time when using massive datasets to solve optimization problems [81]. In Table 1, the best driving fatigue detection performance using SVM had an area under the curve of 0.97 [36], while the highest driving fatigue detection performance utilizing a combination of neural network approaches had an accuracy of 97.9% [15]. Although neural network classifiers appear to be highly accurate at detecting fatigue, it takes a long time for them to process large and complex models. It is rather difficult to evaluate the trained model if it is not tested on new data [82].

We chose an ensemble learning approach for classification in the proposed driving fatigue detection framework (Figure 1). Ensemble learning is a decision-making technique that involves mixing more than one learner. By mixing numerous models, the faults of a single learner model are likely to be compensated for by other learners. As a result, using ensemble methods could improve classification performance. Moreover, these methods have other advantages over single learners: they avoid the possibility of overfitting and have lower computational cost and representation [63]. To obtain the optimal model with the highest-accuracy fatigue state classification, the four most commonly used ensemble learning models in biomedical and healthcare studies [83] are deployed and evaluated in the proposed driving fatigue detection framework (Figure 1): AdaBoost, bagging, gradient boosting, and random forest. Further details of these four ensemble learning models can be found in [84].

3.7. Cross-Validation and Hyperparameter Optimization

The fundamental issue with the data splitting method is how to split the data appropriately; improper data splitting can result in an excessively high variance or bias in model performance [85]. However, the ratio of data between the training and testing datasets was already set at 78% to 22%. Therefore, cross-validation, a common technique for balancing the bias and variance of a model, was employed to reduce the possibility of high variance or bias in the model's performance [86]. In Figure 5, after extracting features from every 5 min window, we applied a k-fold cross-validation method to the training dataset. There is no standard rule for selecting the value of k. However, increasing the value of k reduces the size of the test set, resulting in less precise and more coarse performance metric measurements. Because of this, the data mining community appears to agree that k = 10 is an acceptable compromise. This value of k is particularly favorable because it makes predictions using 90% of the data, making it more likely to generalize to the full dataset [87]. In this study, we chose the 10-fold cross-validation method, which splits the dataset into 10 groups, or folds, of approximately equal size. One fold served as a holdout or validation set for each iteration, while the remaining nine folds served as the training set. Furthermore, the ten iterations utilized all folds for training the model. The model, which was trained using the training set, was validated using the validation set, resulting in an accuracy score for each iteration. The average of all the accuracy scores was the cross-validation accuracy score, or the accuracy score of the training dataset. We use the term "accuracy of the training dataset" in the performance metrics report.

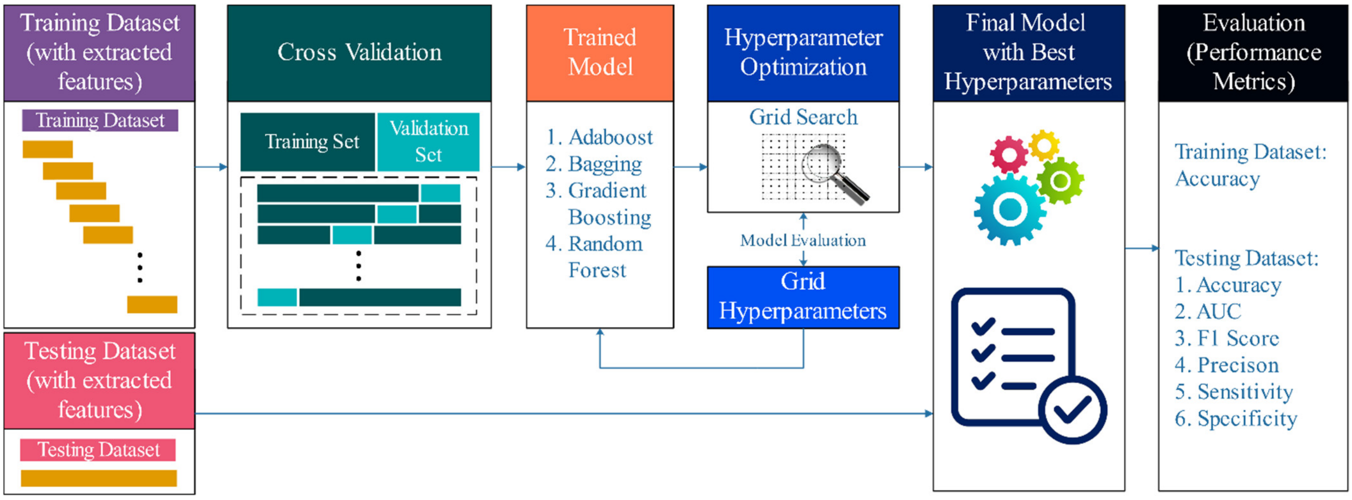


Figure 5. Data splitting, cross-validation, hyperparameter optimization, and evaluation.

In a machine learning model, there are two types of parameters: model parameters, which can be estimated by fitting training data to the model and then updated as the model learns, and hyperparameters, which define the model architecture and must be specified before training. Hyperparameter optimization is a method of constructing the best model architecture with the optimal hyperparameter configuration. Optimized hyperparameters can significantly increase the model's performance [88]. In [89], various hyperparameter optimization algorithms were given. As shown in Figure 5, we selected the grid search strategy for hyperparameter optimization because it is simple to implement and can be executed in parallel. Grid search is one of the most popular techniques for hyperparameter optimization. It is a brute-force approach that evaluates each hyperparameter combination in the grid of hyperparameters, whose values are specified manually by the user [88]. Table 6 shows the grid search hyperparameters used for hyperparameter optimization. Hyperparameter optimization combined with the cross-validation method was used to optimize the trained model and yield the final model with the optimal hyperparameters. Later, the final model was evaluated using the testing dataset. Several performance metrics were used to evaluate the model's performance, such as accuracy, area under the curve, F1 score, precision, sensitivity, and specificity.

Table 6. Grid search hyperparameters and their range for all models.

Model	Hyperparameter	Description	Range
AdaBoost	n_estimators	The maximum number of estimators	[10, 20, 50, 100, 500]
	learning_rate	The weight that is assigned to each weak learner in the model	[0.0001, 0.001, 0.01, 0.1, 1.0]
Bagging	n_estimators	The number of base estimators in the ensemble	[10, 20, 50, 100]
Gradient boosting	n_estimators	The number of boosting stages to perform	[10, 100, 500, 1000]
	learning_rate	The step size that controls the model weight update at each iteration	[0.001, 0.01, 0.1]
	Subsample	A random subset used for fitting the individual base learners	[0.5, 0.7, 1.0]
	max_depth	The maximum number of levels in a decision tree	[3, 7, 9]
Random forest	n_estimators	The number of trees in the forest	[10, 20, 50, 100]
	max_features	The number of features to consider when looking for the best split	['sqrt', 'log2']

4. Results and Discussion

This section presents the results of 40 possible scenarios employed in the proposed driving fatigue detection framework. These scenarios, described in Table 7, combine five resampling scenarios, two feature extraction method scenarios, and four ensemble learning model scenarios.

