

Research Article

Multiple Naïve Bayes Classifiers Ensemble for Traffic Incident Detection

Qingchao Liu,^{1,2} Jian Lu,^{1,2} Shuyan Chen,^{1,2} and Kangjia Zhao³

¹ Jiangsu Key Laboratory of Urban ITS, Southeast University, Nanjing 210096, China

² Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, Nanjing 210096, China

³ Department of Civil & Environment Engineering, National University of Singapore, Singapore 119078

Correspondence should be addressed to Jian Lu; lujian_1972@seu.edu.cn

Received 16 January 2014; Revised 26 March 2014; Accepted 27 March 2014; Published 28 April 2014

Academic Editor: Erik Cuevas

Copyright © 2014 Qingchao Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This study presents the applicability of the Naïve Bayes classifier ensemble for traffic incident detection. The standard Naive Bayes (NB) has been applied to traffic incident detection and has achieved good results. However, the detection result of the practically implemented NB depends on the choice of the optimal threshold, which is determined mathematically by using Bayesian concepts in the incident-detection process. To avoid the burden of choosing the optimal threshold and tuning the parameters and, furthermore, to improve the limited classification performance of the NB and to enhance the detection performance, we propose an NB classifier ensemble for incident detection. In addition, we also propose to combine the Naïve Bayes and decision tree (NBTree) to detect incidents. In this paper, we discuss extensive experiments that were performed to evaluate the performances of three algorithms: standard NB, NB ensemble, and NBTree. The experimental results indicate that the performances of five rules of the NB classifier ensemble are significantly better than those of standard NB and slightly better than those of NBTree in terms of some indicators. More importantly, the performances of the NB classifier ensemble are very stable.

1. Introduction

The functionality of automatically detecting incidents on freeways is a primary objective of advanced traffic management systems (ATMS), an integral component of the Nation's Intelligent Transportation Systems (ITS) [1]. Traffic incidents are defined as nonrecurring events such as accidents, disabled vehicles, spilled loads, temporary maintenance and construction activities, signal and detector malfunctions, and other special and unusual events that disrupt the normal flow of traffic and cause motorist delay [2, 3]. If the incident cannot be handled timely, it will increase traffic delay, reduce road capacity, and often cause second traffic accidents. Timely detection of incidents is critical to the successful implementation of an incident management system on freeways [4].

Incident detection is essentially a pattern classification problem, where the incident and nonincident traffic patterns are to be recognized or classified [5]. That is to say, incident detection can be viewed as a pattern recognition problem that

classifies traffic patterns into one of the two classes: nonincident and incident classes. The classification is normally based on spatial and temporal traffic pattern changes during an incident. The spatial pattern changes refer to the traffic pattern alterations over a stretch of a freeway. The temporal traffic pattern changes refer to the traffic pattern alterations over consecutive time intervals. Typically, traffic flow maintains a consistent pattern at upstream and downstream detector stations. When an incident occurs, however, traffic flow at the upstream of incident scene tends to be congested while that at the downstream station tends to be light due to the blockage at incident site. These changes in the traffic flow are reflected in the detector data obtained from both the upstream and downstream stations [5]. Therefore, an AID problem is essentially a classification problem. Any good classifier is a potential tool for the incident detection problem. Based on this idea, in our approach to incident detection in general, we treat the problem as one of pattern classification problems. Under normal traffic operation, traffic parameters (speeds,

occupancies, and volumes) both upstream and downstream from a given freeway site are expected to have more or less similar patterns in time (except under bottleneck situations). In the case of an incident, this normal pattern is disrupted. Patterns of incident develop increased occupancies and drop in speeds for instance.

Automated incident detection (AID) systems, which employ an incident detection algorithm to detect incidents from traffic data, aim to improve the accuracy and efficiency of incident detection over a large traffic network. Early AID algorithm development focused on simple comparison methods using raw traffic data [6, 7]. To enhance algorithm performance and to achieve real-time incident detection, advanced methods have been suggested, which include image processing [8], artificial neural networks [9], support vector machine [4], and data fusion [10]. Although these new published methods represent significant improvements, performance stability and transferability are still major issues concerning the existing AID algorithms. To enhance freeway incident detection and to fulfill the universality expectations for AID algorithms, the classic Bayesian theory has attracted many scientists' attention. Due to the Naïve Bayes ensemble and Naïve Bayes, and decision tree (NBTree) algorithm simplicity and easy interpretation, the applications of this approach can be found in an abundant literature. However, the report of its application to traffic engineering is rare. It provides ample motivation to investigate this model performance on incident detection.

There are drawbacks which limit its applications; the optimal threshold and parameter of Naïve Bayes (NB) have a great effect on the generalization performance, and setting the parameters of the NB classifier is a challenging task. At present, there is no structured method to choose them. Typically, the optimal threshold and parameters have to be chosen and tuned by trial and error. Some studies have applied search techniques for this problem; however, a large amount of computation time will still be involved in such search techniques, which are themselves computationally demanding.

A natural and reasonable question is whether we can increase or at least maintain NB performance, while, at the same time, avoiding the burden of choosing the optimal threshold and tuning the parameters. Some researchers have proposed classifier ensembles to address this problem. The performance of classifier ensemble has been investigated experimentally, and it appears to consistently give better results [11–13]. Kittler et al. [12] used different combining schemes based on multiple classifier ensembles, and six rules were introduced into ensemble learning to search for accurate and diverse classifiers to construct a good ensemble. The presented method works on a higher level and is more direct than other search based methods of ensemble learning. The previous research illustrated that ensemble techniques are able to increase the classification accuracy by combining their individual outputs [14, 15]. While these general methods are preexisting, their application into the specific problem and their integration into the proposed model for the detection of traffic incident are new. It is expected that the NB classifier ensemble has more generalization ability, but the method has

not been seen discussed in the context of incident detection. Coupling this expectation of reasonably accurate detection with the attractive implementation characteristics of the NB classifier ensemble, we propose to apply the NB classifier ensemble achieved by the combination of different rules to detect incidents. The NB classifier ensemble algorithm trains many individual NB classifiers to construct the classifier ensemble and then uses this classifier ensemble to detect the traffic incidents. It needs to train many times. Taking this into account, we also propose NBTree [16] for incident detection. The NBTree splits the dataset by applying an entropy-based algorithm and uses standard NB classifiers at the leaf node to handle attributes. The NBTree applies strategy to construct a decision tree and replaces leaf nodes with NB classifiers. We have performed some experiments to evaluate the performances of the standard NB, NB ensemble, and NBTree algorithms. The experimental results indicate that the performances of five rules of the NB classifier ensemble are significantly better than those of standard NB and slightly better than those of NBTree in terms of some indicators. More importantly, the performances of the NB classifier ensemble are very stable.

