

Article

A Recognition Method of Aggressive Driving Behavior Based on Ensemble Learning

Hanqing Wang¹, Xiaoyuan Wang^{1,2,*} , Junyan Han¹, Hui Xiang¹, Hao Li¹, Yang Zhang¹ and Shangqing Li¹

¹ College of Electromechanical Engineering, Qingdao University of Science & Technology, Qingdao 266000, China; 2019030020@mails.qust.edu.cn (H.W.); b021030001@mails.qust.edu.cn (J.H.); 2019030024@mails.qust.edu.cn (H.X.); 4019030012@mails.qust.edu.cn (H.L.); 4019030049@mails.qust.edu.cn (Y.Z.); 4019030017@mails.qust.edu.cn (S.L.)

² Collaborative Innovation Center for Intelligent Green Manufacturing Technology and Equipment of Shandong Province, Qingdao 266000, China

* Correspondence: wangxiaoyuan@qust.edu.cn; Tel.: +86-138-6445-5865

Abstract: Aggressive driving behavior (ADB) is one of the main causes of traffic accidents. The accurate recognition of ADB is the premise to timely and effectively conduct warning or intervention to the driver. There are some disadvantages, such as high miss rate and low accuracy, in the previous data-driven recognition methods of ADB, which are caused by the problems such as the improper processing of the dataset with imbalanced class distribution and one single classifier utilized. Aiming to deal with these disadvantages, an ensemble learning-based recognition method of ADB is proposed in this paper. First, the majority class in the dataset is grouped employing the self-organizing map (SOM) and then are combined with the minority class to construct multiple class balance datasets. Second, three deep learning methods, including convolutional neural networks (CNN), long short-term memory (LSTM), and gated recurrent unit (GRU), are employed to build the base classifiers for the class balance datasets. Finally, the ensemble classifiers are combined by the base classifiers according to 10 different rules, and then trained and verified using a multi-source naturalistic driving dataset acquired by the integrated experiment vehicle. The results suggest that in terms of the recognition of ADB, the ensemble learning method proposed in this research achieves better performance in accuracy, recall, and F₁-score than the aforementioned typical deep learning methods. Among the ensemble classifiers, the one based on the LSTM and the Product Rule has the optimal performance, and the other one based on the LSTM and the Sum Rule has the suboptimal performance.

Keywords: aggressive driving behavior; class imbalance dataset; ensemble learning; deep learning; advanced driver assistance system



Citation: Wang, H.; Wang, X.; Han, J.; Xiang, H.; Li, H.; Zhang, Y.; Li, S. A Recognition Method of Aggressive Driving Behavior Based on Ensemble Learning. *Sensors* **2022**, *22*, 644. <https://doi.org/10.3390/s22020644>

Academic Editor: Felipe Jiménez

Received: 28 November 2021

Accepted: 12 January 2022

Published: 14 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Traffic accidents have been around since Karl Benz invented the car. With the development of society and economy, the number of cars is increasing, which has led to the increase in traffic congestion and traffic accidents. The research suggests that more than 90% of traffic accidents are caused by human factors [1]. Among them, a survey of the AAA Foundation for Traffic Safety shows that about 55.7% of fatal traffic accidents were associated with aggressive driving behavior (ADB) [2], and there is a positive correlation between ADB and the probability of traffic accidents [3,4]. As one of the main causes of traffic accidents, ADB is affected by situational factors such as traffic congestion [5,6] and personal factors such as negative emotions [7]. Due to the increasingly crowded traffic system and the accelerated pace of life, it is easier for drivers to exhibit ADB, so it is urgent to accurately recognize ADB. However, there is no uniform definition of ADB. ADB was mostly defined from the perspective of traffic psychology in existing studies as a syndrome of frustration-driven instrumental behaviors, that is, deliberately dangerous

driving to save time at the expense of others [8]; the driving behavior that is likely to increase the risk of collision, and is motivated by impatience, annoyance, hostility, and/or an attempt to save time [9]; or any driving behavior that intentionally (whether fueled by anger or frustration or as a calculated means to an end) endangers others psychologically, physically, or both [10]. The above definition based on traffic psychology is beneficial for people to understand the causes of ADB, but it is difficult to be directly applied to the recognition of ADB. Therefore, for the accurate recognition of ADB, we define ADB as driving behaviors where a driver intentionally harms another driver in any form, which are typically manifested as abnormal acceleration, abnormal deceleration, abnormal lane change, and tailgating.

In recent years, some studies were conducted on the recognition of ADB, which can be divided into studies based on simulated driving datasets [11–14] and studies based on naturalistic driving datasets [15–23], according to the different datasets utilized. The simulated driving experiment is commonly used in studies on ADB due to its high level of safety. Wang et al. used a semi-supervised support vector machine to divide the driving style between aggressive driving style and normal driving style based on the vehicle dynamic parameters collected in simulated driving experiments [11]. Danaf et al. proposed a hybrid model for aggressive driving analysis and prediction based on the state-trait anger theory, which was verified using simulated driving experiment data [12]. Fitzpatrick et al. studied the influence of time pressure on ADB based on simulated driving experiments [13]. Kerwin et al. concluded that people with high trait anger tend to view many driving behaviors as aggressive, based on the ratings of the videos taken by 198 participants on a driving simulator [14]. Compared with the naturalistic driving experiment, the simulated driving experiment is safer and provides easier control of the experimental conditions. However, there is a certain difference between the data collected through the simulation driving experiment and those collected in the actual traffic environment, which may lead to the problems of low recognition accuracy and high miss rate when the relevant recognition methods are applied to the actual environment. With the development and popularization of vehicle sensor technology and computing platforms, the research on ADB based on naturalistic driving datasets has gradually increased. Ma et al. developed an online approach for aggressive driving recognition using the kinematic parameters that were collected by the in-vehicle recorder under naturalistic driving conditions [15]. Feng et al. verified the performance of the vehicle jerk for recognizing ADB through naturalistic driving data [16].

Both naturalistic driving and simulated driving can provide considerable datasets, which provide the conditions for the studies of ADB recognition based on data-driven methods. Compared with the theory-driven methods, the data-driven methods [17–26] are naturally suitable for accurately recognizing the complex behaviors in the actual environment because of their capacity to explore the inherent correlations of the captured data warning [27,28] and have been applied in the studies of ADB recognition [17–23]. Zylius used time and frequency domain features extracted from accelerometer data to build a random forest classifier to recognize aggressive driving styles [17]. Ma et al. used the vehicle motion data collected by the smartphone sensors to compare the recognition performance of the Gaussian mixture model, partial least squares regression, wavelet transformation, and support vector regression on ADB [18]. Carlos et al. used the bag of words method to extract features from accelerometer data and built the models of ADB recognition based on multilayer perceptron, random forest, naive Bayes classifier, and K-nearest neighbor algorithm [19]. Although the recognition of ADB can be realized based on the above methods, a large number of data preprocessing and feature engineering are required in the modeling of time series. In recent years, many deep learning methods have proved to be an effective solution to time series modeling due to their capacity to automatically learn the temporal dependencies present in time series [29]. These deep learning methods have already been applied in the research of ADB recognition [20–23]. Moukafih et al. proposed a recognition method of ADB based on long short-term memory full convolutional network (LSTM-FCN),

and the results showed that the performance of this method is better than some traditional machine learning methods [20]. Matousek et al. realized the recognition of ADB based on long short-term memory (LSTM) and replicator neural network (RNN) [21]. Shahverdy et al. recognized normal, aggressive, distracted, drowsy, and drunk driving styles based on convolutional neural networks (CNN) [22]. Khodairy achieved the recognition of ADB based on stacked long short-term memory (stacked-LSTM) [23].

Although the methods used in the above studies have realized the recognition of ADB, there are still some disadvantages. These methods usually assume that the distribution of the classes in the dataset is relatively balanced and the cost of misclassification is equal. Therefore, these methods cannot properly represent the distribution characteristics of the classes when dealing with class imbalance datasets, which leads to poor recognition performance [30–33]. Unfortunately, the samples of ADB are usually less than the samples of normal driving behavior (NDB) in the naturalistic driving datasets, which leads these methods to focus on correctly predicting NDB, while ignoring ADB as a minority class. Ensemble learning refers to the methods of training and combining multiple classifiers to complete specific machine learning tasks, which is considered as a solution to the class imbalance problem of machine learning [34]. By combining multiple classifiers, the error of a single classifier may be compensated by other classifiers. Therefore, the recognition performance of the ensemble classifier is usually better than that of a single classifier [34].

According to the above analysis, we propose a recognition method of ADB based on ensemble learning. In this method, the majority class data in the dataset is first divided into multiple groups, and each group of data is combined with the minority class data to construct the class balance dataset; next, the base classifiers are built based on the class balance datasets; finally, the base classifiers are combined based on different ensemble rules to build ensemble classifiers. The salient contributions of our work to the research of ADB recognition can be summarized as follows:

- The acquisition of multi-source naturalistic driving data: combined with the development status of intelligent and connected technology, an integrated experimental vehicle for driving behavior and safety based on the multi-sensor array is constructed. Based on this integrated experimental vehicle, a real vehicle experiment is designed and completed, and a naturalistic driving dataset containing ADB data is acquired;
- The research of the recognition performance of ensemble classifiers: to solve the problem of the poor recognition performance of machine learning method for the ADB data as a minority class in the dataset, a recognition method of ADB based on ensemble learning is proposed.

The rest of this paper is organized as follows. Section 2 introduces the composition of the integrated experimental vehicle for driving behavior and safety, the scheme of the real vehicle experiment, and the method of the data processing. Section 3 introduces the recognition method of ADB based on ensemble learning. Section 4 introduces the comparison results and discussion of the performance of the established ADB identification method and three typical deep learning methods. Section 5 presents the conclusion of this research.

2. Data Acquisition and Processing

In order to acquire the dataset suitable for the training and verification of the ADB recognition method based on ensemble learning, an integrated experimental vehicle for driving behavior and safety was constructed. As shown in Figure 1, the integrated experimental vehicle consists of the sensors, the data acquisition device, the cameras, the computing center, and the experimental vehicle. The functions of each component of the integrated experimental vehicle are shown in Table 1. The sensors used in the integrated experimental vehicle include long range radar (LRR), short range radar (SRR), inertial measurement unit (IMU), global positioning system (GPS), and an ultrasonic sensor. The functions and installation positions of the above sensors are shown in Table 2. The detection range of the LRR and SRR are shown in Figure 2. The coordinates of the IMU are shown in Figure 3.

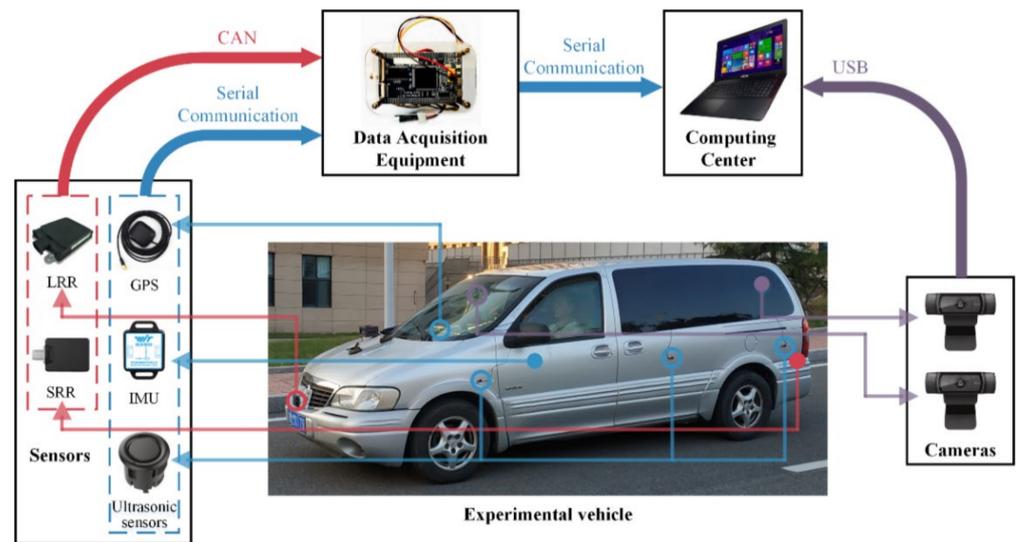


Figure 1. Integrated experimental vehicle for driving behavior and safety.