Table 7. Summary of all scenarios in the data resampling, feature extraction, and classification stages.

Data Resampling Scenarios		Feature Extraction Scenarios	Classification Scenarios
Term	**Description**		
NoR RO ROW210 ROW240 ROW270	No Resampling Resampling only (T_o = 0 s) Resampling with overlapping window T_o = 210 s Resampling with overlapping window T_o = 240 s Resampling with overlapping window T_o = 270 s	• 29 features (total features extracted by time and frequency domain analysis, shown in Table 4) • 54 features (total features extracted by time domain, frequency domain, and nonlinear analysis, shown in Tables 4 and 5)	• AdaBoost • Bagging • Gradient boosting • Random forest

This section is split into four subsections that present and discuss the results. Sections 4.1 and 4.2 analyze the effects of different resampling methods and the number of features used in the driving fatigue detection framework on the model's performance. Section 4.3 discusses the considerations in selecting the models that were applied in the proposed driving fatigue detection framework and compares the proposed driving fatigue detection framework to other driving fatigue detection studies. Section 4.4 discusses future work directions.

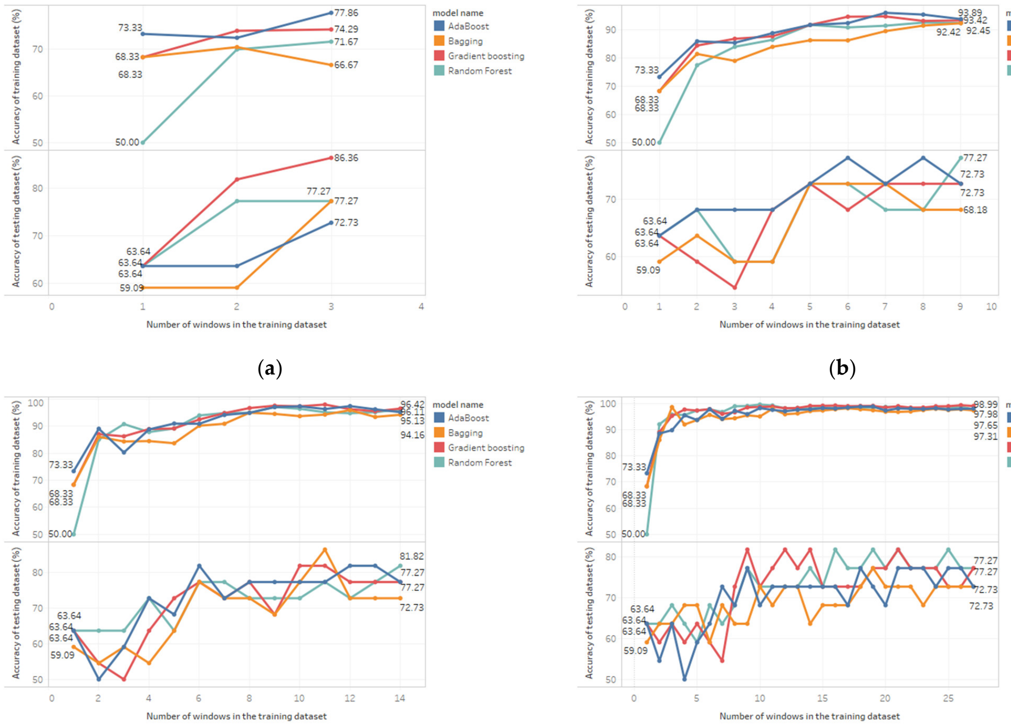
4.1. *The Effect of Various Resampling Methods on the Model's Performance*

As shown in Tables 3 and 7, there are five resampling scenarios: NoR, RO, ROW210, ROW240, and ROW270, which are analyzed and discussed. In this subsection, we focus on analyzing the other four resampling scenarios: RO, ROW210, ROW240, and ROW270, which resampled the training dataset. To describe the effect of the four resampling methods on the model's performance, we present Table 8, showing comprehensive accuracy results for the four resampling scenarios, and Figures 6 and 7, showing the accuracy performance of the four classification models with twenty-nine features and fifty-four features, respectively.

Table 8. The accuracy of the classification models that used 1 window or all windows in the training dataset and testing dataset for 4 resampling scenarios (RO, ROW210, ROW240, and ROW270) with 29 and 54 features. In each classifier with different features and resampling scenarios, the best result in terms of accuracy of the training dataset is shown in bold.

Classifier	Features	Resampling Scenario	Accuracy on the Training Dataset (%)			Accuracy on the Testing Dataset (%)		
			1 Window	All Windows	Increase	1 Window	All Windows	Increase
AdaBoost	29	RO	73.33	77.86	4.53	63.64	72.73	9.09
		ROW210	73.33	93.89	20.56	63.64	72.73	9.09
		ROW240	73.33	95.13	21.8	63.64	77.27	13.63
		ROW270	73.33	**97.98**	24.65	63.64	72.73	9.09
	54	RO	75	76.43	1.43	54.55	72.73	18.18
		ROW210	75	94.45	19.45	54.55	86.36	31.81
		ROW240	75	96.12	21.12	54.55	86.36	31.81
		ROW270	75	**98.82**	23.82	54.55	81.82	27.27
Bagging	29	RO	68.33	66.67	-	59.09	77.27	18.18
		ROW210	68.33	92.42	24.09	59.09	68.18	9.09
		ROW240	68.33	94.16	25.83	59.09	72.73	13.64
		ROW270	68.33	**97.31**	28.98	59.09	72.73	13.64
	54	RO	60	72.86	12.86	45.45	77.27	31.82
		ROW210	60	90.89	30.89	45.45	68.18	22.73
		ROW240	60	93.53	33.53	45.45	63.64	18.19
		ROW270	60	**97.31**	37.31	45.45	63.64	18.19
Gradient boosting	29	RO	68.33	74.29	5.96	63.64	86.36	22.72
		ROW210	68.33	93.42	25.09	63.64	72.73	9.09
		ROW240	68.33	96.42	28.09	63.64	77.27	13.63
		ROW270	68.33	**98.99**	30.66	63.64	77.27	13.63
	54	RO	66.67	74.52	7.85	50	72.73	22.73
		ROW210	66.67	94.42	27.75	50	72.73	22.73
		ROW240	66.67	95.78	29.11	50	72.73	22.73
		ROW270	66.67	**98.66**	31.99	50	72.73	22.73
Random forest	29	RO	50	71.67	21.67	63.64	77.27	13.63
		ROW210	50	92.45	42.45	63.64	77.27	13.63
		ROW240	50	96.11	46.11	63.64	81.82	18.18
		ROW270	50	**97.65**	47.65	63.64	77.27	13.63
	54	RO	51.67	67.14	15.47	50	77.27	27.27
		ROW210	51.67	90.95	39.28	50	68.18	18.18
		ROW240	51.67	94.82	43.15	50	68.18	18.18
		ROW270	51.67	**97.98**	46.31	50	86.36	36.36