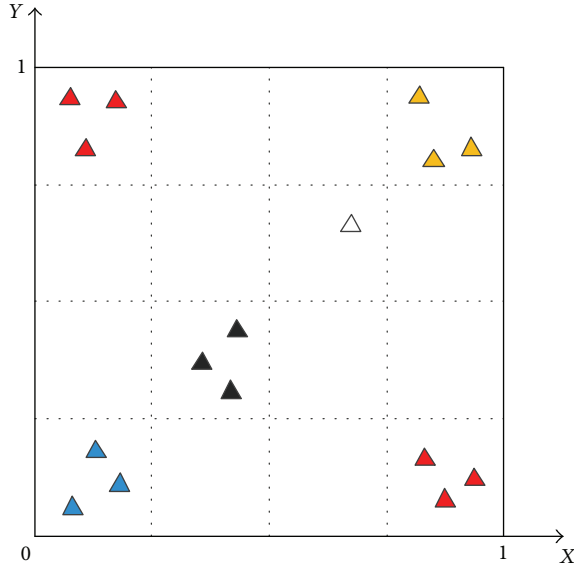
The remaining part of the paper is structured as follows. Section 2 introduces the combination schemes of the NB classifier ensemble. The general performance criteria of AID are presented in Section 3. Section 4 is devoted to empirical results. In this section, we evaluate the performances of the three algorithms: standard NB, NB ensemble, and NBTree. Finally, the conclusions are drawn in Section 5.

2. Naïve Bayes Classifier and Combining Schemes

2.1. Classification and Classifier. A dataset generally consists of feature vectors, where each feature vector is a description of an object by using a set of features. For example, take a look at the synthetic dataset as shown in Figure 1. Here, each object is a data point described by the features x -coordinate, y -coordinate, and color, and a feature vector looks like (0.8, 0.9, yellow) or (0.9, 0.1, red). Features are also called attributes, a feature vector is also called an instance, and sometimes a dataset is called a sample.

Naïve Bayes classifier ensemble is a predictive model that we want to construct or discover from the dataset. The process of generating models from data is called learning or training, which is accomplished by a learning algorithm. In supervised learning, the goal is to predict the value of a target feature on unseen instances, and the learned model is also called a predictor. For example, if we want to predict the color of the synthetic data points, we call “yellow” and “red” labels, and the predictor should be able to predict the label of an instance for which the label information is unknown, for example, (0.7, 0.7). If the label is categorical, such as color, the task is called classification and the learner is also called classifier.

2.2. Naïve Bayes Classifier. To classify a test instance x , one approach is to formulate a probabilistic model to estimate the posterior probability $P(\omega | x)$ of different ω 's and predict



Instance: (X, Y; color)

- (1) If $(X < 0.25 \text{ and } Y > 0.75)$ or $(X > 0.75 \text{ and } Y < 0.25)$ then \rightarrow ▲
- (2) If $(X > 0.75 \text{ and } Y > 0.75)$ then \rightarrow ▲
- (3) If $(X < 0.25 \text{ and } Y < 0.25)$ then \rightarrow ▲
- (4) If $(X \in [0.25, 0.50] \text{ and } Y \in [0.25, 0.50])$ then \rightarrow ▲

FIGURE 1: The synthetic dataset.

the one with the largest posterior probability; this is the maximum a posteriori (MAP) rule. By Bayes Theorem, we have

$$P(\omega | x) = \frac{P(\omega | x) P(\omega)}{P(x)}, \quad (1)$$

where $P(\omega)$ can be estimated by counting the proportion of class ω in the training set and $P(x)$ can be ignored since we are comparing different ω 's on the same x . Thus we only need to consider $P(x | \omega)$. If we can get an accurate estimate of $P(x | \omega)$, we will get the best classifier in theory from the given training data, that is, the Bayes optimal classifier with the Bayes error rate, the smallest error rate in theory. However, estimating $P(x | \omega)$ is not straightforward, since it involves the estimation of exponential numbers of joint-probabilities of the features. To make the estimation tractable, some assumptions are needed. The naive Bayes classifier assumes that, given the class label, the n features are independent of each other within each class. Thus, we have

$$P(x | \omega) = \prod_{i=1}^n P(x_i | \omega) \quad (2)$$

which implies that we only need to estimate each feature value in each class in order to estimate the conditional probability, and therefore the calculation of joint-probabilities is avoided. In the training stage, the naive Bayes classifier estimates

the probabilities $P(\omega)$ for all classes $\omega \in \omega$ and $P(x_i | \omega)$ for all features $i = 1, 2, \dots, n$ and all feature values x_i from the training set. In the test stage, a test instance will be predicted with label ω if ω leads to the largest value of all the class labels

$$P(\omega | x) \propto P(\omega) \prod_{i=1}^n P(x_i | \omega). \quad (3)$$

As demonstrated in paper [17], for a threshold level of 0.0006, 65.4% of incidents were identified. However, if the threshold cannot be chosen appropriately, fewer incidents will be identified correctly. As known from machine learning, the naive Bayesian classifier provides a simple approach, with clear semantics, for representing, using, and learning probabilistic knowledge. The method is designed for use in supervised induction tasks, in which the performance goal is to accurately predict the class of test instances and in which the training instances include class information. In this way, it avoids choosing the optimal threshold and tuning the parameters manually.

2.3. Combining Schemes

2.3.1. Five Rules. The most widely used probability combination rules [12] are the product rule, the sum rule, the min rule, and the max rule. Given N classifiers and c_1, \dots, c_N , classes $\omega_1, \dots, \omega_K$, these are defined as follows.

Product rule:

$$P(\omega_k | \mathbf{x}_1, \dots, \mathbf{x}_N) = \frac{1}{P(\omega_k)} \prod_{n=1}^N P(\omega_k | \mathbf{x}_n), \quad (4)$$

where \mathbf{x}_n is the input to the n 'th classifier and $P(\omega_k)$ is the a priori probability for class ω_k .