Table 1. The composition and function of the integrated experimental vehicle.

Compositions	Functions
Sensors	Acquire the vehicle motion parameters and the driving environment parameters.
Data acquisition device	Receive the data acquired by sensors and send it to the computing center.
Cameras	Record the video of the driving environment in the front and rear of the integrated experimental vehicle.
Computing center	Receive and save the data acquired by the data acquisition device and the video recorded by the cameras.
Experimental vehicle	The carrier for the sensors, data acquisition device, cameras, and computing center.

Table 2. The functions and installation positions of the above sensors.

Sensors	Functions	Installation Positions
Long range radar	Acquire the distance and relative speed between the integrated experimental vehicle and the front objects.	Above the front bumper.
Short range radar	Acquire the distance and the relative speed between the integrated experimental vehicle and the rear objects.	Above the rear bumper.
Inertial measurement unit	Acquire the acceleration and yaw rate of the vehicle.	About 1.8 m away from the front of the vehicle in the cab.
Global positioning system	Acquire the speed of the vehicle.	Above the console.
Ultrasonic sensor	Acquire the distance between the integrated experimental vehicle and the objects on the left and right sides.	On the left and right sides of the car.

The vehicle motion parameters and the driving environment parameters are collected at 10 Hz through the integrated experimental vehicle. The vehicle motion parameters include speed, acceleration, yaw rate, etc. Driving environment parameters include the distance and the relative speed between the integrated experimental vehicle and the objects, etc.

Six consecutive weeks of real vehicle experiment was conducted based on the integrated experimental vehicle for driving behavior and safety. The real vehicle experiment was conducted on one working day and one non-working day every week, and the data acquired every day included the data in rush hour and non-rush hour. Sixteen drivers were selected, including thirteen males and three females, to take part in the experiment. The age distribution of the drivers was between 23 and 50 years, and the average age was 28.9 years old. The driving age distribution was between 2 and 20 years, and the average driving age was 5.3 years. As shown in Figure 4, the road sections of Songling Road-Xianggang East Road in Laoshan, Qingdao were selected as the real vehicle experimental route. The route is a two-direction six-lane urban road with a total length of about 12 km.

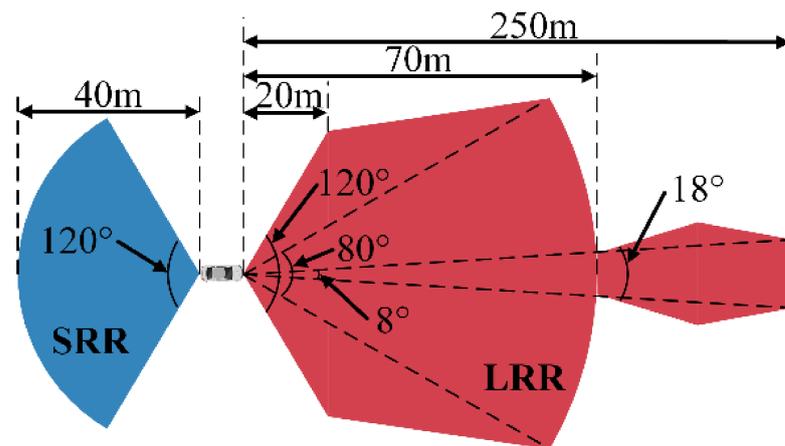


Figure 2. The detection range of the radars.

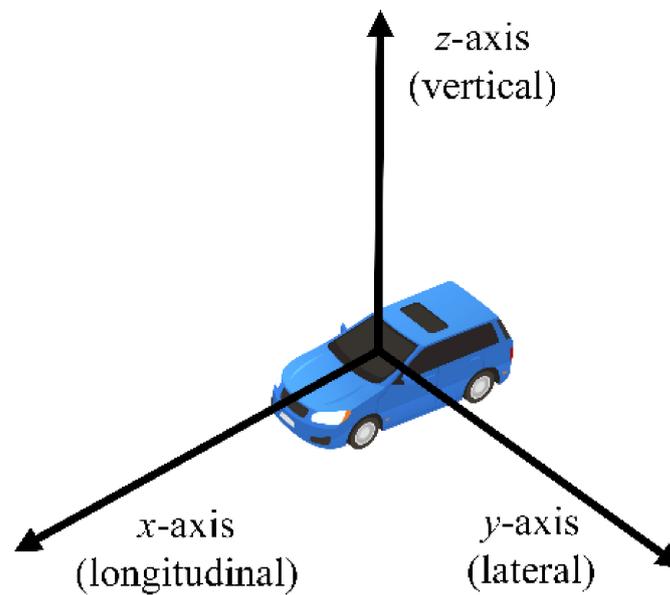


Figure 3. The coordinates of the IMU.



Figure 4. The experimental route of actual driving.

According to the aforementioned definition of ADB and previous studies, the longitudinal acceleration, lateral acceleration, yaw rate, distance between the experimental vehicle and the front vehicle, and relative speed between the experimental vehicle and the front vehicle were selected as the features of ADB. The above features are listed in Table 3. Among them, a_x is related to abnormal acceleration and deceleration, because the abnormal acceleration and deceleration are usually manifested as large longitudinal acceleration and deceleration; a_y and ω_z are related to abnormal lane changes because abnormal lane changes are usually manifested as large lateral accelerations and the large yaw rate; and d_f and v_f are related to the tailgating.

Table 3. Features.

Features	Descriptions	Units
a_x	The acceleration in the x-axis direction of the IMU, that is, the longitudinal acceleration of the vehicle.	m/s ²
a_y	The acceleration in the y-axis direction of the IMU, that is, the lateral acceleration of the vehicle.	m/s ²
ω_z	The angular velocity in the z-axis direction of the IMU, that is, the yaw rate of the vehicle.	deg/s
d_f	The distance between the vehicle and the front vehicle.	m
v_f	The relative speed between the vehicle and the front vehicle.	m/s

The essence of ADB recognition is a problem of the time series classification. Therefore, before building the model, a sliding window of fixed length is utilized to segment the data into overlapping series [23,35,36]. The length of the sliding window should be longer than the duration of the four abnormal driving events recorded in our experiment. However, the difference between the ADB series and the NDB series may be reduced if the sliding window is too long, which will lead to an increase in the miss rate and a decrease in the recognition accuracy. To balance the recognition and real-time performance of the recognition method of ADB, a sliding window with 50 time steps and 80% overlap is selected to process the raw data after several iterations of experiments.

Because the features we selected have different scales, the z-score is used to standardize the features, and the definition is shown in Equation (1).

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

where x is the unstandardized data, μ is the mean of the feature vector, σ is the standard deviation of the feature vector, and z is the standardized data.

After the above processing steps, a class imbalance dataset consisting of 31,506 standardized driving behavior series was obtained, which contained 28,908 NDB series and 2598 ADB series.

3. Recognition Method

Deep learning methods such as CNN [37], LSTM [38], and gated recurrent unit (GRU) [39] are widely used in time series modeling [40–44] due to their capacity to automatically learn the temporal dependencies present in time series [29]. The essence of ADB recognition is the classification of the time series, so CNN, LSTM, and GRU are utilized in this research. However, the recognition performance of the above methods is also sensitive to the imbalance of classes. Aiming to deal with this problem, an ensemble learning method [45] is employed to realize the recognition of ADB. The ensemble learning method models the class imbalance datasets by transforming one class imbalance problem into multiple class balance problems. The processes are as follows:

- (1) Dataset balancing: the majority class data in the dataset are divided into several groups so that the amount of data in each group is similar to that of the minority class data, and then each group of data is combined with the minority data to form multiple class balance datasets.

- (2) Base classifiers building: a base classifier is built for each class balance dataset based on a specific classification method.
- (3) Ensemble classifiers building: the obtained multiple base classifiers are combined into an ensemble classifier based on ensemble rules.

The framework of the recognition method of ADB based on ensemble learning is shown in Figure 5.

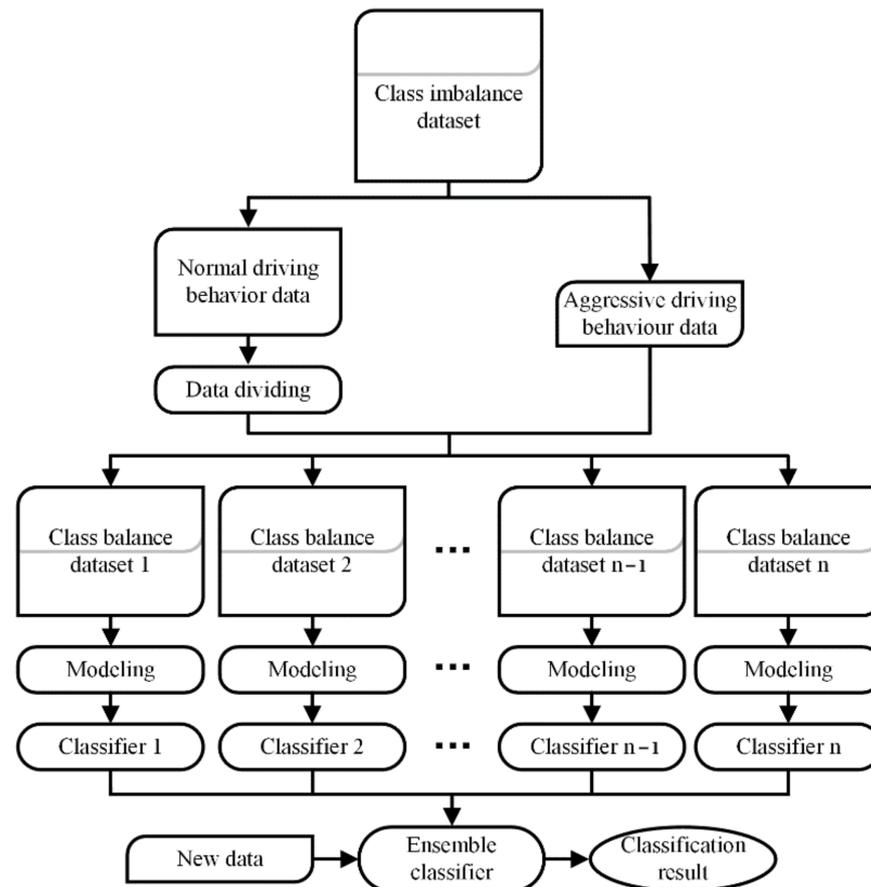


Figure 5. The framework of the recognition method of ADB based on ensemble learning.