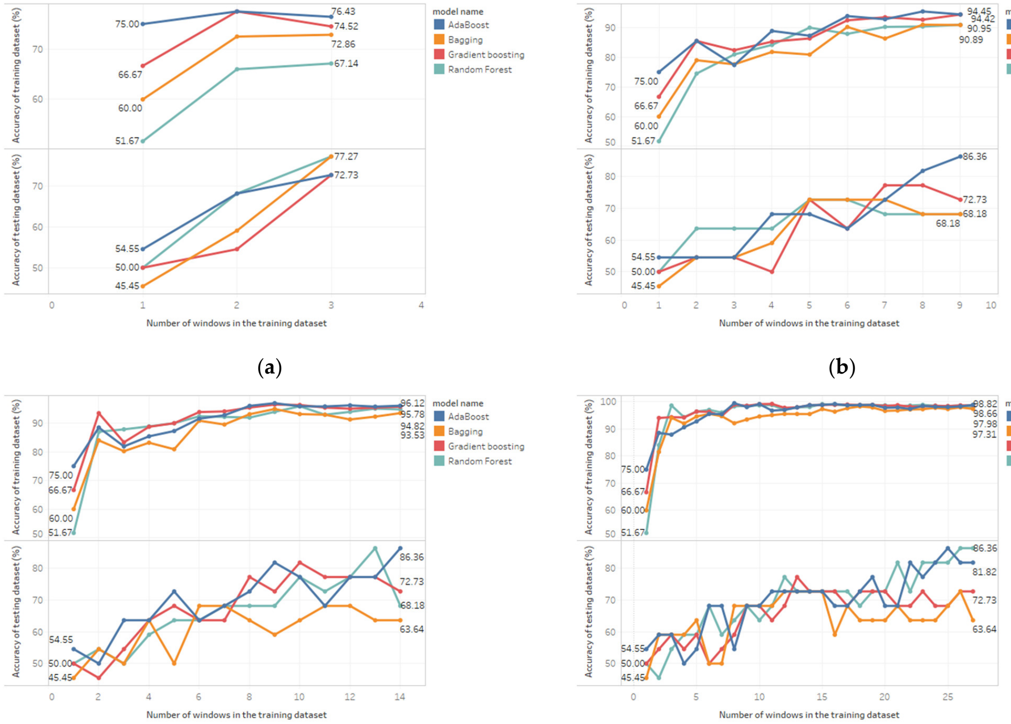
The NoR scenario was not further investigated because it was the only one that did not resample the training dataset. Besides that, the driving fatigue detection framework using the NoR scenario had the lowest model performance, resulting in an accuracy ranging from 55% with random forest and 54 features to 71.67% with AdaBoost and 29 features on the training dataset, as shown in Table 9.



(**a**) (**b**)

(**c**) (**d**)

Figure 6. The accuracy of the 4 classification models on training and testing datasets with 29 features using resampling scenario (**a**) RO; (**b**) ROW210; (**c**) ROW240; (**d**) ROW270.



(**a**) (**b**)

(**c**) (**d**)

Figure 7. The accuracy of the 4 classification models on training and testing datasets with 54 features using resampling scenario (**a**) RO; (**b**) ROW210; (**c**) ROW240; (**d**) ROW270.

Table 9. Summary of performance metrics using all windows in the training dataset and testing dataset for all 5 resampling scenarios with 29 and 54 features applied to the 4 classification models. In each classifier with different features and resampling scenarios, the best result in terms of accuracy of the training dataset is shown in bold.

Classifier	Features	Resampling Scenario	Performance Metrics (%)						
			Training Dataset	Testing Dataset					
			Acc	Acc	F1 Score	Precision	Sensitivity	Specificity	AUC
AdaBoost	29	NoR	71.67	59.09	64	57.14	72.73	45.45	0.71
		RO	77.86	72.73	76.92	66.67	90.91	54.55	0.77
		ROW210	93.89	72.73	75	69.23	81.82	63.64	0.88
		ROW240	95.13	77.27	78.26	75	81.82	72.73	0.9
		ROW270	**97.98**	72.73	75	69.23	81.82	63.64	0.88
	54	NoR	56.67	68.18	63.16	75	54.55	81.82	0.68
		RO	76.43	72.73	76.92	66.67	90.91	54.55	0.81
		ROW210	94.45	86.36	86.96	83.33	90.91	81.82	0.9
		ROW240	96.12	86.36	85.71	90	81.82	90.91	0.89
		ROW270	**98.82**	81.82	81.82	81.82	81.82	81.82	0.9
Bagging	29	NoR	56.67	72.73	75	69.23	81.82	63.64	0.83
		RO	66.67	77.27	80	71.43	90.91	63.64	0.86
		ROW210	92.42	68.18	69.57	66.67	72.73	63.64	0.85
		ROW240	94.16	72.73	72.73	72.73	72.73	72.73	0.86
		ROW270	**97.31**	72.73	72.73	72.73	72.73	72.73	0.86
	54	NoR	61.67	68.18	69.57	66.67	72.73	63.64	0.76
		RO	72.86	77.27	80	71.43	90.91	63.64	0.76
		ROW210	90.89	68.18	69.57	66.67	72.73	63.64	0.81
		ROW240	93.53	63.64	63.64	63.64	63.64	63.64	0.84
		ROW270	**97.31**	63.64	63.64	63.64	63.64	63.64	0.84
Gradient boosting	29	NoR	66.67	63.64	66.67	61.54	72.73	54.55	0.64
		RO	74.29	86.36	86.96	83.33	90.91	81.82	0.91
		ROW210	93.42	72.73	75	69.23	81.82	63.64	0.9
		ROW240	96.42	77.27	78.26	75	81.82	72.73	0.83
		ROW270	**98.99**	77.27	78.26	75	81.82	72.73	0.85
	54	NoR	61.67	68.18	66.67	70	63.64	72.73	0.84
		RO	74.52	72.73	76.92	66.67	90.91	54.55	0.72
		ROW210	94.42	72.73	75	69.23	81.82	63.64	0.86
		ROW240	95.78	72.73	75	69.23	81.82	63.64	0.88
		ROW270	**98.66**	72.73	75	69.23	81.82	63.64	0.87
Random forest	29	NoR	55	72.73	72.73	72.73	72.73	72.73	0.78
		RO	71.67	77.27	80	71.43	90.91	63.64	0.84
		ROW210	92.45	77.27	80	71.43	90.91	63.64	0.87
		ROW240	96.11	81.82	83.33	76.92	90.91	72.73	0.9
		ROW270	**97.65**	77.27	78.26	75	81.82	72.73	0.9
	54	NoR	55	68.18	66.67	70	63.64	72.73	0.81
		RO	67.14	77.27	78.26	75	81.82	72.73	0.88
		ROW210	90.95	68.18	72	64.29	81.82	54.55	0.86
		ROW240	94.82	68.18	66.67	70	63.64	72.73	0.82
		ROW270	**97.98**	86.36	86.96	83.33	90.91	81.82	0.96