Sum rule:

$$P(\omega_k | \mathbf{x}_1, \dots, \mathbf{x}_N) = \frac{1}{N} \sum_{n=1}^N P(\omega_k | \mathbf{x}_n). \quad (5)$$

Min rule:

$$P(\omega_k | \mathbf{x}_1, \dots, \mathbf{x}_N) = \frac{\min_n P(\omega_k | \mathbf{x}_n)}{\sum_{k=1}^K \min_n P(\omega_k | \mathbf{x}_n)}. \quad (6)$$

Max rule:

$$P(\omega_k | \mathbf{x}_1, \dots, \mathbf{x}_N) = \frac{\max_n P(\omega_k | \mathbf{x}_n)}{\sum_{k=1}^K \max_n P(\omega_k | \mathbf{x}_n)}. \quad (7)$$

Majority vote rule:

$$\Delta_{ki} = \begin{cases} 1 & \text{if } P(\omega_k | \mathbf{x}_i) = \max_{j=1}^K P(\omega_j | \mathbf{x}_i) \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

$$\sum_{i=1}^N \Delta_i = \max_{k=1}^K \sum_{i=1}^N \Delta_{ki}.$$

Note that for each class ω_k , the sum on the right hand side of (8) simply counts the votes received for this hypothesis from the individual classifiers. The class that receives the largest number of votes is then selected as the majority decision.

2.3.2. Combine Decision Tree and Naïve Bayes. The Naïve Bayesian tree (NBTree) algorithm is similar to the classical recursive partitioning schemes, except that the leaf nodes created are naïve Bayesian classifiers instead of nodes predicting a single class [16]. First, define a measure called entropy that characterizes the purity of an arbitrary collection of instances. Given a collection S , if the target attribute can take on c different values, then the entropy of S relative to this c -wise classification is defined as

$$\text{Entropy}(S) = -\sum_{i=1}^c p_i \log_2(p_i), \quad (9)$$

where p_i is the proportion of S belonging to class i . The information gain, $\text{Gain}(S, A)$, of an attribute A , the expected reduction in entropy caused by partitioning the examples according to this attribute relative to S , is defined as

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{value}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v), \quad (10)$$

where $\text{value}(A)$ is the set of all possible values for attributes A and S_v is the subset of S for which attribute A has value v . NBTree is a hybrid approach that attempts to utilize the advantage of both decision trees and naïve Bayesian classifiers. It splits the dataset by applying an entropy-based algorithm and uses standard naïve Bayesian classifiers at the leaf node to handle attributes. NBTree applies strategy to construct a decision tree and replaces leaf nodes with NB classifiers.

3. Performance Criteria of Aid

3.1. Definition of DR, FAR, MTTD, and CR. Four primary measures of performance, namely, detection rate (DR), false alarm rate (FAR), mean time to detection (MTTD), and classification rate (CR), are used to evaluate traffic incident detection algorithms. We will quote the definitions from [18, 19].

DR is defined as the number of incidents correctly detected by the traffic incident detection algorithm divided by the total number of incidents known to have occurred during the observation period:

$$\text{DR} = \frac{\text{number of incident cases detected}}{\text{total number of incident cases}} \times 100\%. \quad (11)$$

FAR is defined as the proportion of instances that were incorrectly classified as incident instances based on the total instances in the testing set. Out of the total number of applications of the model to the dataset, FAR is calculated to determine how many incident alarms were falsely set. In order to decrease FAR, persistent test is often used. An incident alarm is triggered whenever n consecutive outputs of the model exceed the threshold, which is called persistent check of n

$$\text{FAR} = \frac{\text{number of false alarm cases}}{\text{total number of non-incident cases}} \times 100\%. \quad (12)$$

MTTD is computed as the average length of time between the start of the incident and the time the alarm is initiated. When multiple alarms are declared for a single incident, only the first correct alarm is used for computing the detection rate and the mean time to detect

$$\text{MTTD} = \frac{t_1 + t_2 + \cdots + t_i + \cdots + t_m}{m}. \quad (13)$$

Besides these three measures, we also use classification rate (CR) as an index to test traffic incident detection algorithms. Of the total number of applications of cycle length data or input instances, the percentage of correctly classified instances (including both incident and nonincident instances) by the model is computed as CR

$$\text{CR} = \frac{\text{number of instances correctly classified}}{\text{total number of instances}} \times 100\%. \quad (14)$$

3.2. Area under the ROC Curve (AUC). Receiver operator characteristic (ROC) curves illustrate the relationship between the DR and the FAR from 0 to 1. Often the comparison of two or more ROC curves consists of either looking at the area under the ROC curve (AUC) or focusing on a particular part of the curves and identifying which curve dominates the other in order to select the best-performing algorithm. AUC, when using normalized units, is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (assuming “positive” ranks higher than “negative”) [20]. It can be shown that the area under the ROC curve is closely related to the Mann-Whitney U , which tests whether positives are ranked higher than negatives. It is also equivalent to the Wilcoxon test of ranks [21]. The AUC is related to the Gini coefficient (G_1) by the formula [22]. In this way, it is possible to calculate the AUC by using an average of a number of trapezoidal approximations:

$$\text{AUC} = \frac{G_1 + 1}{2}, \quad (15)$$

where $G_1 = 1 - \sum_{k=1}^n (X_k - X_{k-1})(Y_k + Y_{k-1})$.

3.3. Statistics Indicators. In statistics, the mean absolute error (MAE) is a quantity used to measure how close forecasts or predictions are to the eventual outcomes. The mean absolute error is given by

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{Y}_i - Y_i|. \quad (16)$$

The root-mean-square error (RMSE) is a frequently used measure of the differences between values predicted by a model or an estimator and the values actually observed. These individual differences are called residuals when the calculations are performed over the data sample that was used for estimation and are called prediction errors when computed out of sample. The RMSE serves to aggregate

the magnitudes of the errors in predictions for various times into a single measure of predictive power:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2}. \quad (17)$$

The equality coefficient (EC) is useful for comparing different forecast methods; for example, whether a fancy forecast is in fact any better than a naïve forecast repeating the last observed value. The closer the value of EC is to 1, the better the forecast method. A value of zero means the forecast is no better than a naïve guess:

$$\text{EC} = 1 - \frac{\sqrt{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}}{\sqrt{\sum_{i=1}^n Y_i^2 + \sum_{i=1}^n \hat{Y}_i^2}}. \quad (18)$$

Kappa measures the agreement between two raters who each classify N items into C mutually exclusive categories. Kappa is computed as formula (19), $P(A)$ is the observed agreement among the raters, and $P(E)$ is the expected agreement; that is, $P(E)$ represents the probability that the raters agree by chance. The values of Kappa are constrained to the interval $[-1, 1]$. Kappa = 1 means perfect agreement, Kappa = 0 means that agreement is equal to chance, and Kappa = -1 means “perfect” disagreement:

$$\text{Kappa} = \frac{P(A) - P(E)}{1 - P(E)}. \quad (19)$$

4. Experiments on Traffic Incident Detection

4.1. Parameters and Procedures of Experiments. To describe the experiments clearly, we first present the definitions for all parameters and symbols used in experiments. Then, we describe the experiment procedures in detail.