3.1. Dataset Balancing

In the dataset balancing, the majority class data is divided into multiple groups, and each group of data is combined with the minority data to form multiple class balance datasets. Therefore, a clustering-based dataset balancing method is utilized [45], aiming to make the divided groups of majority class data have a similar amount to the minority class data in the dataset, and make the difference of the data within each group smaller. In this research, the self-organizing map (SOM) [46] is used to cluster the majority class data. SOM is an unsupervised learning neural network method that can map the high-dimension data to low-dimension space and is widely used in various fields, such as emotional intelligence [47], big data analysis [48], water quality assessment [49], and fault prediction [50]. The basic structure of SOM is shown in Figure 6. The output layer of SOM consists of a two-dimensional regular grid of neurons, and each neuron is represented by a weight vector \mathbf{m}_k , $\mathbf{m}_k = [m_{k1}, m_{k2}, \dots, m_{kd}]$, where d is the dimension of the input data, and k is the number of SOM neurons. In the clustering, SOM assigns the input data to the nearest neuron and updates the weight vector to minimize the distance of the data in the same neuron.

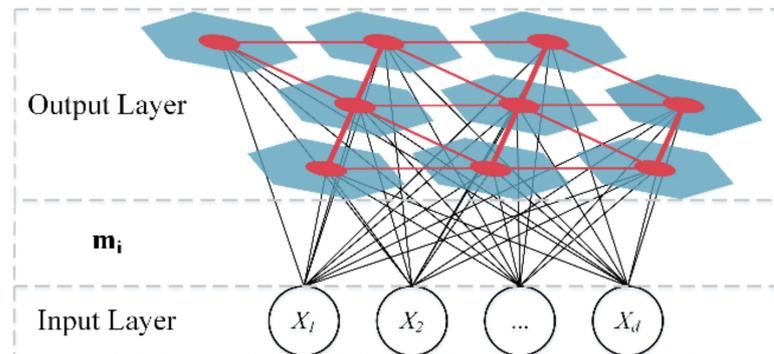


Figure 6. The basic structure of SOM.

The processes of dataset balancing based on SOM are as follows:

- (1) Randomly initialize the weight vectors m_k .
- (2) Input a 5-dimensional NDB series with 50-time steps as a 250-dimensional sample X and calculate the distance between the sample X and the weight vectors m_k . Calculate the Best-matching unit (BMU) m_c according to Equation (2), where BMU is the weight vector closest to the sample X .

$$\|X - m_c\| = \min_k \{\|X - m_k\|\} \quad (2)$$

- (3) BMU and its topological neighbors are updated according to Equation (3).

$$m_k(T+1) = m_k(T) + G_{c,k}(T)(X - m_k(T)) \quad (3)$$

$$G_{c,k}(T) = \alpha(T) \exp\left(-\frac{L_{c,k}^2}{2\sigma^2(T)}\right) \quad (4)$$

where T is the regression steps, $G_{c,k}(T)$ is the neighborhood function, $L_{c,k}$ is the distance between the sample X and BMU m_c , $\sigma(T)$ is the neighborhood kernel radius, and $\alpha(T)$ is the learning rate factor. Both $\sigma(T)$ and $\alpha(T)$ decrease monotonically with the regression steps.

- (4) Repeat steps (2) and (3) until the training is completed and the samples are divided into k groups.
- (5) Combine k groups of NDB samples with ADB samples to form k groups of class balance datasets.

3.2. Base Classifiers Building

After dataset balancing, the deep learning methods of CNN, LSTM, and GRU are employed to build multiple base classifiers with the multiple groups of class balance datasets. CNN is a deep neural network, usually composed of the convolutional layer, pooling layer, and fully connected layer. The basic structure of CNN is shown in Figure 7a. CNN can automatically extract features from high-dimensional raw data with network topology through convolution operations and is often used in machine vision and image processing [51,52]. The convolution operation is a sliding filter, which can capture repetitive patterns in time series through learning. The process of convolution operation is shown in Figure 8. Due to the above characteristics, CNN has been applied in time series modeling, such as financial market prediction [40], natural language processing [41], and driving behavior prediction [22]. CCN performs better than LSTM in some time series modeling tasks [53,54] and has a faster calculation speed [29,53]. LSTM and GRU are two improved recurrent neural networks (RNN), which can solve the problems of gradient disappearance and gradient explosion in traditional RNN when learning long-term dependence. RNN is widely used in time series modeling because it can connect each time step with the

previous time step to model the temporal dependencies of time series, such as traffic flow prediction [42], natural language processing [43], and financial market prediction [44]. The basic structure of RNN is shown in Figure 7b. As shown in Figure 9, the difference between RNN, LSTM, and GRU is the hidden layer.

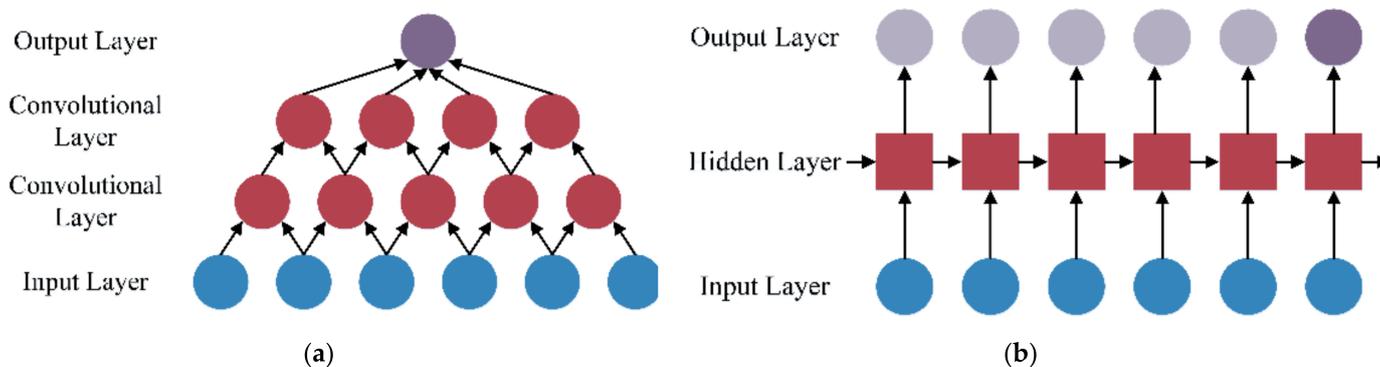


Figure 7. (a) The basic structure of CNN; (b) the basic structure of RNN.

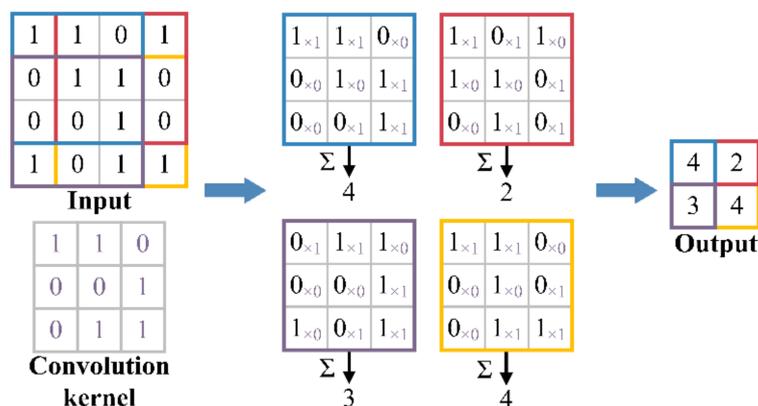


Figure 8. The process of the convolution operation.

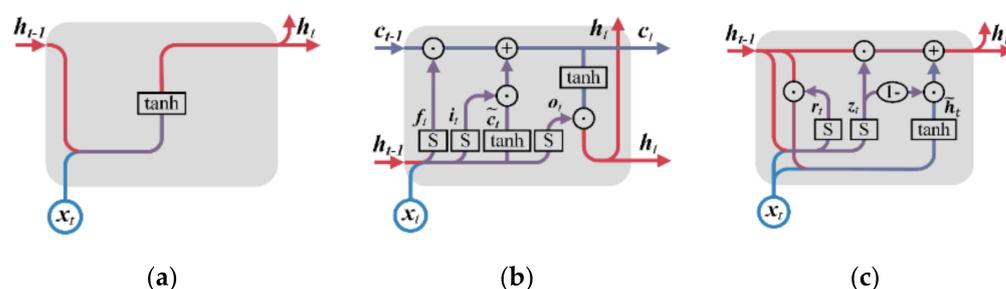


Figure 9. (a) The structure of RNN; (b) the structure of LSRM; (c) the structure of GRU.

LSTM is widely used in time series modeling tasks in various fields [42–44,55]. As shown in Figure 9b, LSTM solves the problem of exploding and vanishing gradients through the cell state c_t , which stores the long-term memory and retains or deletes the information passing through the hidden layer by the forget gate f_t , the input gate i_t , and the output gate o_t . The definitions of forget gate, input gate, and output gate are as follows:

$$f_t = S(w_{f1}h_{t-1} + w_{f2}x_t + b_f) \tag{5}$$

$$i_t = S(w_{i1}h_{t-1} + w_{i2}x_t + b_i) \tag{6}$$

$$o_t = S(w_{o1}h_{t-1} + w_{o2}x_t + b_o) \tag{7}$$

where S is the sigmoid activation function, and h_{t-1} is the hidden state of time $t - 1$. The current cell state and hidden state are defined as follows:

$$h_t = o_t \odot \tanh(c_t) \quad (8)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(\tilde{c}_t) \quad (9)$$

$$\tilde{c}_t = \tanh(w_{c1}h_{t-1} + w_{c2}x_t + b_c) \quad (10)$$

where \odot is the element-wise product, and \tanh is the tanh activation function.

GRU and LSTM have similar performance, but the GRU is simpler to calculate and implement [39] and has better convergence and generalization [56]. As shown in Figure 9c, the hidden unit of the GRU retains or deletes the input information at the current time x_t and the hidden state at the previous time H_{t-1} through the reset gate r_t and the update gate z_t to achieve the capture of short-term and long-term dependence. The reset gate and update gate are defined as follows:

$$r_t = S(w_{r1}H_{t-1} + w_{r2}x_t + b_r) \quad (11)$$

$$z_t = S(w_{z1}H_{t-1} + w_{z2}x_t + b_z) \quad (12)$$

The hidden state H_t is defined as follows:

$$H_t = z_t \odot H_{t-1} + (1 - z_t) \odot \tilde{H}_t \quad (13)$$

$$\tilde{H}_t = \tanh(r_t w_{H1} H_{t-1} + w_{H2} x_t + b_H) \quad (14)$$

3.3. Ensemble Classifiers Building

After the base classifiers building, the base classifiers are combined into the ensemble classifiers based on different ensemble rules. Referring to the ensemble process of the base classifiers of the ensemble learning method employed in this research [45], 10 different ensemble rules are applied to combine the base classifiers. There are five ensemble rules based on classification probabilities, including Max Rule, Min Rule, Product Rule, Majority Vote Rule, and Sum Rule [57]; and five ensemble rules based on classification probability combined with distance weighting mechanism, including MaxDistance Rule, MinDistance Rule, ProDistance Rule, MajDistance Rule, and SumDistance Rule [45]. The 10 ensemble rules and their strategies are shown in Table 4. The C_1 and C_2 are the class labels of data. The R_1 and R_2 represent the ensemble rules of the classes C_1 and C_2 . The P_{j1} represents the probability that the j th classifier classifies the data into C_1 . The P_{j2} represents the probability that the j th classifier classifies the data into C_2 . The D_{j1} represents the average distance between the new data and the data with the class label C_1 in the j th class balance dataset. The D_{j2} represents the average distance between the new data and the data with the class label C_2 in the j th class balance dataset.