According to the accuracy results illustrated in Figures 6 and 7, it is clear that the number of windows in the training dataset affected the accuracy of each model in both the training and testing datasets. More windows in the training dataset used in the driving fatigue detection framework tend to increase the accuracy of the classification model on both the training and testing datasets. For example, as shown in Figure 6b, the driving fatigue detection framework using the AdaBoost classifier, twenty-nine features, and the ROW210 scenario results in an accuracy of 73.33% with one window, which then increases to 93.89% with nine windows on the training dataset. As another example, as shown in Figure 7d, using a random forest classifier and fifty-four features, ROW270, results in an accuracy of 51.57% with one window, then increasing to 98.66% with twenty-seven windows on the training dataset. As shown in Figures 6 and 7, and Table 8, almost all of the 4 resampling scenarios with 29 and 54 features show an increase in accuracy on both the training and testing datasets, except the driving fatigue detection framework using the bagging classifier, 29 features, and RO scenario, which show a decrease in accuracy from 68.3% to 66.67% on the training dataset.

In Table 8, the driving fatigue detection framework, using the ROW270 scenario (all windows) and random forest classifier, produced the largest increase in accuracy: 47.65% for 29 features and 46.31% for 54 features on the training dataset. It also affected the increase in accuracy: 13.63% for 29 features and 36.36% for 54 features on the testing dataset. In contrast, the driving fatigue detection framework using the RO scenario (all windows) and AdaBoost classifier produced the smallest increase in accuracy: 4.53% for 29 features and 1.43% for 54 features on the training dataset. This shows that the use of all the windows in the training dataset for model training likely produced the highest accuracy for each model on both the training and testing datasets.

Moreover, if we examine and compare the accuracy results of all four resampling scenarios in Table 8, we find that the ROW270 scenario (all windows) has the largest impact on the increase in accuracy of the training dataset, starting from 23.82% using AdaBoost and 54 features to 47.65% using random forest and 29 features. The resampling method works very well on the random forest classifier.

Last, it can be observed that a resampling method with a longer overlapping window significantly affects the accuracy of the trained model because the resampling method can improve diversity. This is confirmed by the studies [62,63], which show that an optimal ensemble learning method depends on the diversity of each learner. Diversity can be enhanced by dividing the original dataset into smaller subsets of data. This can be performed with the resampling method applied in the proposed driving fatigue detection framework.

4.2. The Effect of 29 and 54 Features on the Model's Performance

In this subsection, we discuss the results shown in Table 8 and determine whether feature extraction or resampling methods had a larger impact on the model's performance. There are 2 scenarios of feature usage in the proposed driving fatigue detection framework to be assessed: 29 and 54 features. The twenty-nine features are the total combination of the features extracted from the time and frequency domain analysis, as shown in Table 4. These features are the most commonly used in biomedical applications for heart rate variability analysis. In contrast, the fifty-four features are the total combined features extracted from time domain analysis, frequency domain analysis, and nonlinear analysis, as shown in Tables 4 and 5. We focused on the accuracy of the results using the ROW270 scenario and all windows in the training dataset.

The driving fatigue detection framework using the random forest classifier and ROW270 (all windows) resulted in an accuracy of 97.65% with 29 features and 97.98% with 54 features on the training dataset. There was a small increase of 0.33% in accuracy. Compared to the driving fatigue detection framework using the bagging classifier and ROW270 (all windows), there was no change in accuracy between 29 and 54 features on the training dataset. This occurred because the random forest employs a "randomized" decision tree approach to assess subsets of features for splitting, while bagging assesses all features for splitting using a "deterministic" decision tree approach [62]. It is clear that using nonlinear analysis features improved the performance of the random forest model.

Furthermore, the usage of feature extraction methods with a nonlinear analysis approach improved the model's performance of AdaBoost more than that of the random forest in the proposed driving fatigue detection framework. AdaBoost achieved an accuracy of 97.98% with 29 features and 98.82% with 54 features on the training dataset. This was an increase of 0.84% in accuracy, larger than that of random forest.

Testing the trained model on unseen data or the testing dataset is another way to evaluate the benefit of nonlinear analysis. Some classifiers showed a greater effect of nonlinear analysis on the accuracy of the testing dataset than on that of the training dataset. For example, the driving fatigue detection framework using a random forest and

ROW270 (all windows) resulted in an accuracy of 77.27% with 29 features and 86.36% with 54 features on the testing dataset. This was an increase of 9.09% in accuracy on the testing dataset. Another example is the driving fatigue detection framework using AdaBoost and ROW270 (all windows), which resulted in an accuracy of 72.73% with 29 features and 81.82% with 54 features on the testing dataset. It had an increase of 9.09% in accuracy on the testing dataset.

However, using nonlinear analysis did not work well with bagging and gradient boosting classifiers in ROW270 because the classifiers showed decreases in the accuracy of 72.73% to 63.64% and 77.27% to 72.73%, respectively, on the testing dataset. This result possibly occurred because one or more features of nonlinear analysis were irrelevant to the model's performance of the bagging and gradient classifiers. This type of feature is called a redundant feature, which may represent more noisy information than useful information [90].

By comparing the effects of using 29 or 54 features in the same resampling scenario to the effects of using the same number of features (29 or 54) in different resampling scenarios, it can be seen that the resampling method has a greater impact on the model's performance than the feature extraction method.

4.3. Model Selection Considerations

According to the analysis of resampling methods on the model's performance, the ROW270 (all windows) scenario generally had the greatest impact on the accuracy of the training datasets. Therefore, we focused on analyzing the accuracy results using the ROW270 (all windows) scenario to select the model. Table 9 shows the overall performance metrics using all the windows in the training dataset for training the model. The gradient boosting classifier seemed to be the best model because its classifier resulted in the highest accuracy: 98.99% for 29 features and 98.66% for 54 features on the training dataset. However, the actual performance of gradient boosting on the testing dataset resulted in an accuracy of 77.27% for 29 features and 72.73% for 54 features. Therefore, there was a difference of 21.72% in accuracy for 29 features and 25.93% in accuracy for 54 features. It can be concluded that gradient boosting has low generalizability over unseen data.

Furthermore, we searched for a model that produced the second-highest accuracy on the training dataset in Table 9 and found AdaBoost, which resulted in an accuracy of 97.98% for 29 features and 98.82% for 54 features. The actual performance of AdaBoost resulted in an accuracy of 72.73% for 29 features and 81.82% for 54 features. Therefore, there was a difference of 25.25% in accuracy for 29 features and 17% in accuracy for 54 features. AdaBoost, with 54 features, seems to have better generalization than gradient boosting over unseen data.