4.1.1. Parameters of Experiments. Some parameters are adopted to make the procedures of the experiments more automatic and optimized. In addition, some symbols are used to denote specified conceptions. For clarity, we have presented the definitions of each parameter and symbol in Table 1.

4.1.2. Construction of Datasets for Training and Testing. The traffic data mentioned refers to three basic traffic flow parameters, namely, volume, speed, and occupancy. The incident is detected based on section, which means that the traffic data collected from the upstream and the downstream detection stations are usually used as model inputs in AID systems. In Figure 2, A1, namely, the first attribute, so the number of X -variables (predictor variables) is 6. This means that the matrix X used in training the model has the size $n_S \times 6$. The test data X form a matrix of size $n_T \times 6$. The formal description of matrix X and Y can be written as shown in Figure 2.

One instance consists of at least the following items:

- (i) speed, volume, and occupancy of the upstream detector,

- (ii) speed, volume, and density of the downstream detector,
- (iii) traffic state (incident or nonincident),

where the item “traffic state” is a label. The value of the label is -1 or 1 , referring to nonincident or incident, respectively, which is determined by the incident dataset. Typically, the model is fit for part of the data (the training set), and the quality of the fit is judged by how well it predicts the other part of the data (the test set). The entire dataset was divided into two parts: a training set that was used to build the model and a test set that was used to test the model’s detection ability. Where each row is composed of one observation, n is the number of instances and $y_i \in \{-1, 1\}$. The data analysis problem is to relate the matrix Y as some function of the matrix X to predict Y (e.g., traffic state) using the data of X , $y = f(x)$. The training set was used to develop a Naïve Bayes classifier ensemble that was, in turn, used to detect incidents for the test set samples. The output values of the detection models were then compared with the actual ones for each of the calibration samples, and the performance criteria were calculated and compared.

4.1.3. Experiments Procedures. The experiments were performed according to the procedures as shown in Figure 3.

Step 1. Divide the whole dataset D into training set S and test set T . We take part of the whole dataset D as the training set S , the other as test set T and use the parameter n_s to control the size of the training set S , and S is obtained by taking out n_s samples from the front to the back of D .

Step 2. Perform sampling with replacement from training set S n times and then obtain the training subsets S_1, S_2, \dots, S_n . We use the parameter r to control the ratio of the number of samples in the training subset to the number of samples in the training set; that is, the number of samples in the training subset is $n_{\text{sub}} = n_s \times r$. The range of r is $[0, 1]$.

Step 3. While $k = 1, 2, 3, \dots, n$, perform the following:

- (1) use the training subset S_k to train the k th individual NB classifier INBC_k ;
- (2) use the training subset S_k to train the k th NBTree classifier INBTreeC_k ;
- (3) use the all the k existing individual NB classifiers to construct the k th ensemble NB classifier ENBC_k .

Step 4. While $k = 1, 2, 3, \dots, n$, perform the following:

- (1) test the performances of the k th individual NB classifier INBC_k on test set T ;
- (2) test the performances of the k th NBTree classifier INBTreeC_k on test set T ;
- (3) test the performances of the k th ensemble NB classifier ENBC_k on test set T .

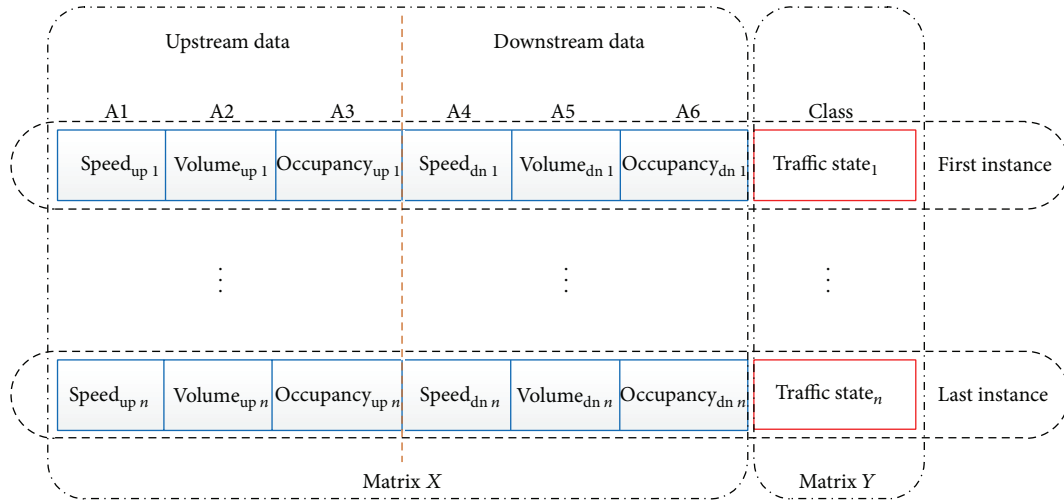


FIGURE 2: Construction of traffic datasets.