The definition of the function $f(x, y)$ is shown in Equation (15).

$$f(x, y) = \begin{cases} 1 & x \geq y \\ 0 & x < y \end{cases} \quad (15)$$

The final classification result of the data is obtained based on the ensemble rules in Table 4. The data class is C_1 if $R_1 \geq R_2$, otherwise C_2 .

Table 4. The strategies of the ensemble rules.

Ensemble Rules	Strategies
Max Rule	$R_1 = \arg \max_{1 < j < K} P_{j1}, R_2 = \arg \max_{1 < j < K} P_{j2}$
Min Rule	$R_1 = \arg \min_{1 < j < K} P_{j1}, R_2 = \arg \min_{1 < j < K} P_{j2}$
Product Rule	$R_1 = \prod_{j=1}^K P_{j1}, R_2 = \prod_{j=1}^K P_{j2}$
Majority Vote Rule	$R_1 = \sum_{j=1}^K f(P_{j1}, P_{j2}), R_2 = \sum_{j=1}^K f(P_{j2}, P_{j1})$
Sum Rule	$R_1 = \sum_{j=1}^K P_{j1}, R_2 = \sum_{j=1}^K P_{j2}$
MaxDistance Rule	$R_1 = \arg \max_{1 < j < K} \frac{P_{j1}}{D_{j1+1}}, R_2 = \arg \max_{1 < j < K} \frac{P_{j2}}{D_{j2+1}}$
MinDistance Rule	$R_1 = \arg \min_{1 < j < K} \frac{P_{j1}}{D_{j1+1}}, R_2 = \arg \min_{1 < j < K} \frac{P_{j2}}{D_{j2+1}}$
ProDistance Rule	$R_1 = \prod_{j=1}^K \frac{P_{j1}}{D_{j1+1}}, R_2 = \prod_{j=1}^K \frac{P_{j2}}{D_{j2+1}}$
MajDistance Rule	$R_1 = \sum_{j=1}^K \frac{f(P_{j1}, P_{j2})}{D_{j1+1}}, R_2 = \sum_{j=1}^K \frac{f(P_{j2}, P_{j1})}{D_{j2+1}}$
SumDistance Rule	$R_1 = \sum_{j=1}^K \frac{P_{j1}}{D_{j1+1}}, R_2 = \sum_{j=1}^K \frac{P_{j2}}{D_{j2+1}}$

4. Results and Discussion

The validation set is composed of 300 NDB samples and 300 ADB samples, which are randomly selected from NDB samples and ADB samples, respectively. The training set is composed of the remaining 28,608 NDB samples and 2298 ADB samples. After the training is completed, the accuracy (a), precision (p), recall (r), and F₁-score (F) of each classifier are calculated. The F₁-score is the harmonic mean of precision and recall, which is closer to the smaller of the two; a high F₁-score can ensure that both the precision and recall are high [30]. Therefore, the performance of the classifiers in recognizing ADB is evaluated by F₁-score as the main evaluation metric and accuracy, precision, and recall as the supplementary evaluation metrics. The definition of F₁-score is shown in Equation (16):

$$F = \frac{2pr}{p+r} \quad (16)$$

The definitions of accuracy, precision, and recall are as follows:

$$a = \frac{TP + TN}{TP + FP + TN + FN} \quad (17)$$

$$p = \frac{TP}{TP + FP} \quad (18)$$

$$r = \frac{TP}{TP + FN} \quad (19)$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

In the dataset balancing, the NDB samples in the training set are divided into multiple groups, and the number of NDB samples in each group should be as close as possible to the number of ADB samples in the training set. Therefore, according to the ratio of the ADB samples to the NDB samples in the training set, the number of SOM neurons is set as 12, and the mapping size is set to 4×3 after several tests. After the clustering, the NDB data in the training set are divided into 12 groups. As shown in Figure 10, the white numbers in grids represent the number of NDB samples in the group. The size of the purple shape in

the grid is proportional to the number of NDB samples in this group. By combining the 12 groups of the NDB samples with the ADB samples in the training set, 12 groups of class relative balance datasets are obtained.

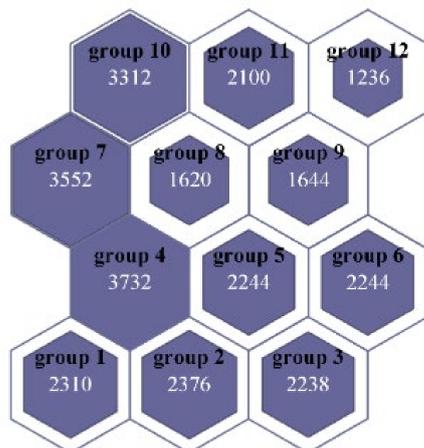


Figure 10. The number of NDB samples in each group.

The weight vectors of 12 neurons after training are shown in Figure 11. To show the characteristics of the 12 weight vectors more intuitively, they are compared with several randomly selected ADB samples. As shown in Figure 11a–c, the values and the fluctuations with time steps of the a_x , the a_y , and the ω_z of the 12 weight vectors are small, whereas the values and the fluctuations with time steps of the a_x , the a_y , and the ω_z of most ADB samples are large. Because the difference between d_f and v_f of the 12 weight vectors and the ADB samples is difficult to be directly observed, the time to collision (TTC) is calculated based on d_f and v_f . The definition of TTC is shown in Equation (20), and the negative value of TTC means that the experimental vehicle is approaching the vehicle in front. As shown in Figure 11d, the TTC of the 12 weight vectors and some ADB samples is less than 0. However, the TTC of these ADB samples is closer to 0, which means a higher risk of collision.

$$TTC = \frac{d_f}{v_f} \quad (20)$$

After the dataset balancing, all the methods, including CNN, LSTM, and GRU, are used to build 12 base classifiers with the 12 groups of class balance datasets, respectively, and the 12 base classifiers are combined into the ensemble classifiers based on the 10 different ensemble rules shown in Table 4. In addition, CNN, LSTM, and GRU are used to directly build the classifiers on the class imbalance dataset without ensemble learning. The main parameters of CNN, LSTM, and GRU are shown in Table 5, which are selected after several tests. The number of layers in Table 5 indicates the number of convolutional layers, LSTM layers, or GRU layers in the models. The CNN is designed with a single convolutional layer. LSTM is designed with a single LSTM layer with 128 hidden units. GRU is designed with a single GRU layer with 128 hidden units. In addition, the “/” in Table 5 indicates that the parameter is not utilized in the models.

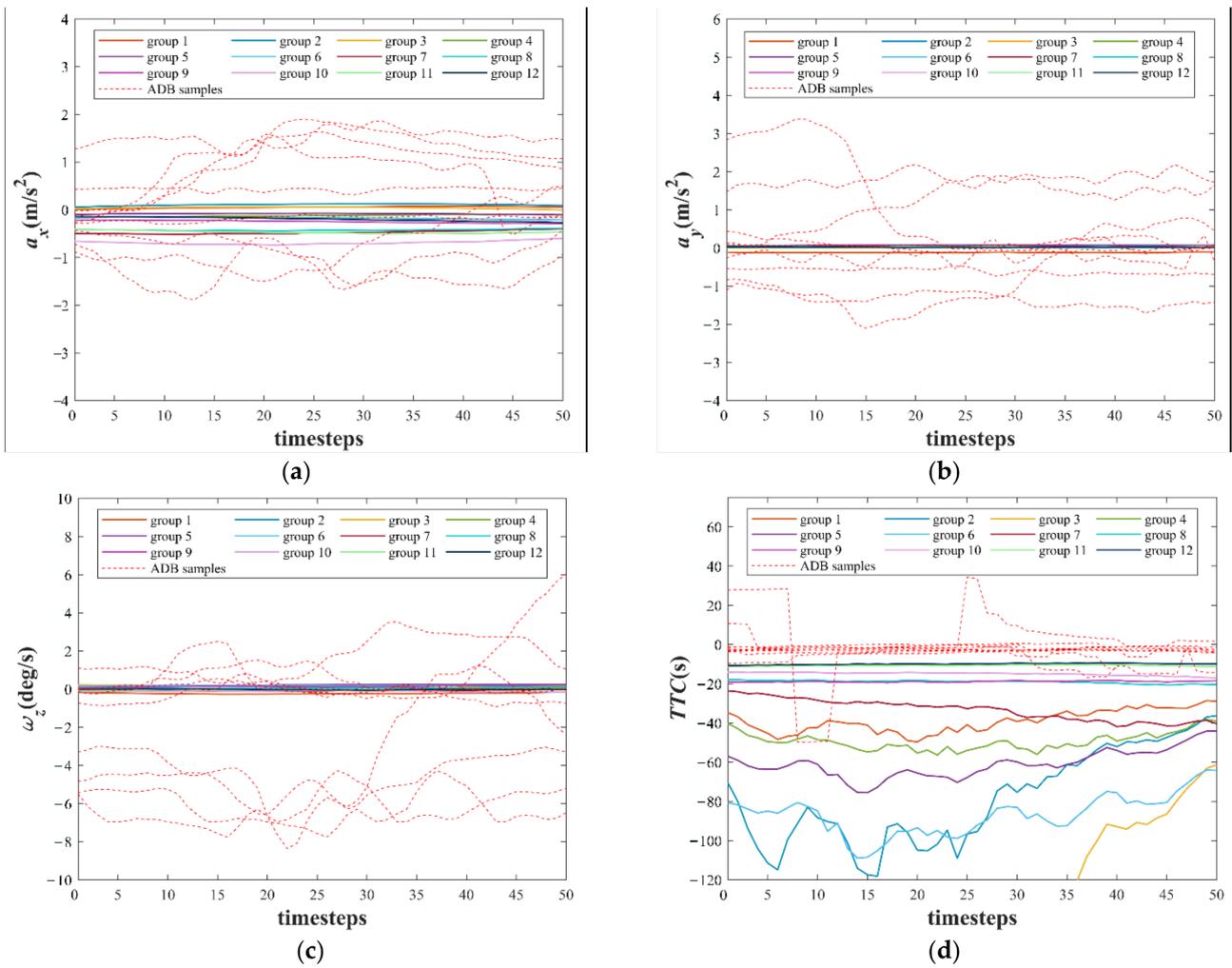


Figure 11. (a) The a_x of the weight vectors and ADB samples; (b) the a_y of the weight vectors and ADB samples; (c) the ω_z of the weight vectors and ADB samples; (d) the TTC of the weight vectors and ADB samples.

Table 5. The parameters of models.

Models	Batch Size	Learning Rate	Layers	Units	Convolution			Max Pooling			
					Filters	Filters Size	Stride	Padding Size	Size	Stride	Padding Size
CNN	32	0.001	1	/	10	5 × 2	1	0	2 × 2	2	0
LSTM	32	0.001	1	128			/			/	
GRU	32	0.001	1	128			/			/	

The confusion matrices of all the classifiers and ensemble classifiers obtained by verification are shown in Figures 12–14. The results obtained by the three deep learning methods before and after the application of ensemble learning have similar characteristics. Compared with the classifier built without ensemble learning, the ensemble classifier built with ensemble learning has a slight increase in the misclassification of NDB samples, but it greatly improves the accuracy of the classification of ADB samples. For the ensemble classifiers, the ones built with the ensemble rules based on classification probability have obtained similar results, and they have fewer misclassification of ADB samples. However, after the ensemble rules based on classification probability are combined with the distance weighting mechanism, they have more misclassification of ADB samples and fewer misclassification of NDB samples.