The third-highest accuracy on the training dataset in Table 9 is that of the random forest classifier, which resulted in an accuracy of 97.65% for 29 features and 97.98% for 54 features. The actual performance of the random forest resulted in an accuracy of 77.27% for 29 features and 86.36% for 54 features. Therefore, it has a difference of 20.38% in accuracy for 29 features and 11.62% in accuracy for 54 features. Random forest, an extension of the bagging technique, focuses on variance reduction, whereas AdaBoost, a boosting technique, focuses on bias reduction [62]. This explains why the random forest has a smaller accuracy difference between the training and testing datasets than AdaBoost. In contrast, AdaBoost outperformed random forest on the training dataset.

The remaining model is the bagging classifier. We did not evaluate this model because it was the least accurate. In addition, random forest, a bagging algorithm extension, outperformed the bagging classifier.

Other performance metrics need to be considered when choosing the optimal model: the area under the curve (AUC), F1 score, precision, sensitivity, and specificity, as shown in Table 9. Random forest with 54 features showed a better accuracy on the testing dataset than AdaBoost, and its accuracy also affected the other performance metrics, AUC, F1 score, precision, sensitivity, and specificity: 0.96, 86.36%, 83,33%, 90.91%, and 81.82%, respectively.

Random forest had better generalization than AdaBoost when tested on unseen or future data, but it was only tested on 1 window or 300 s of NN interval data for each participant. It needs to be tested on unseen data with more participants.

Finally, we chose 2 optimal classification models that worked well in the proposed driving fatigue detection framework: random forest and AdaBoost with ROW270 (all windows) and 54 feature scenarios because both classifiers showed very good results on both the training and testing datasets. Our proposed driving fatigue detection framework is compared with driving fatigue detection studies, as shown in Table 1. We selected driving fatigue detection studies whose datasets included real or virtual driving and produced an accuracy of more than 90%, or 0.9, in the AUC metric. The comparison results are shown in Table 10.

Table 10. Comparison of the proposed driving fatigue detection framework with previous driving fatigue detection studies.

Source	Number of Participants	Record. Time	Measurement	Features	Classification	Accuracy [1]

[28]	1st:18;
	2nd:24;
	3rd:44
		90 min	EEG, ECG, EOG, and vehicle data	54	Random forest	94.1

[30]	6	67 min	ECG	12	SVM	0.95 (AUC)
[32]	25	80 min	ECG	24	Ensemble logistic regression	92.5
[33]	47	30 min	ECG and vehicle data	49	Random forest	91.2
[11]	16	30 min	EEG, ECG, and vehicle data	80	Random forest	95.4
[15]	9	>10 min	EDA, RESP, and PPG	15	ANN, backpropagation neural network (BPNN), cascade forward neural network (CFNN)	97.9
[38]	20	20 min	EEG and ECG	Product fuzzy convolutional network (PFCN)		94.19
Ours	11	30 min	ECG	54 features, resampling with overlapping windows ($T_w = 300$ s, $T_o = 270$ s)	Random forest AdaBoost	97.98 [1] 86.36 [2] 98.82 [1] 81.82 [2]

[1] The accuracy of training data; [2] The accuracy of testing data

It can be concluded that the proposed driving fatigue detection framework using ECG alone can yield a higher-accuracy model in fatigue detection than driving fatigue detection studies using more than one physiological sensor.

4.4. Future Work Developments

The proposed driving fatigue detection framework demonstrated the highest accuracy level among previous driving fatigue detection studies. However, attaching the two required ECG Biosemi Active electrodes [18] to the participant's chest for detection of the fatigue state is impractical in real applications because it could interfere with driving, and the driver's movement could create artifacts in the ECG-recorded signals, consequently reducing the accuracy of driving fatigue detection. Future real-world driving fatigue detection applications will need a heart rate measuring device that is easy to use, does not disturb the driver, and has a high level of accuracy in detecting R waves.

Most driving fatigue detection studies used a heart rate variability analysis approach to extract features from NN interval data, such as Huang et al. [15], Awais et al. [27], Mårtensson et al. [28], Lei et al. [29], Kim and Shin [30], Babaeian and Mozumdar [32], Arefnezhad et al. [33], Papakostas et al. [35], Hasan et al. [37]. Our proposed method also used the same approach. There are still possible areas for improvement in feature engineering for future work on driving fatigue detection. In the future, we are considering using another approach to extract features from raw ECG signals, for example, statistical features: feature-based information retrieval with a self-similarity matrix [91], morphological features: Gaussian with a synthesized mathematical model [92], and wavelet features: wavelet transform [93].

In this study, the ECG signal of the 11th participant in the SG condition was recorded up to 23.5 min, so to have a balanced dataset, 23 min out of 30 min of ECG recordings were extracted from the dataset of all participants. As a result, only one window (≈300 s) of NN interval data was used for the testing dataset, calculating 22% of the total NN interval data. Therefore, we cannot further evaluate the performance of each classification model based on unseen or future data. The proposed driving fatigue detection framework needs to be evaluated with more participants and a longer-duration driving fatigue dataset for future improvements in driving fatigue detection.

5. Conclusions

Our proposed driving fatigue detection framework utilizing ECG alone demonstrated a higher accuracy than previous driving fatigue detection studies utilizing multiple physiological sensors. Moreover, the resampling method in the preprocessing step had the greatest impact on the model's performance, especially with the random forest classifier. Adding nonlinear analysis features also improved the model's performance on the random forest and AdaBoost classifiers. Most driving fatigue detection studies evaluated the model's performance using the training dataset only or performed cross-validation. This paper shows that testing the trained model on unseen data can be an effective tool for further investigating the model's generalizability. Moreover, it can be used to analyze the effect on the model's performance of utilizing nonlinear analysis features in the driving fatigue detection framework.

Author Contributions: Conceptualization, J.H. and K.R.; methodology, J.H.; software, J.H.; validation, K.R., D.S. and T.S.G.; formal analysis, J.H.; investigation, J.H., K.R., D.S. and T.S.G.; resources, K.R. and D.S.; data curation, J.H.; writing—original draft preparation, J.H.; writing—review and editing, J.H., K.R., D.S. and T.S.G.; visualization, J.H.; supervision, K.R.; project administration, M.S.; funding acquisition, M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Universitas Indonesia through the Hibah Publikasi Terindeks Internasional (PUTI) Q2 Scheme (Grant no. NKB-701/UN2.RST/HKP.05.00/2022).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Restrictions apply to the availability of the dataset. The dataset used in this work was obtained from Deep BCI, Korea, to which any request for data should be addressed (kyunghowon0712@gist.ac.kr).