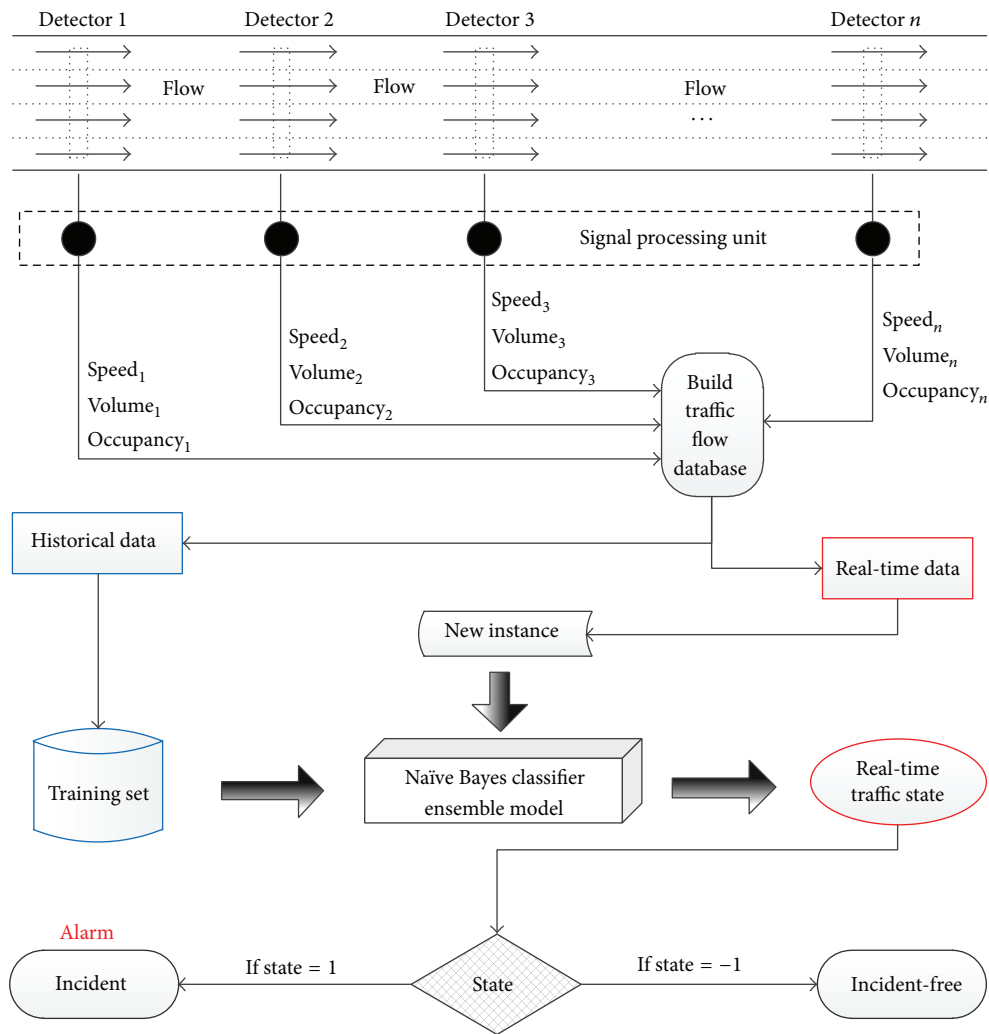


FIGURE 3: A freeway incident detection model based on Naïve Bayes classifier ensemble.

TABLE 1: Definition of Symbols and Parameters.

Parameter/Symbol	Definition	
Symbol	D	The whole dataset
	S	Training set
	T	Test set
	S_k	The k th subset of training set
	$INBC_k$	The k th individual Naïve Bayes classifier
	$ENBC_k$	The k th ensemble Naïve Bayes classifier
Parameter	$INBTreeC_k$	The k th individual NBTree classifier
	n	The total number of the training subsets
	r	The ration of n_{sub} to n_S
	n_T	The number of samples in test set T
	n_S	The number of samples in training set S
	n_{all}	The total number of samples in the whole dataset D
	n_{sub}	The number of samples in the training subset (each training subset has the same number of samples)
	$n_{incident}$	The number of incident samples in the whole dataset
	$n_{nonincident}$	The number of non-incident samples in the whole dataset

TABLE 2: Parameter Setting of Experiments.

Parameter Setting of Experiments for I-880 Dataset								
Parameter	n	r	n_T	n_S	n_{sub}	n_{all}	$n_{incident}$	$n_{nonincident}$
Value	20	0.05	45138	45518	2275	90656	4136	86520
Parameter Setting of Experiments for AYE Dataset								
Parameter	n	r	n_T	n_S	n_{sub}	n_{all}	$n_{incident}$	$n_{nonincident}$
Value	20	0.05	16000	13500	675	29500	6000	23500

4.2. Experiments on I-880 Dataset

4.2.1. Data Description for I-880 Dataset. We proceeded to real world data. These data were collected by Petty et al. from the I-880 Freeway in the San Francisco Bay area, California, USA. This is the most recent and probably the most well-known freeway incident dataset collected, and these data have been used in many studies related to incident detection. Loop detector data, with and without incident, was collected from a 9.2 miles (14.8 km) segment of the I-880 Freeway between the Marina and Wipple exits, in both directions. There were 18 loop detector stations in the northbound direction and 17 stations in the southbound direction. The data collected included traffic volume, occupancy, and speed, averaged across all lanes in 30 s intervals at the same station. In summary, the training dataset has 45,518 training instances, of which 2100 are incident instances (from 22 incident cases). The testing dataset has 45,138 instances in all, including 2036 incident instances (from 23 incident cases). Thus, incident examples are very rare in this dataset, as only approximately 4.6% and 4.5% incident examples are contained in the training set and the testing set, respectively. Each instance has seven features. In addition to the measurements of speed, volume, and occupancy collected at both the upstream detector station and the downstream detector station, the last one is the class label, -1 for nonincident state and 1 for incident state.

4.2.2. Parameter Setting of Experiments for I-880 Dataset. To divide n_S into n_{sub} properly, there are some parameters that need to be set, which can be seen in Table 2. We set values for each parameter and present the results in Table 2.

In our experiments, we constructed 20 individual NB classifiers, 20 NB ensemble classifiers, and 20 NBTree classifiers. We tested the performances of the five rules of each classifier on the I-880 dataset. Then, we calculated the averages and variances of the performances of the 20 individual NB classifiers, 20 NB ensemble classifiers, and 20 NBTree classifiers. The summarized results are in Table 3. In Table 3, the results are presented with the form average \pm variance, and the best results are highlighted in bold. To make a visual comparison of the performances of all classifiers, we plotted them in Figures 4 and 5.

4.3. Experiments on AYE Dataset with Noisy Data

4.3.1. Data Description for AYE Dataset. The traffic data used in this study for the development of the incident detection models was produced from a simulated traffic system. A 5.8 km section of the Ayer Rajah Expressway (AYE) in Singapore was selected to simulate incident and nonincident conditions. This site was selected for incident detection study because of its diverse geometric configurations that can cover a variety of incident patterns [23, 24].

TABLE 3: Experimental Results of NB, Five Rules and NBTree Based as Applied to the I-880 Dataset (The performances are presented in the form average \pm variance).