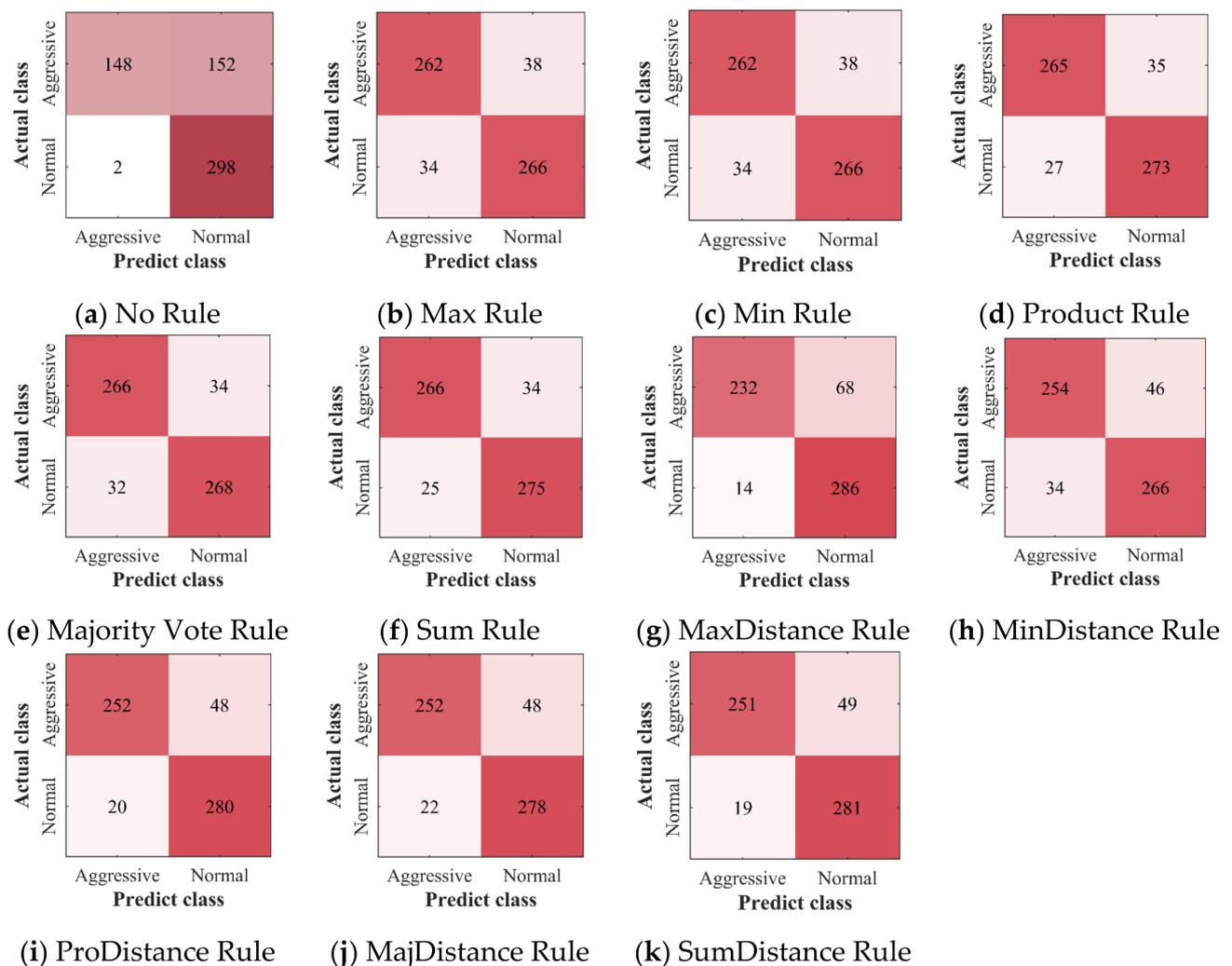


Figure 12. The confusion matrices of CNN: (a) the confusion matrix of the classifier built without ensemble learning; (b–k) the confusion matrices of ensemble classifiers built with 10 different ensemble rules.

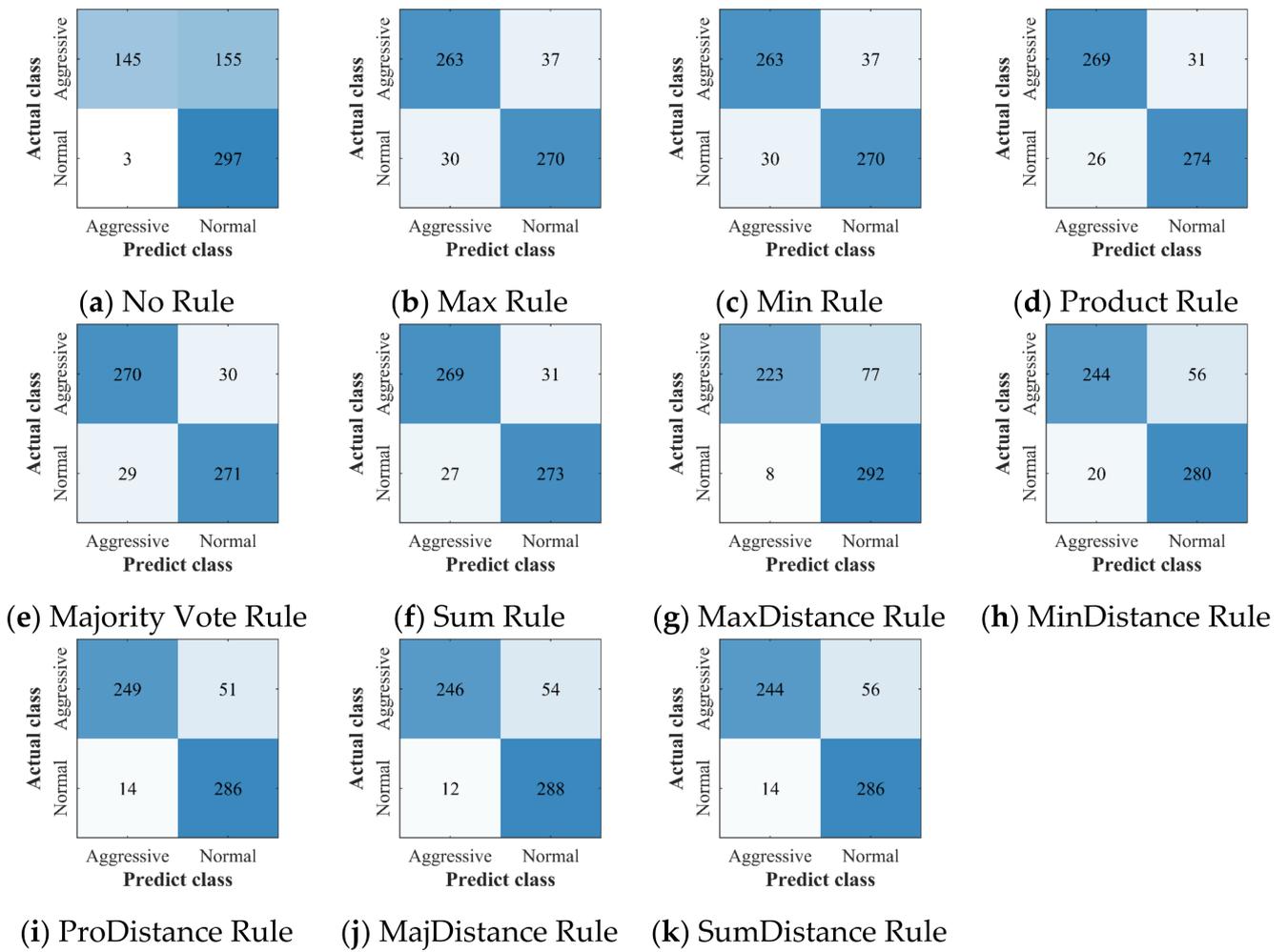


Figure 13. The confusion matrices of LSTM: (a) the confusion matrix of the classifier built without ensemble learning; (b–k) the confusion matrices of ensemble classifiers built with 10 different ensemble rules.

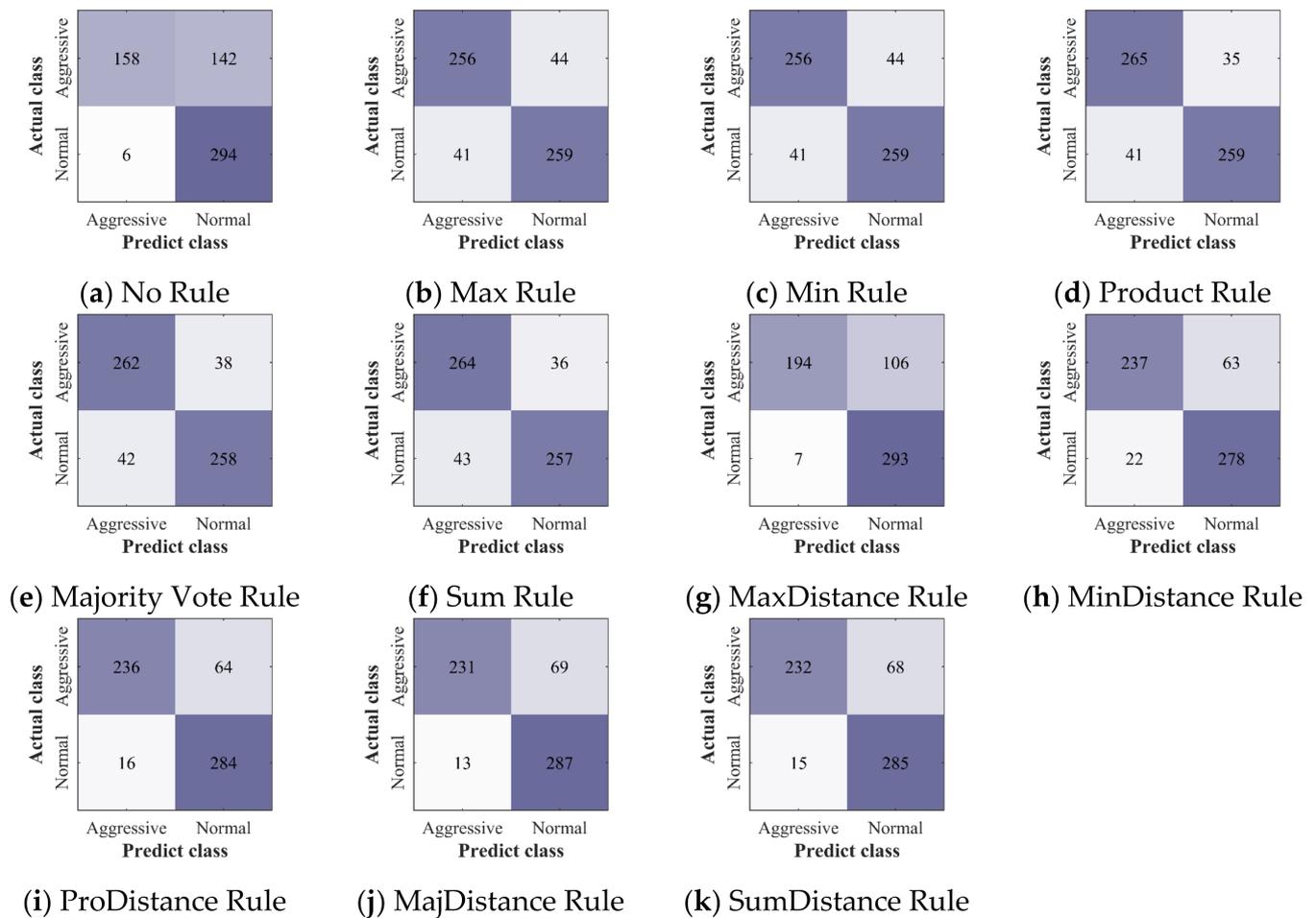


Figure 14. The confusion matrices of GRU: (a) the confusion matrix of the classifier built without ensemble learning; (b–k) the confusion matrices of ensemble classifiers built with 10 different ensemble rules.