Acknowledgments: The authors would like to thank Deep BCI, Institute of Information & Communications Technology Planning & Evaluation (IITP), Korea, for providing the dataset.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. WHO. Road Traffic Injuries. 2022. Available online: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries (accessed on 22 September 2022).
2. Chand, A.; Jayesh, S.; Bhasi, A. Road traffic accidents: An overview of data sources, analysis techniques and contributing factors. *Mater. Today Proc.* **2021**, *47*, 5135–5141. [CrossRef]
3. Razzaghi, A.; Soori, H.; Kavousi, A.; Abadi, A.; Khosravi, A.K.; Alipour, A. Risk factors of deaths related to road traffic crashes in World Health Organization regions: A systematic review. *Arch. Trauma Res.* **2019**, *8*, 57–86.
4. Bucsuházy, K.; Matuchová, E.; Zůvala, R.; Moravcová, P.; Kostíková, M.; Mikulec, R. Human factors contributing to the road traffic accident occurrence. *Transp. Res. Procedia* **2020**, *45*, 555–561. [CrossRef]
5. Smith, A.P. A UK survey of driving behaviour, fatigue, risk taking and road traffic accidents. *BMJ Open* **2016**, *6*, e011461. [CrossRef]
6. Albadawi, Y.; Takruri, M.; Awad, M. A review of recent developments in driver drowsiness detection systems. *Sensors* **2022**, *22*, 2069. [CrossRef]

7. Khunpisuth, O.; Chotchinasri, T.; Koschakosai, V.; Hnoohom, N. Driver drowsiness detection using eye-closeness detection. In Proceedings of the 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Naples, Italy, 28 November–1 December 2016; pp. 661–668.
8. Khare, S.K.; Bajaj, V. Entropy-Based Drowsiness Detection Using Adaptive Variational Mode Decomposition. *IEEE Sens. J.* **2021**, *21*, 6421–6428. [CrossRef]
9. Khushaba, R.N.; Kodagoda, S.; Lal, S.; Dissanayake, G. Driver drowsiness classification using fuzzy wavelet-packet-based feature-extraction algorithm. *IEEE Trans. Biomed. Eng.* **2010**, *58*, 121–131. [CrossRef]
10. Babaeian, M.; Amal Francis, K.; Dajani, K.; Mozumdar, M. Real-time driver drowsiness detection using wavelet transform and ensemble logistic regression. *Int. J. Intell. Transp. Syst. Res.* **2019**, *17*, 212–222. [CrossRef]
11. Gwak, J.; Hirao, A.; Shino, M. An investigation of early detection of driver drowsiness using ensemble machine learning based on hybrid sensing. *Appl. Sci.* **2020**, *10*, 2890. [CrossRef]
12. Singhal, S.; Jena, M. A study on WEKA tool for data preprocessing, classification and clustering. *Int. J. Innov. Technol. Explor. Eng. (IJITEE)* **2013**, *2*, 250–253.
13. Benhar, H.; Idri, A.; Fernández-Alemán, J. Data preprocessing for heart disease classification: A systematic literature review. *Comput. Methods Programs Biomed.* **2020**, *195*, 105635. [CrossRef] [PubMed]
14. Chandrasekar, P.; Qian, K. The impact of data preprocessing on the performance of a naive bayes classifier. In Proceedings of the 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), Atlanta, GA, USA, 10–14 June 2016; pp. 618–619.
15. Huang, Y.; Deng, Y. A Hybrid Model Utilizing Principal Component Analysis and Artificial Neural Networks for Driving Drowsiness Detection. *Appl. Sci.* **2022**, *12*, 6007. [CrossRef]
16. Shi, Z.; He, L.; Suzuki, K.; Nakamura, T.; Itoh, H. Survey on neural networks used for medical image processing. *Int. J. Comput. Sci.* **2009**, *3*, 86. [PubMed]
17. Noguerol, T.M.; Paulano-Godino, F.; Martín-Valdivia, M.T.; Menias, C.O.; Luna, A. Strengths, weaknesses, opportunities, and threats analysis of artificial intelligence and machine learning applications in radiology. *J. Am. Coll. Radiol.* **2019**, *16*, 1239–1247. [CrossRef] [PubMed]
18. Ahn, S.; Nguyen, T.; Jang, H.; Kim, J.G.; Jun, S.C. Exploring neuro-physiological correlates of drivers' mental fatigue caused by sleep deprivation using simultaneous EEG, ECG, and fNIRS data. *Front. Hum. Neurosci.* **2016**, *10*, 219. [CrossRef]
19. Bier, L.; Wolf, P.; Hilsenbek, H.; Abendroth, B. How to measure monotony-related fatigue? A systematic review of fatigue measurement methods for use on driving tests. *Theor. Issues Ergon. Sci.* **2020**, *21*, 22–55. [CrossRef]
20. May, J.F.; Baldwin, C.L. Driver fatigue: The importance of identifying causal factors of fatigue when considering detection and countermeasure technologies. *Transp. Res. Part F Traffic Psychol. Behav.* **2009**, *12*, 218–224. [CrossRef]
21. Johns, M.W. A new method for measuring daytime sleepiness: The Epworth sleepiness scale. *Sleep* **1991**, *14*, 540–545. [CrossRef]
22. Samn, S.W.; Perelli, L.P. *Estimating Aircrew Fatigue: A Technique with Application to Airlift Operations*; School of Aerospace Medicine: Brooks AFB, TX, USA, 1982.
23. Shahid, A.; Wilkinson, K.; Marcu, S.; Shapiro, C.M. Stanford sleepiness scale (SSS). In *Stop, That and One Hundred Other Sleep Scales*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 369–370.
24. Åkerstedt, T.; Gillberg, M. Subjective and objective sleepiness in the active individual. *Int. J. Neurosci.* **1990**, *52*, 29–37. [CrossRef]
25. Cella, M.; Chalder, T. Measuring fatigue in clinical and community settings. *J. Psychosom. Res.* **2010**, *69*, 17–22. [CrossRef]
26. Ramzan, M.; Khan, H.U.; Awan, S.M.; Ismail, A.; Ilyas, M.; Mahmood, A. A survey on state-of-the-art drowsiness detection techniques. *IEEE Access* **2019**, *7*, 61904–61919. [CrossRef]
27. Awais, M.; Badruddin, N.; Drieberg, M. A hybrid approach to detect driver drowsiness utilizing physiological signals to improve system performance and wearability. *Sensors* **2017**, *17*, 1991. [CrossRef] [PubMed]
28. Mårtensson, H.; Keelan, O.; Ahlström, C. Driver sleepiness classification based on physiological data and driving performance from real road driving. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 421–430. [CrossRef]
29. Lei, J.; Liu, F.; Han, Q.; Tang, Y.; Zeng, L.; Chen, M.; Ye, L.; Jin, L. Study on driving fatigue evaluation system based on short time period ECG signal. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2466–2470.
30. Kim, J.; Shin, M. Utilizing HRV-derived respiration measures for driver drowsiness detection. *Electronics* **2019**, *8*, 669. [CrossRef]
31. Lee, H.; Lee, J.; Shin, M. Using wearable ECG/PPG sensors for driver drowsiness detection based on distinguishable pattern of recurrence plots. *Electronics* **2019**, *8*, 192. [CrossRef]
32. Babaeian, M.; Mozumdar, M. Driver drowsiness detection algorithms using electrocardiogram data analysis. In Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 7–9 January 2019; pp. 0001–0006.
33. Arefnezhad, S.; Eichberger, A.; Frühwirth, M.; Kaufmann, C.; Moser, M. Driver Drowsiness Classification Using Data Fusion of Vehicle-based Measures and ECG Signals. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 11–14 October 2020; pp. 451–456.
34. Peng, Z.; Rong, J.; Wu, Y.; Zhou, C.; Yuan, Y.; Shao, X. Exploring the different patterns for generation process of driving fatigue based on individual driving behavior parameters. *Transp. Res. Rec.* **2021**, *2675*, 408–421. [CrossRef]