Algorithm	DR	FAR	MTTD	CR	AUC	Kappa	MAE	RMSE	EC
Naïve Bayes classifier	$0.8228 \pm 2.50E - 05$	$0.0398 \pm 6.32E - 07$	1.2991 ± 0.01727	$0.9540 \pm 6.43E - 07$	$0.8915 \pm 6.50E - 06$	$0.59479 \pm 2.63E - 05$	$0.01619 \pm 3.11E - 07$	$0.17995 \pm 9.47E - 06$	$0.99184 \pm 8.03E - 08$
Product rule ensemble	$0.8404 \pm 2.91E - 04$	$0.0380 \pm 3.03E - 06$	1.7615 ± 1.04744	$0.9557 \pm 2.22E - 06$	$0.9012 \pm 7.78E - 05$	$0.61383 \pm 2.28E - 04$	$0.01602 \pm 5.33E - 07$	$0.17898 \pm 1.60E - 05$	$0.99192 \pm 1.38E - 07$
Sum rule ensemble	$0.8963 \pm 5.47E - 06$	$0.0297 \pm 5.87E - 07$	1.6906 ± 0.89473	$0.9636 \pm 3.66E - 07$	$0.9333 \pm 1.06E - 06$	$0.6932 \pm 1.91E - 05$	$0.01599 \pm 2.04E - 07$	$0.17881 \pm 6.34E - 06$	$0.99194 \pm 5.26E - 08$
Max rule ensemble	$0.8960 \pm 5.58E - 06$	$0.0297 \pm 1.62E - 07$	1.6236 ± 0.84537	$0.9636 \pm 1.63E - 07$	$0.9331 \pm 1.33E - 06$	$0.69268 \pm 6.81E - 06$	$0.01606 \pm 8.09E - 08$	$0.17924 \pm 2.52E - 06$	$0.99190 \pm 2.09E - 08$
Min Rule ensemble	$0.8962 \pm 2.29E - 06$	$0.0304 \pm 2.99E - 06$	1.6193 ± 0.85452	$0.9629 \pm 3.13E - 06$	$0.9329 \pm 2.53E - 06$	$0.68845 \pm 1.28E - 04$	$0.01603 \pm 1.06E - 07$	$0.17902 \pm 3.28E - 06$	$0.99192 \pm 2.73E - 08$
MV rule ensemble	$0.8193 \pm 6.25E - 05$	$0.0410 \pm 2.72E - 06$	1.7802 ± 1.15158	$0.9527 \pm 2.02E - 06$	$0.8892 \pm 1.29E - 05$	$0.58639 \pm 5.16E - 05$	$0.01630 \pm 5.09E - 07$	$0.18052 \pm 1.52E - 05$	$0.99178 \pm 1.32E - 07$
NBTree classifier	0.8143 ± 0.00112	$0.0089 \pm 1.463E - 5$	1.3622 ± 0.00798	$0.9831 \pm 1.474E - 5$	$0.9027 \pm 2.7953E - 4$	0.8052 ± 0.00147	$0.02626 \pm 6.3487E - 6$	$0.22894 \pm 1.202E - 4$	$0.98669 \pm 1.67E - 6$

The simulation system generated volume, occupancy, and speed data at upstream and downstream sites for both incident and nonincident traffic conditions. The traffic dataset consisted of 300 incident cases that had been simulated based on AYE traffic. The simulation of each incident case consisted of three parts. The first part was the nonincident period that lasted for 5 min. This was after a simulation of a 5 min warm-up time. During the warm-up time, the data contains noise. The second part was the 10 min incident period. This was followed by a 30 min postincident period. Each input pattern included traffic volume, speed, and lane occupancy accumulated at 30 s intervals, averaged across all the lanes, as well as the traffic state. The value of the traffic state label is -1 or 1 , referring to nonincident or incident states, respectively.

4.3.2. Parameter Setting of Experiments for the AYE Dataset. As we used a new dataset to perform the experiments, the parameter values of the experiments needed to be updated. The updated parameter values can be seen in Table 2.

4.3.3. Experimental Results with the AYE Dataset. As mentioned in Section 4.3.1, the AYE dataset includes noisy data, which seriously reduces the quality of the detection. Therefore, the experimental results obtained with the AYE dataset are much worse than the experimental results from the I-880 dataset overall. The AYE dataset experimental results are summarized in Figures 6 and 7 and Table 4. In Table 4, for each algorithm of standard NB, NB ensemble, and NBTree, we calculate the averages and variances of the performances of the total 20 individual classifiers or ensemble classifiers. The results are presented as average \pm variance, and the best results are highlighted in bold.

4.4. Performance Evaluation. In this subsection, we have evaluated the performances of all three algorithms, standard NB, NB ensemble, and NBTree, using the I-880 dataset and the AYE dataset with noisy data. In Figures 4 and 6, the subfigures (a)~(f) evaluate the performances of five rules on the indicators DR, FAR, MTTD, CR, AUC, and Kappa. In Figures 5 and 7, the subfigures evaluate the performances of MAE, RMSE, and EC.

4.4.1. Performance Evaluation for the I-880 Dataset. From Figures 4 and 5, we can see that the performances of the sum rule, the max rule, and the min rule are stable and significantly better than the performances of the product rule and the majority vote rule. The performances of the product rule and the majority vote rule fluctuate violently dynamically, which demonstrates that the performances are unstable. The reason for this is that the results of the Naïve Bayes algorithm depend on the appropriate threshold and parameters, and the algorithm is not resilient to estimation errors. For example, in Figure 4(a), the DR of the product rule and the majority vote rule reaches 88% and 82% or even higher. In contrast, DR can also reach as low as 78% or even lower. In Figure 5 concerning the product rule, when the number of classifiers is less than 10, the RMSE reaches as low as 0.17 or even lower. In contrast, the RMSE

can reach 0.19 or even higher. These results indicate that when the threshold and parameters are chosen appropriately, the detection performance can improve. We need to select an appropriate threshold and parameters to make the NB ensemble achieve optimal performances, But the procedures for choosing an appropriate threshold and parameters need to use the trial and error method. Until now, there has not been a structured way to choose these values. There is another significant phenomenon that can be found from the data in Figures 4 and 5. When the number of classifiers increases, the performance of the five rules mitigates their fluctuation, and they tend to achieve a stable value. The reason for this is that the multiple classifiers ensemble can compensate for the defect of a single classifier to some extent. From Figure 4, we can also see that the performances of the sum rule, the max rule, and the min rule on each indicator are very close to each other. As for the indicators DR, FAR, AUC, and Kappa, the performance of the sum rule is slightly better than that of the max rule and the min rule. As mentioned above, AUC and Kappa can evaluate the performances more comprehensively than can the other four indicators. To make an overall evaluation, the performances of the NB ensemble are slightly better than those of the Naïve Bayes.