In order to express the performance of each classifier more intuitively, the accuracy, precision, recall, and F₁-score of each classifier are calculated and listed in Tables 6–8.

Table 6. The validation results of CNN classifiers based on different ensemble rules.

Ensemble Rules	Accuracy	Precision	Recall	F ₁ -Score
No Rule	74.33%	98.66%	49.33%	65.78%
Max Rule	88.00%	88.51%	87.33%	87.92%
Min Rule	88.00%	88.51%	87.33%	87.92%
Product Rule	89.67%	90.75%	88.33%	89.53%
Majority Vote Rule	89.00%	89.26%	88.67%	88.96%
Sum Rule	90.17%	91.41%	88.67%	90.02%
MaxDistance Rule	86.33%	94.31%	77.33%	84.98%
MinDistance Rule	86.67%	88.19%	84.67%	86.39%
ProDistance Rule	88.67%	92.65%	84.00%	88.11%
MajDistance Rule	88.33%	91.97%	84.00%	87.80%
SumDistance Rule	88.67%	92.96%	83.67%	88.07%

Table 7. The validation results of LSTM classifiers based on different ensemble rules.

Ensemble Rules	Accuracy	Precision	Recall	F ₁ -Score
No Rule	73.67%	97.97%	48.33%	64.73%
Max Rule	88.83%	89.76%	87.67%	88.70%
Min Rule	88.83%	89.76%	87.67%	88.70%
Product Rule	90.50%	91.19%	89.67%	90.42%
Majority Vote Rule	90.17%	90.30%	90.00%	90.15%
Sum Rule	90.33%	90.88%	89.67%	90.27%
MaxDistance Rule	85.83%	96.54%	74.33%	83.99%
MinDistance Rule	87.33%	92.42%	81.33%	86.52%
ProDistance Rule	89.17%	94.68%	83.00%	88.45%
MajDistance Rule	89.00%	95.35%	82.00%	88.17%
SumDistance Rule	88.33%	94.57%	81.33%	87.46%

Table 8. The validation results of GRU classifiers based on different ensemble rules.

Ensemble Rules	Accuracy	Precision	Recall	F ₁ -Score
No Rule	75.33%	96.34%	52.67%	68.10%
Max Rule	86.00%	86.49%	85.33%	85.91%
Min Rule	86.00%	86.49%	85.33%	85.91%
Product Rule	87.17%	86.32%	88.33%	87.31%
Majority Vote Rule	86.67%	86.18%	87.33%	86.75%
Sum Rule	86.83%	85.99%	88.00%	86.99%
MaxDistance Rule	81.17%	96.52%	64.67%	77.45%
MinDistance Rule	85.83%	91.51%	79.00%	84.79%
ProDistance Rule	86.67%	93.64%	78.67%	85.51%
MajDistance Rule	86.33%	94.67%	77.00%	84.93%
SumDistance Rule	86.17%	93.93%	77.33%	84.83%

As shown in Tables 6–8, the ensemble classifiers achieve higher accuracy, recall, and F₁-score, which shows that compared with classifiers built without ensemble learning, the ensemble classifiers can recognize ADB more accurately. Classifiers built without ensemble learning achieve higher precision and lower recall, which reflects the problem that some machine learning methods are more likely to misclassify minority classes. Compared with the ensemble rules based on classification probability, the ensemble rules combined with the distance weighting mechanism achieve higher precision and lower recall, which means that they have more misclassification of ADB samples. This may be caused by the high dimension of time series data because the distance difference between different data points gradually decreases with the increase of dimension [58]. Therefore, the ensemble classifiers built without distance weighting mechanism are more suitable for the recognition of ADB.

Among the classifiers built without ensemble learning, the one based on the GRU achieves the highest accuracy of 75.33%, recall rate of 52.67%, and F₁-score of 68.10%, whereas the one based on the CNN achieves the highest precision of 98.66%. Therefore, among the three classifiers built without ensemble learning, the one based on the GRU with the highest F₁-score achieves the best performance in the recognition of ADB.

Among the ensemble classifiers, the one based on the LSTM and the Product Rule achieves the highest accuracy of 90.50%, which indicates that only 9.50% of the samples are misclassified. The one based on the LSTM and the Majority Vote Rule achieves the highest recall of 90.00%, which indicates that only 10.00% of ADB samples are misclassified. The one based on the LSTM and the MaxDistance Rule achieves the highest precision of 96.54%, which indicates that only 3.46% of the samples classified as ADB are misclassified. The one based on the LSTM and the Product Rule achieves the highest F₁-score of 90.42%.

Ensemble learning has the greatest improvement to the LSTM, which makes the LSTM ensemble classifiers perform the best in the recognition of ADB. The performance of the CNN ensemble classifiers in the recognition of ADB is second only to the LSTM ensemble classifiers, and the GRU ensemble classifiers have the worst performance.

To intuitively compare the influence of different ensemble rules on the recognition performance under different evaluation metrics, the classifier built without ensemble learning, which has the worst performance in each evaluation metric, is used as the benchmark “1” to calculate the increase rate and decrease rate of the evaluation metrics for ensemble classifiers. The results are shown in Figures 15–18.

As shown in Figures 15–18, the accuracy, recall, and F_1 -score of the three deep learning methods are significantly improved by ensemble learning. As shown in Figure 15, the increase rate of the accuracy for each ensemble classifier is more than 10%, among which the one based on the LSTM and the Product Rule achieves the highest increase rate of 22.85%, followed by the one based on the LSTM and the Sum Rule. This means that although the ensemble learning method increases the misclassification of NDB samples, it reduces the misclassification of more ADB samples. As shown in Figure 16, the precision of most ensemble classifiers has slightly decreased, among which the one based on the GRU and the Sum Rule achieves the highest decrease rate of 10.74%. Although only the ensemble classifiers based on the MaxDistance rule have increased precision, their increase rates of other evaluation metrics are the lowest. Therefore, we consider that the ensemble classifiers based on the MaxDistance Rule have the worst performance. As shown in Figure 17, the increase rate of the recall for each ensemble classifier is more than 33%, among which the one based on the LSTM and the Majority Vote Rule achieves the highest increase rate of 86.22%, followed by the one based on the LSTM and the Product Rule and the Sum Rule. The recall has been significantly improved, which shows that the misclassification of minority class samples is substantially reduced after the application of the ensemble learning method. As shown in Figure 18, the increase rate of the F_1 -score for each ensemble classifier is more than 19%, among which the one based on the LSTM and the Product Rule achieves the highest increase rate of 39.69%, followed by the one based on the LSTM and the Sum Rule. The F_1 -score of most ensemble classifiers is about 30%, which shows that the ensemble learning method effectively improves the recognition performance of three deep learning methods for ADB.

Among the 10 ensemble rules, the Product Rule has the highest improvement to the LSTM and the GRU. Compared with the LSTM and the GRU classifiers built without ensemble learning, the increase rate of the F_1 -score for the LSTM and the GRU ensemble classifier based on the Product Rule are 39.69% and 34.88%, and the LSTM ensemble classifier based on the Product Rule achieves the highest F_1 -score of 90.42%. The Sum Rule achieves the highest improvement to the CNN. Compared with the CNN classifier built without ensemble learning, the increase rate of the F_1 -score for the CNN ensemble classifier based on the Sum Rule is 39.07%. Compared with the other ensemble classifiers, CNN, LSTM, and GRU ensemble classifiers based on the MaxDistance Rule achieves higher precision and lower accuracy, recall, and F_1 -score, and have the worse performance in the recognition of ADB. Overall, ensemble learning significantly improves the recognition performances of the three deep learning methods for ADB. The LSTM ensemble classifier based on the Product Rule with the highest F_1 -score of 90.42% achieves the best performance for ADB recognition, followed by the LSTM ensemble classifier based on the Sum Rule. However, most of the ensemble classifiers built based on ensemble learning have a slight decrease in precision, which means that their recognition performance for NDB has decreased.

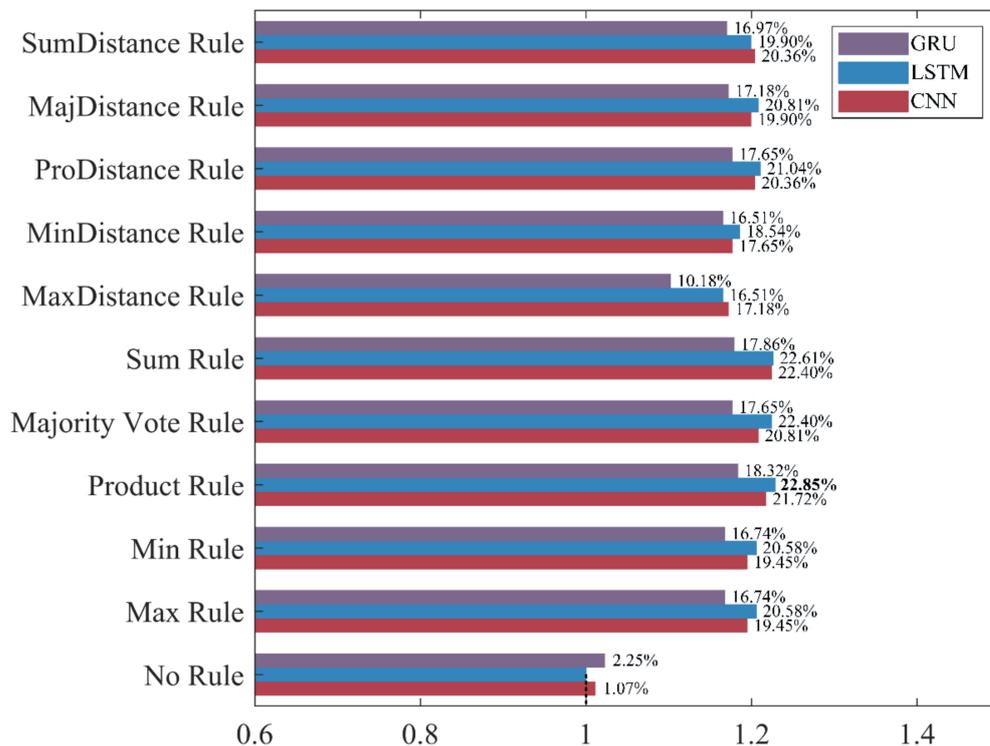


Figure 15. The increase rate and decrease rate of the accuracy for ensemble classifiers. The bold data indicates the highest increase rate.