35. Papakostas, M.; Das, K.; Abouelenien, M.; Mihalcea, R.; Burzo, M. Distracted and drowsy driving modeling using deep physiological representations and multitask learning. *Appl. Sci.* **2020**, *11*, 88. [CrossRef]
36. Chui, K.T.; Lytras, M.D.; Liu, R.W. A generic design of driver drowsiness and stress recognition using MOGA optimized deep MKL-SVM. *Sensors* **2020**, *20*, 1474. [CrossRef]
37. Hasan, M.M.; Watling, C.N.; Larue, G.S. Physiological signal-based drowsiness detection using machine learning: Singular and hybrid signal approaches. *J. Saf. Res.* **2022**, *80*, 215–225. [CrossRef]
38. Du, G.; Long, S.; Li, C.; Wang, Z.; Liu, P.X. A Product Fuzzy Convolutional Network for Detecting Driving Fatigue. *IEEE Trans. Cybern.* **2022**, 1–14. [CrossRef]
39. Rather, A.A.; Sofi, T.A.; Mukhtar, N. A Survey on Fatigue and Drowsiness Detection Techniques in Driving. In Proceedings of the 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 19–20 February 2021; pp. 239–244.
40. Fujiwara, K.; Abe, E.; Kamata, K.; Nakayama, C.; Suzuki, Y.; Yamakawa, T.; Hiraoka, T.; Kano, M.; Sumi, Y.; Masuda, F. Heart rate variability-based driver drowsiness detection and its validation with EEG. *IEEE Trans. Biomed. Eng.* **2018**, *66*, 1769–1778. [CrossRef] [PubMed]
41. Hendra, M.; Kurniawan, D.; Chrismiantari, R.V.; Utomo, T.P.; Nuryani, N. Drowsiness detection using heart rate variability analysis based on microcontroller unit. *Proc. J. Phys. Conf. Ser.* **2019**, *1153*, 012047. [CrossRef]
42. Halomoan, J.; Ramli, K.; Sudiana, D. Statistical analysis to determine the ground truth of fatigue driving state using ECG Recording and subjective reporting. In Proceedings of the 2020 1st International Conference on Information Technology, Advanced Mechanical and Electrical Engineering (ICITAMEE), Yogyakarta, Indonesia, 13–14 October 2020; pp. 244–248.
43. 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]
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*, 537–554. [CrossRef]
45. Sahayadhas, A.; Sundaraj, K.; Murugappan, M. Detecting driver drowsiness based on sensors: A review. *Sensors* **2012**, *12*, 16937–16953. [CrossRef]
46. Kwon, O.; Jeong, J.; Kim, H.B.; Kwon, I.H.; Park, S.Y.; Kim, J.E.; Choi, Y. Electrocardiogram sampling frequency range acceptable for heart rate variability analysis. *Healthc. Inform. Res.* **2018**, *24*, 198–206. [CrossRef]
47. Shaffer, F.; Ginsberg, J.P. An overview of heart rate variability metrics and norms. *Front. Public Health* **2017**, *5*, 258. [CrossRef]
48. Oweis, R.J.; Al-Tabbaa, B.O. QRS detection and heart rate variability analysis: A survey. *Biomed. Sci. Eng.* **2014**, *2*, 13–34.
49. Fariha, M.; Ikeura, R.; Hayakawa, S.; Tsutsumi, S. Analysis of Pan-Tompkins algorithm performance with noisy ECG signals. *Proc. J. Phys. Conf. Ser.* **2020**, *1532*, 012022. [CrossRef]
50. Pan, J.; Tompkins, W.J. A real-time QRS detection algorithm. *IEEE Trans. Biomed. Eng.* **1985**, *BME-32*, 230–236. [CrossRef]
51. Wang, L.; Lin, Y.; Wang, J. A RR interval based automated apnea detection approach using residual network. *Comput. Methods Programs Biomed.* **2019**, *176*, 93–104. [CrossRef] [PubMed]
52. Van, R.G.; Drake, F. Python 3 reference manual. *Scotts Val. CA Creat.* **2009**, *10*, 1593511.
53. McKinney, W. Pandas: A foundational Python library for data analysis and statistics. *Python High Perform. Sci. Comput.* **2011**, *14*, 1–9.
54. Van Der Walt, S.; Colbert, S.C.; Varoquaux, G. The NumPy array: A structure for efficient numerical computation. *Comput. Sci. Eng.* **2011**, *13*, 22–30. [CrossRef]
55. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
56. Raybaut, P. Spyder-Documentation. 2009. Available online: https://www.spyder-ide.org/ (accessed on 22 September 2022).
57. Arlot, S.; Celisse, A. A survey of cross-validation procedures for model selection. *Stat. Surv.* **2010**, *4*, 40–79. [CrossRef]
58. Meng, Z.; McCreadie, R.; Macdonald, C.; Ounis, I. Exploring data splitting strategies for the evaluation of recommendation models. In Proceedings of the Fourteenth ACM Conference on Recommender Systems, Online, 22–26 September 2020; pp. 681–686.
59. Vilette, C.; Bonnell, T.; Henzi, P.; Barrett, L. Comparing dominance hierarchy methods using a data-splitting approach with real-world data. *Behav. Ecol.* **2020**, *31*, 1379–1390. [CrossRef]
60. Malik, M. Heart rate variability: Standards of measurement, physiological interpretation, and clinical use: Task force of the European Society of Cardiology and the North American Society for Pacing and Electrophysiology. *Ann. Noninvasive Electrocardiol.* **1996**, *1*, 151–181. [CrossRef]
61. El-Amir, H.; Hamdy, M. Data Resampling. In *Deep Learning Pipeline*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 207–231.
62. Zhou, Z.-H. Ensemble learning. In *Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 181–210.
63. Sagi, O.; Rokach, L. Ensemble learning: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1249. [CrossRef]
64. Rahi, P.K.; Mehra, R. Analysis of power spectrum estimation using welch method for various window techniques. *Int. J. Emerg. Technol. Eng.* **2014**, *2*, 106–109.