In Table 3, the average DR value of Naïve Bayes is 82.28%, whereas the average DR value of the five rules ranges from 81.93% to 89.63%. This indicates that the NB ensemble algorithm is more sensitive to the traffic incidents and can detect more traffic incidents than can the standard Naïve Bayes algorithm. The average MTTD value of NBTree is 1.36 min, which indicates that the NBTree algorithm can detect the incidents more quickly than NB ensemble algorithm. The average value of CR of NBTree is 98.31%, which indicates that NBTree can achieve the higher classification accuracy of the incident instance than can the Naïve Bayes and NB ensemble algorithm. The average Kappa value of NBTree is 0.8052, which indicates that the performance of NBTree classifier is significantly better than those of the other classifiers. In addition, it can detect more incidents than the NB ensemble, but the experimental results are not the same. The reason for this is that the FAR of NBTree is the highest. FAR improves the performance of the NBTree classifier. The MAE, RMSE, and EC of the product rule are best, which indicates that the predicted value of the product rule is the closest approximation.

4.4.2. Performance Evaluation Using AYE Dataset with Noisy Data. From Figures 6 and 7, we can see that when the dataset includes noisy data, the performances of the sum rule, the max rule, and the min rule are still stable and significantly better than the performances of the product rule and the majority vote rule. In addition, we also find that the performances of the sum rule, the max rule, and the min rule are much better than those of the product rule and the majority vote rule on the indicators of DR, FAR, CR, and Kappa. Among the five rules, it appears that the min rule yields the highest average MAE, whereas the other rules are similar. As far as EC is concerned, the five rules perform at a similar level that is better than that of the NBTree. The opposite case

TABLE 4: Experimental Results of NB, Five Rules and NBTree Based as Applied to the AYE Dataset (The performances are presented in the form average \pm variance).

Algorithm	DR	FAR	MTTD	CR	AUC	Kappa	MAE	RMSE	EC
Naïve Bayes classifier	0.7112 \pm 0.00348	0.0499 \pm 8.16E - 04	1.394 \pm 0.08869	0.9094 \pm 0.00111	0.8663 \pm 4.39E - 04	0.6247 \pm 9.14E - 04	0.16769 \pm 3.07E - 04	0.57895 \pm 5.31E - 04	0.90854 \pm 1.83E - 04
Product rule ensemble	0.6457 \pm 0.00127	0.0557 \pm 2.40E - 06	1.8685 \pm 1.24139	0.8732 \pm 1.64E - 06	0.8619 \pm 4.85E - 07	0.7479 \pm 3.58E - 05	0.16690 \pm 1.04E - 06	0.57775 \pm 3.11E - 06	0.90895 \pm 3.68E - 07
Sum rule ensemble	0.7923 \pm 5.57E - 05	0.0500 \pm 6.73E - 07	1.4384 \pm 0.63868	0.8774 \pm 3.55E - 07	0.8667 \pm 3.75E - 07	0.7439 \pm 1.29E - 04	0.16739 \pm 4.94E - 07	0.57861 \pm 1.47E - 06	0.90866 \pm 1.75E - 07
Max rule ensemble	0.7869 \pm 1.79E - 04	0.0500 \pm 2.21E - 06	1.4571 \pm 0.72272	0.8771 \pm 1.25E - 06	0.8663 \pm 2.02E - 07	0.7511 \pm 4.12E - 05	0.16803 \pm 4.96E - 06	0.57969 \pm 1.46E - 05	0.90828 \pm 1.77E - 06
Min Rule ensemble	0.7960 \pm 1.25E - 04	0.0498 \pm 2.76E - 06	1.4646 \pm 0.73539	0.8781 \pm 1.10E - 06	0.8667 \pm 1.14E - 06	0.6052 \pm 1.86E - 06	0.16638 \pm 3.25E - 06	0.57684 \pm 9.83E - 06	0.90926 \pm 1.15E - 06
MV rule ensemble	0.6246 \pm 1.15E - 05	0.0572 \pm 2.29E - 06	1.8358 \pm 1.19124	0.8720 \pm 3.84E - 07	0.7837 \pm 1.18E - 06	0.7341 \pm 5.31E - 05	0.16687 \pm 2.27E - 06	0.57769 \pm 6.87E - 06	0.90897 \pm 8.03E - 07
NBTree classifier	0.7275 \pm 3.25E - 4	0.0250 \pm 1.85E - 5	1.2103 \pm 0.10383	0.9031 \pm 3.52E - 5	0.8512 \pm 1.04E - 4	0.7085 \pm 3.26E - 4	0.1711 \pm 6.42E - 5	0.49189 \pm 2.62E - 4	0.90553 \pm 2.06E - 5