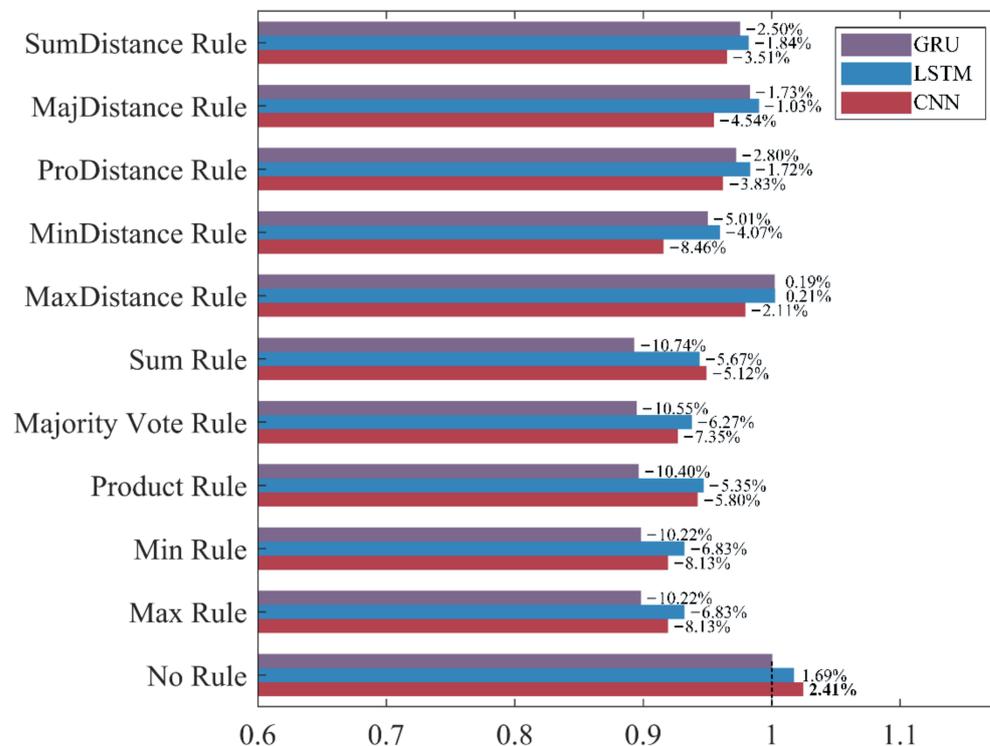


Figure 16. The increase rate and decrease rate of the precision for ensemble classifiers. The bold data indicates the highest increase rate.

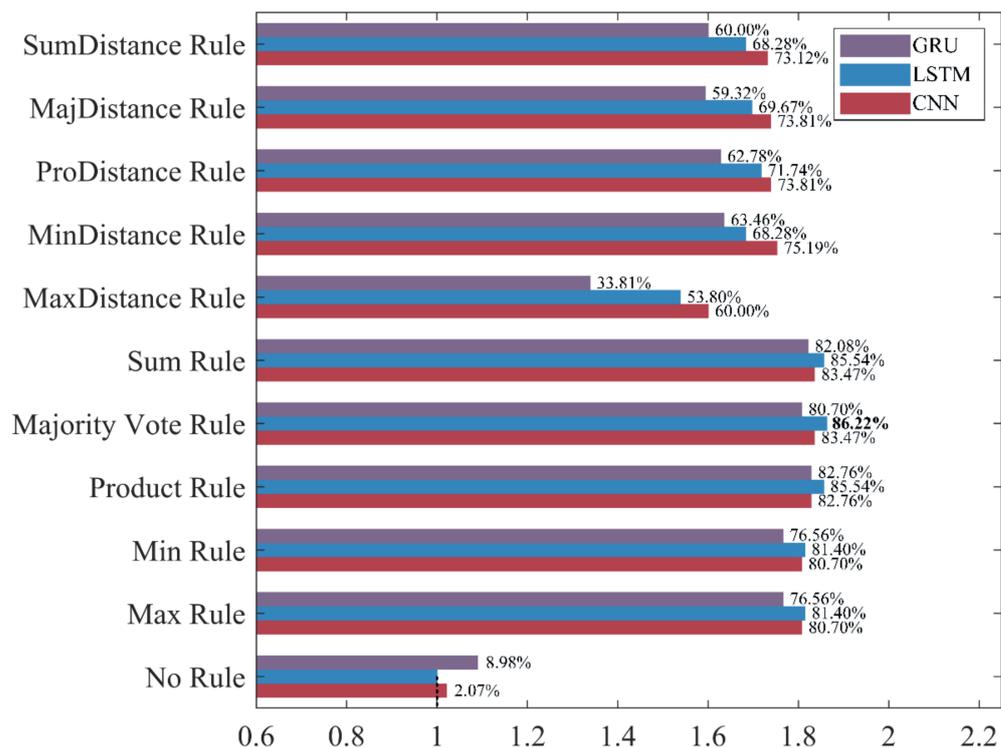


Figure 17. The increase rate and decrease rate of the recall for ensemble classifiers. The bold data indicates the highest increase rate.

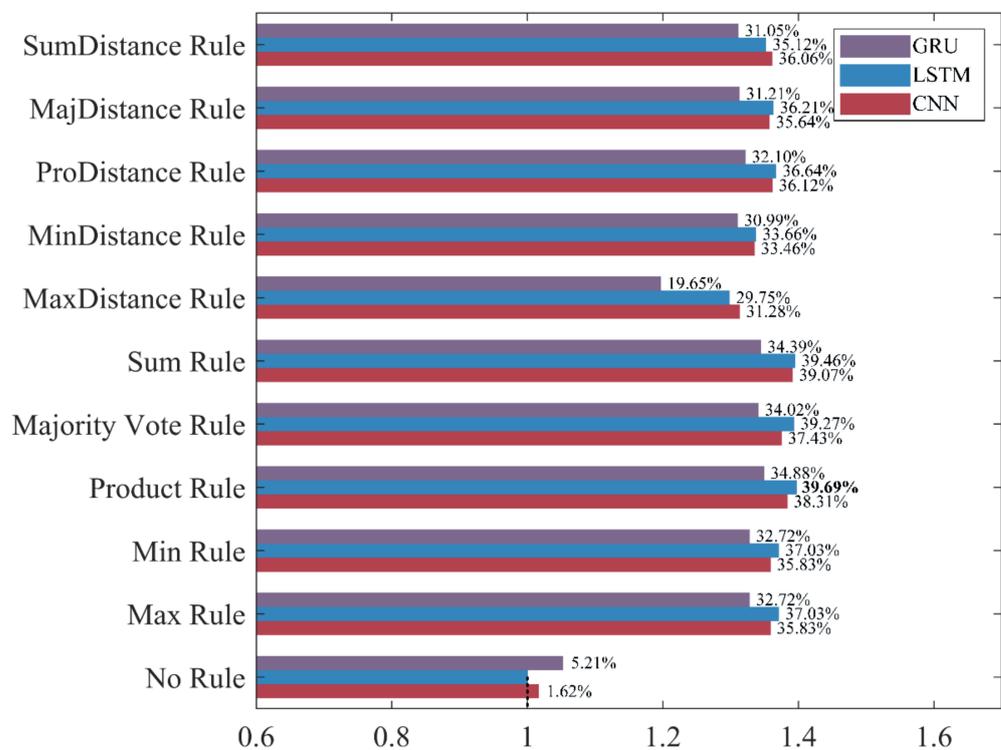


Figure 18. The increase rate and decrease rate of the F₁-score for ensemble classifiers. The bold data indicates the highest increase rate.

In the research, the recognition of ADB is realized based on the motion parameters of vehicles. However, ADB is not only reflected in the four abnormal driving behaviors specified in this research, it is also reflected in behaviors such as frequent whistles and disregard of traffic rules. Therefore, the research on the recognition of aggressive driving behavior that integrates existing parameters and other behavior-related parameters is the focus topic of further work. In addition, verifying the application of other methods in this ensemble learning framework is also the focus of future work, such as applying other clustering methods or directly dividing the majority class samples in the dataset balancing and the application of other deep learning methods in the base classifiers building. Moreover, we will also focus on the application of unsupervised learning and semi-supervised learning methods in the research of aggressive driving behavior recognition.

5. Conclusions

The accurate recognition of ADB is the premise to timely and effectively conduct warning or intervention to the driver, which is of great importance for improving driving safety. In this paper, a recognition method of ADB is built based on ensemble learning through the dataset balancing, base classifiers building, and ensemble classifiers building, and the method is trained and verified by a multi-source driving behavior dataset acquired under naturalistic driving conditions. The results suggest that the ensemble classifiers built with ensemble learning achieve higher accuracy, recall, and F_1 -score. In contrast, although the classifiers built without ensemble learning achieve higher precision, they have lower accuracy, recall, and F_1 -score. This comparison result suggests that the ensemble classifier is more suitable for accurately recognizing the ADB with a small proportion in the dataset, whereas the classifier built without ensemble learning is more suitable for recognizing the NDB that is more common. Among the ensemble classifiers built with different rules, the one based on the LSTM and the Product Rule obtains the highest accuracy (90.50%) and F_1 -score (90.42%), which has the optimal performance for ADB recognition. The one based on the LSTM and the Sum Rule has the suboptimal performance for ADB recognition. In summary, the recognition method of ADB based on ensemble learning proposed in this paper can solve the problem of class imbalance in the dataset and achieve a significant improvement in recognition performance. The results can provide a reference for the improvement of the advanced driver assistance system and the realization of personalized driver assistance systems as well as the anthropomorphic automatic vehicle.

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

Funding: This research was funded by the Natural Science Foundation of Shandong Province, grant number ZR2020MF082; the Collaborative Innovation Center for Intelligent Green Manufacturing Technology and Equipment of Shandong Province, grant number IGSD-2020-012; the Qingdao Top Talent Program of Entrepreneurship and Innovation, grant number 19-3-2-11-zhc; and the National Key Research and Development Program, grant number 2018YFB1601500.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Conflicts of Interest: There are no conflict of interest.