65. Fell, J.; Röschke, J.; Mann, K.; Schäffner, C. Discrimination of sleep stages: A comparison between spectral and nonlinear EEG measures. *Electroencephalogr. Clin. Neurophysiol.* **1996**, *98*, 401–410. [CrossRef] [PubMed]
66. Makowski, D.; Pham, T.; Lau, Z.J.; Brammer, J.C.; Lespinasse, F.; Pham, H.; Schölzel, C.; Chen, S. NeuroKit2: A Python toolbox for neurophysiological signal processing. *Behav. Res. Methods* **2021**, *53*, 1689–1696. [CrossRef]
67. Mourot, L.; Bouhaddi, M.; Perrey, S.; Cappelle, S.; Henriet, M.T.; Wolf, J.P.; Rouillon, J.D.; Regnard, J. Decrease in heart rate variability with overtraining: Assessment by the Poincare plot analysis. *Clin. Physiol. Funct. Imaging* **2004**, *24*, 10–18. [CrossRef]
68. Zeng, C.; Wang, W.; Chen, C.; Zhang, C.; Cheng, B. Poincaré plot indices of heart rate variability for monitoring driving fatigue. In Proceedings of the 19th COTA International Conference of Transportation Professionals (CICTP), Nanjing, China, 6–8 July 2019; pp. 652–660.
69. Guo, W.; Xu, C.; Tan, J.; Li, Y. Review and implementation of driving fatigue evaluation methods based on RR interval. In Proceedings of the International Conference on Green Intelligent Transportation System and Safety, Changchun, China, 1–2 July 2017; pp. 833–843.
70. Hsu, C.-H.; Tsai, M.-Y.; Huang, G.-S.; Lin, T.-C.; Chen, K.-P.; Ho, S.-T.; Shyu, L.-Y.; Li, C.-Y. Poincaré plot indexes of heart rate variability detect dynamic autonomic modulation during general anesthesia induction. *Acta Anaesthesiol. Taiwanica* **2012**, *50*, 12–18. [CrossRef] [PubMed]
71. Tulppo, M.P.; Makikallio, T.H.; Takala, T.; Seppanen, T.; Huikuri, H.V. Quantitative beat-to-beat analysis of heart rate dynamics during exercise. *Am. J. Physiol.-Heart Circ. Physiol.* **1996**, *271*, H244–H252. [CrossRef] [PubMed]
72. Roy, R.; Venkatasubramanian, K. EKG/ECG based driver alert system for long haul drive. *Indian J. Sci. Technol.* **2015**, *8*, 8–13. [CrossRef]
73. Mohanavelu, K.; Lamshe, R.; Poonguzhali, S.; Adalarasu, K.; Jagannath, M. Assessment of human fatigue during physical performance using physiological signals: A review. *Biomed. Pharmacol. J.* **2017**, *10*, 1887–1896. [CrossRef]
74. Talebinejad, M.; Chan, A.D.; Miri, A. Fatigue estimation using a novel multi-fractal detrended fluctuation analysis-based approach. *J. Electromyogr. Kinesiol.* **2010**, *20*, 433–439. [CrossRef]
75. Wang, F.; Wang, H.; Zhou, X.; Fu, R. Study on the effect of judgment excitation mode to relieve driving fatigue based on MF-DFA. *Brain Sci.* **2022**, *12*, 1199. [CrossRef]
76. Rogers, B.; Mourot, L.; Doucende, G.; Gronwald, T. Fractal correlation properties of heart rate variability as a biomarker of endurance exercise fatigue in ultramarathon runners. *Physiol. Rep.* **2021**, *9*, e14956. [CrossRef]
77. Wang, F.; Wang, H.; Zhou, X.; Fu, R. A Driving Fatigue Feature Detection Method Based on Multifractal Theory. *IEEE Sens. J.* **2022**, *22*, 19046–19059. [CrossRef]
78. Kantelhardt, J.W.; Zschiegner, S.A.; Koscielny-Bunde, E.; Havlin, S.; Bunde, A.; Stanley, H.E. Multifractal detrended fluctuation analysis of nonstationary time series. *Phys. A Stat. Mech. Appl.* **2002**, *316*, 87–114. [CrossRef]
79. Jeppesen, J.; Beniczky, S.; Johansen, P.; Sidenius, P.; Fuglsang-Frederiksen, A. Detection of epileptic seizures with a modified heart rate variability algorithm based on Lorenz plot. *Seizure* **2015**, *24*, 1–7. [CrossRef] [PubMed]
80. Ihlen, E.A. Introduction to multifractal detrended fluctuation analysis in Matlab. *Front. Physiol.* **2012**, *3*, 141. [CrossRef]
81. Gholami, R.; Fakhari, N. Support vector machine: Principles, parameters, and applications. In *Handbook of Neural Computation*; Elsevier: Amsterdam, The Netherlands, 2017; pp. 515–535.
82. Ansari, A.; Bakar, A.A. A comparative study of three artificial intelligence techniques: Genetic algorithm, neural network, and fuzzy logic, on scheduling problem. In Proceedings of the 2014 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology, Sabah, Malaysia, 3–5 December 2014; pp. 31–36.
83. Emanet, N.; Öz, H.R.; Bayram, N.; Delen, D. A comparative analysis of machine learning methods for classification type decision problems in healthcare. *Decis. Anal.* **2014**, *1*, 6. [CrossRef]
84. González, S.; García, S.; Del Ser, J.; Rokach, L.; Herrera, F. A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities. *Inf. Fusion* **2020**, *64*, 205–237. [CrossRef]
85. Reitermanova, Z. Data splitting. In Proceedings of the WDS, Prague, Czech Republic, 1–4 June 2010; pp. 31–36.
86. Xu, Y.; Goodacre, R. On splitting training and validation set: A comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. *J. Anal. Test.* **2018**, *2*, 249–262. [CrossRef]
87. Refaeilzadeh, P.; Tang, L.; Liu, H. Cross-validation. *Encycl. Database Syst.* **2009**, *5*, 532–538.
88. Yang, L.; Shami, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing* **2020**, *415*, 295–316. [CrossRef]
89. Andonie, R. Hyperparameter optimization in learning systems. *J. Membr. Comput.* **2019**, *1*, 279–291. [CrossRef]
90. Kotsiantis, S. Feature selection for machine learning classification problems: A recent overview. *Artif. Intell. Rev.* **2011**, *42*, 157–176. [CrossRef]
91. Rodrigues, J.; Liu, H.; Folgado, D.; Belo, D.; Schultz, T.; Gamboa, H. Feature-Based Information Retrieval of Multimodal Biosignals with a Self-Similarity Matrix: Focus on Automatic Segmentation. *Biosensors* **2022**, *12*, 1182. [CrossRef] [PubMed]

92. do Vale Madeiro, J.P.; Marques, J.A.L.; Han, T.; Pedrosa, R.C. Evaluation of mathematical models for QRS feature extraction and QRS morphology classification in ECG signals. *Measurement* **2020**, *156*, 107580. [CrossRef]
93. Khan, T.T.; Sultana, N.; Reza, R.B.; Mostafa, R. ECG feature extraction in temporal domain and detection of various heart conditions. In Proceedings of the 2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), Savar, Dhaka, Bangladesh, 21–23 May 2015; pp. 1–6.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.