References

1. Petridou, E.; Moustaki, M. Human Factors in the Causation of Road Traffic Crashes. *Eur. J. Epidemiol.* **2000**, *16*, 819–826. [CrossRef] [PubMed]
2. Aggressive Driving Research Update. Available online: https://safety.fhwa.dot.gov/speedmgmt/ref_mats/fhwasa1304/resources2/38%20-%20Aggressive%20Driving%202009%20Research%20Update.pdf (accessed on 22 May 2021).
3. Wickens, C.M.; Mann, R.E.; Ialomiteanu, A.R.; Stoduto, G. Do Driver Anger and Aggression Contribute to the Odds of a Crash? A Population-Level Analysis. *Transp. Res. Part F Traffic Psychol. Behav.* **2016**, *42*, 389–399. [CrossRef]
4. Vahedi, J.; Shariat Mohaymany, A.; Tabibi, Z.; Mehdizadeh, M. Aberrant Driving Behaviour, Risk Involvement, and Their Related Factors Among Taxi Drivers. *Int. J. Environ. Res. Public Health* **2018**, *15*, 1626. [CrossRef] [PubMed]
5. Shinar, D.; Compton, R. Aggressive Driving: An Observational Study of Driver, Vehicle, and Situational Variables. *Accid. Anal. Prev.* **2004**, *36*, 429–437. [CrossRef]
6. Hennessy, D.; Wiesenthal, D. Traffic Congestion, Driver Stress, and Driver Aggression. *Aggress. Behav.* **1999**, *25*, 409–423. [CrossRef]
7. Kováčsová, N.; Lajunen, T.; Rošková, E. Aggression on the Road: Relationships between Dysfunctional Impulsivity, Forgiveness, Negative Emotions, and Aggressive Driving. *Transp. Res. Part F Traffic Psychol. Behav.* **2016**, *42*, 286–298. [CrossRef]
8. Shinar, D. Aggressive Driving: The Contribution of the Drivers and the Situation. *Transp. Res. Part F Traffic Psychol. Behav.* **1998**, *1*, 137–160. [CrossRef]
9. A Review of the Literature on Aggressive Driving Research. Available online: <https://www.stopandgo.org/research/aggressive/tasca.pdf> (accessed on 15 June 2021).
10. Ellison-Potter, P.; Bell, P.; Deffenbacher, J. The Effects of Trait Driving Anger, Anonymity, and Aggressive Stimuli on Aggressive Driving Behavior. *J. Appl. Soc. Psychol.* **2001**, *31*, 431–443. [CrossRef]
11. Wang, J.; Zhu, S.; Gong, Y. Driving Safety Monitoring Using Semisupervised Learning on Time Series Data. *IEEE Trans. Intell. Transport. Syst.* **2010**, *11*, 728–737. [CrossRef]
12. Danaif, M.; Abou-Zeid, M.; Kaysi, I. Modeling Anger and Aggressive Driving Behavior in a Dynamic Choice–Latent Variable Model. *Accid. Anal. Prev.* **2015**, *75*, 105–118. [CrossRef]
13. Fitzpatrick, C.D.; Samuel, S.; Knodler, M.A. The Use of a Driving Simulator to Determine How Time Pressures Impact Driver Aggressiveness. *Accid. Anal. Prev.* **2017**, *108*, 131–138. [CrossRef]
14. Kerwin, T.; Bushman, B.J. Measuring the Perception of Aggression in Driving Behavior. *Accid. Anal. Prev.* **2020**, *145*, 105709. [CrossRef] [PubMed]
15. Ma, Y.; Tang, K.; Chen, S.; Khattak, A.J.; Pan, Y. On-Line Aggressive Driving Identification Based on in-Vehicle Kinematic Parameters under Naturalistic Driving Conditions. *Transp. Res. Part C Emerg. Technol.* **2020**, *114*, 554–571. [CrossRef]
16. Feng, F.; Bao, S.; Sayer, J.R.; Flannagan, C.; Manser, M.; Wunderlich, R. Can Vehicle Longitudinal Jerk Be Used to Identify Aggressive Drivers? An Examination Using Naturalistic Driving Data. *Accid. Anal. Prev.* **2017**, *104*, 125–136. [CrossRef] [PubMed]
17. Zylus, G. Investigation of Route-Independent Aggressive and Safe Driving Features Obtained from Accelerometer Signals. *IEEE Intell. Transport. Syst. Mag.* **2017**, *9*, 103–113. [CrossRef]
18. Ma, Y.; Zhang, Z.; Chen, S.; Yu, Y.; Tang, K. A Comparative Study of Aggressive Driving Behavior Recognition Algorithms Based on Vehicle Motion Data. *IEEE Access* **2019**, *7*, 8028–8038. [CrossRef]
19. Carlos, M.R.; Gonzalez, L.C.; Wahlstrom, J.; Ramirez, G.; Martinez, F.; Runger, G. How Smartphone Accelerometers Reveal Aggressive Driving Behavior?—The Key Is the Representation. *IEEE Trans. Intell. Transport. Syst.* **2020**, *21*, 3377–3387. [CrossRef]
20. Moukafih, Y.; Hafidi, H.; Ghogho, M. Aggressive driving detection using deep learning-based time series classification. In Proceedings of the 2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), Sofia, Bulgaria, 3–5 July 2019; pp. 1–5.
21. Matousek, M.; EL-Zohairy, M.; Al-Momani, A.; Kargl, F.; Bosch, C. Detecting anomalous driving behavior using neural networks. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 2229–2235.
22. Shahverdy, M.; Fathy, M.; Berangi, R.; Sabokrou, M. Driver Behavior Detection and Classification Using Deep Convolutional Neural Networks. *Expert Syst. Appl.* **2020**, *149*, 113240. [CrossRef]
23. Khodairy, M.A.; Abosamra, G. Driving Behavior Classification Based on Oversampled Signals of Smartphone Embedded Sensors Using an Optimized Stacked-LSTM Neural Networks. *IEEE Access* **2021**, *9*, 4957–4972. [CrossRef]
24. Carvalho Barbosa, R.; Shoaib Ayub, M.; Lopes Rosa, R.; Zegarra Rodríguez, D.; Wuttisittikulij, L. Lightweight PVIDNet: A Priority Vehicles Detection Network Model Based on Deep Learning for Intelligent Traffic Lights. *Sensors* **2020**, *20*, 6218. [CrossRef]
25. Silva, J.C.; Saadi, M.; Wuttisittikulij, L.; Militani, D.R.; Rosa, R.L.; Rodriguez, D.Z.; Otaibi, S.A. Light-Field Imaging Reconstruction Using Deep Learning Enabling Intelligent Autonomous Transportation System. *IEEE Trans. Intell. Transport. Syst.* **2021**, 1–9. [CrossRef]
26. Ribeiro, D.A.; Silva, J.C.; Lopes Rosa, R.; Saadi, M.; Mumtaz, S.; Wuttisittikulij, L.; Zegarra Rodríguez, D.; Al Otaibi, S. Light Field Image Quality Enhancement by a Lightweight Deformable Deep Learning Framework for Intelligent Transportation Systems. *Electronics* **2021**, *10*, 1136. [CrossRef]
27. Rokach, L. Ensemble-Based Classifiers. *Artif. Intell. Rev.* **2010**, *33*, 1–39. [CrossRef]
28. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
29. Lara-Benitez, P.; Carranza-García, M.; Riquelme, J.C. An Experimental Review on Deep Learning Architectures for Time Series Forecasting. *Int. J. Neur. Syst.* **2021**, *31*, 2130001. [CrossRef] [PubMed]

30. Sun, Y.; Kamel, M.S.; Wong, A.K.C.; Wang, Y. Cost-Sensitive Boosting for Classification of Imbalanced Data. *Pattern Recognit.* **2007**, *40*, 3358–3378. [[CrossRef](#)]
31. Liu, X.Y.; Wu, J.; Zhou, Z.H. Exploratory Undersampling for Class-Imbalance Learning. *IEEE Trans. Syst. Man Cybern. B* **2009**, *39*, 539–550.
32. Zhu, C.; Wang, Z. Entropy-Based Matrix Learning Machine for Imbalanced Data Sets. *Pattern Recognit. Lett.* **2017**, *88*, 72–80. [[CrossRef](#)]
33. Wang, K.; Xue, Q.; Xing, Y.; Li, C. Improve Aggressive Driver Recognition Using Collision Surrogate Measurement and Imbalanced Class Boosting. *Int. J. Environ. Res. Public Health* **2020**, *17*, 2375. [[CrossRef](#)]
34. Sagi, O.; Rokach, L. Ensemble Learning: A Survey. *WIREs Data Min. Knowl. Discov.* **2018**, *8*, e1249. [[CrossRef](#)]
35. Saleh, K.; Hossny, M.; Nahavandi, S. Driving behavior classification based on sensor data fusion using LSTM recurrent neural networks. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–6.
36. Ordóñez, F.; Roggen, D. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors* **2016**, *16*, 115. [[CrossRef](#)] [[PubMed](#)]
37. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
38. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
39. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using rnn encoder–decoder for statistical machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 1724–1734.
40. Tsantekidis, A.; Passalis, N.; Tefas, A.; Kannianen, J.; Gabbouj, M.; Iosifidis, A. Forecasting stock prices from the limit order book using convolutional neural networks. In Proceedings of the 2017 IEEE 19th Conference on Business Informatics (CBI), Thessaloniki, Greece, 24–27 July 2017; pp. 7–12.
41. Hassan, A.; Mahmood, A. Convolutional Recurrent Deep Learning Model for Sentence Classification. *IEEE Access* **2018**, *6*, 13949–13957. [[CrossRef](#)]
42. Tian, Y.; Pan, L. Predicting short-term traffic flow by long short-term memory recurrent neural network. In Proceedings of the 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), Chengdu, China, 19–21 December 2015; pp. 153–158.
43. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent Trends in Deep Learning Based Natural Language Processing. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75. [[CrossRef](#)]
44. Fischer, T.; Krauss, C. Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions. *Eur. J. Oper. Res.* **2018**, *270*, 654–669. [[CrossRef](#)]
45. Sun, Z.; Song, Q.; Zhu, X.; Sun, H.; Xu, B.; Zhou, Y. A Novel Ensemble Method for Classifying Imbalanced Data. *Pattern Recognit.* **2015**, *48*, 1623–1637. [[CrossRef](#)]
46. Kohonen, T. The Self-Organizing Map. *Proc. IEEE* **1990**, *78*, 1464–1480. [[CrossRef](#)]
47. Alanazi, S.A.; Alruwaili, M.; Ahmad, F.; Alaerjan, A.; Alshammari, N. Estimation of Organizational Competitiveness by a Hybrid of One-Dimensional Convolutional Neural Networks and Self-Organizing Maps Using Physiological Signals for Emotional Analysis of Employees. *Sensors* **2021**, *21*, 3760. [[CrossRef](#)] [[PubMed](#)]
48. Malondkar, A.; Corizzo, R.; Kiringa, I.; Ceci, M.; Japkowicz, N. Spark-GHSOM: Growing Hierarchical Self-Organizing Map for Large Scale Mixed Attribute Datasets. *Inf. Sci.* **2019**, *496*, 572–591. [[CrossRef](#)]
49. Yotova, G.; Varbanov, M.; Tcherkezova, E.; Tsakovski, S. Water Quality Assessment of a River Catchment by the Composite Water Quality Index and Self-Organizing Maps. *Ecol. Indic.* **2021**, *120*, 106872. [[CrossRef](#)]
50. Betti, A.; Tucci, M.; Crisostomi, E.; Piazzini, A.; Barmada, S.; Thomopoulos, D. Fault Prediction and Early-Detection in Large PV Power Plants Based on Self-Organizing Maps. *Sensors* **2021**, *21*, 1687. [[CrossRef](#)]
51. Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 645–657. [[CrossRef](#)]
52. Li, X.; Ye, M.; Liu, Y.; Zhu, C. Adaptive Deep Convolutional Neural Networks for Scene-Specific Object Detection. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *29*, 2538–2551. [[CrossRef](#)]
53. Koprinska, I.; Wu, D.; Wang, Z. Convolutional neural networks for energy time series forecasting. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
54. Kuo, P.-H.; Huang, C.-J. A High Precision Artificial Neural Networks Model for Short-Term Energy Load Forecasting. *Energies* **2018**, *11*, 213. [[CrossRef](#)]
55. Wei, D.; Wang, B.; Lin, G.; Liu, D.; Dong, Z.; Liu, H.; Liu, Y. Research on Unstructured Text Data Mining and Fault Classification Based on RNN-LSTM with Malfunction Inspection Report. *Energies* **2017**, *10*, 406. [[CrossRef](#)]
56. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, arXiv:1412.3555.

-
57. Kittler, J.; Hatef, M.; Duin, R.P.W.; Matas, J. On Combining Classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 226–239. [[CrossRef](#)]
 58. Beyer, K.; Goldstein, J.; Ramakrishnan, R.; Shaft, U. When is “nearest neighbor” meaningful? In *Database Theory—ICDT’99*; Beeri, C., Buneman, P., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1999; Volume 1540, pp. 217–235, ISBN 978-3-540-65452-0.