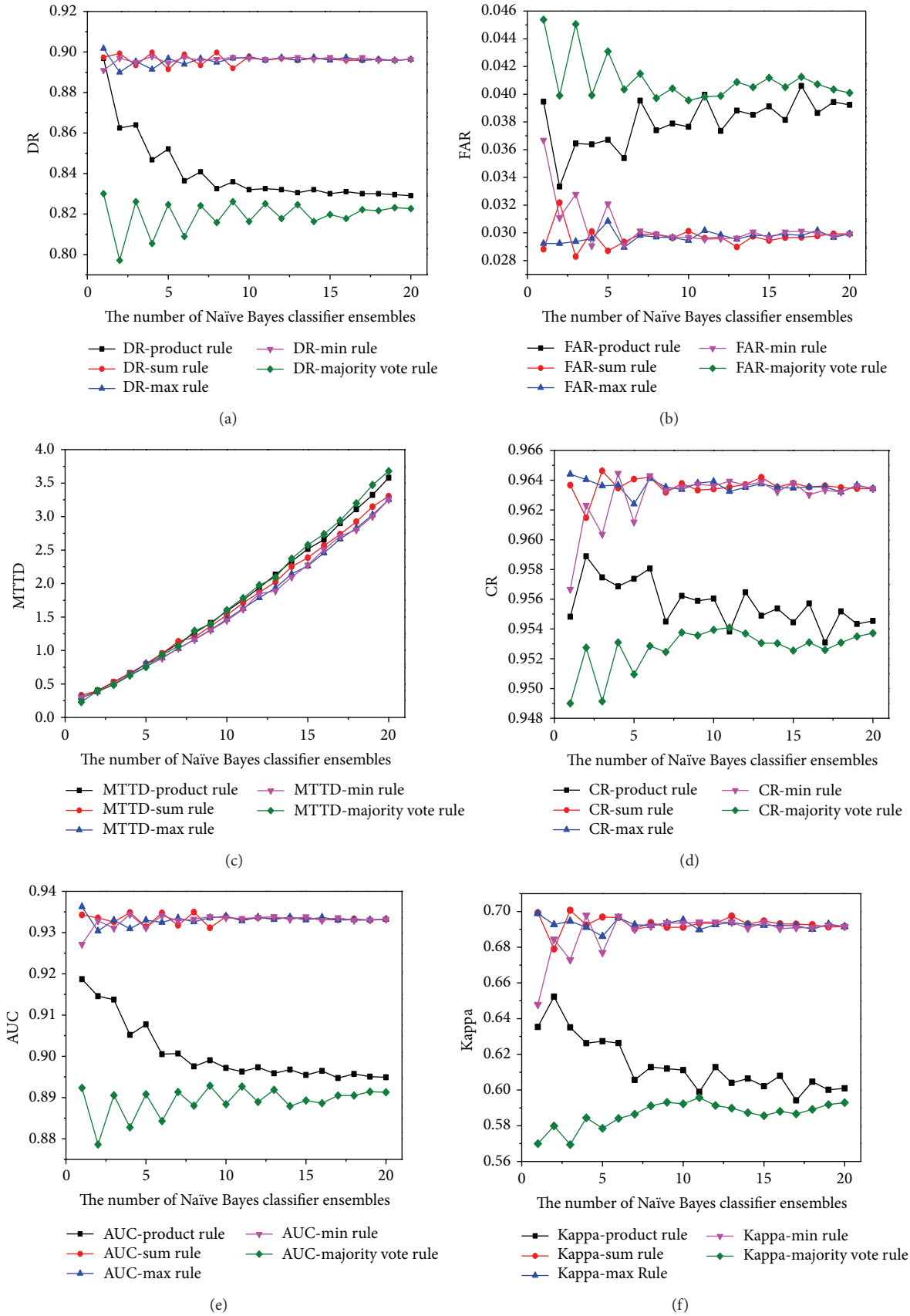


FIGURE 4: Experimental results of five rules for the Naïve Bayes ensembles as applied to the I-880 dataset: (a) performance with DR; (b) performance with FAR; (c) performance with MTTD; (d) performance with CR; (e) performance with AUC; (f) performance with Kappa.

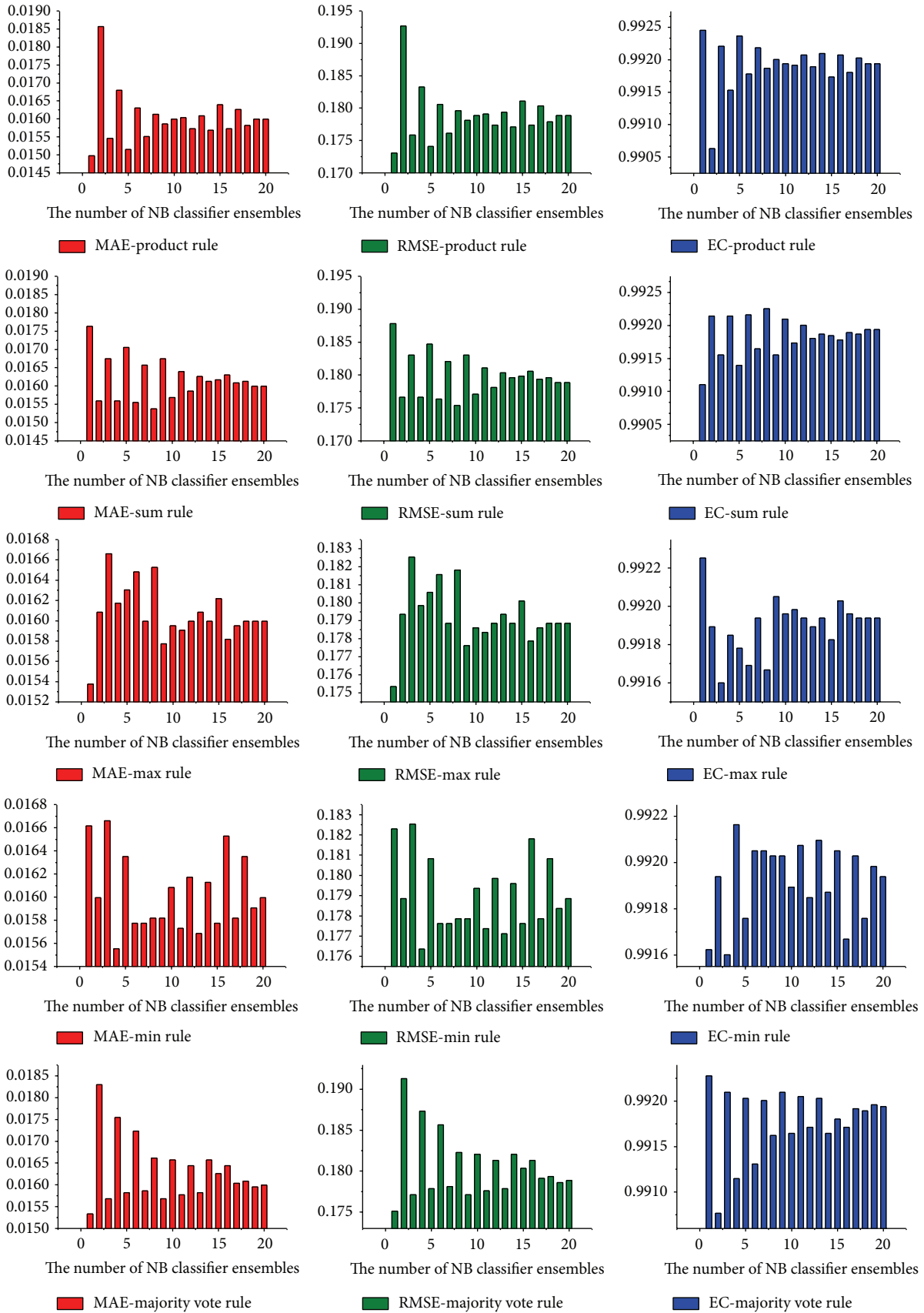


FIGURE 5: Bar chart comparison of five rules for Naïve Bayes classifier ensembles as applied to the I-880 dataset. The red bar is MAE, the green bar is RMSE, and the blue bar is EC.

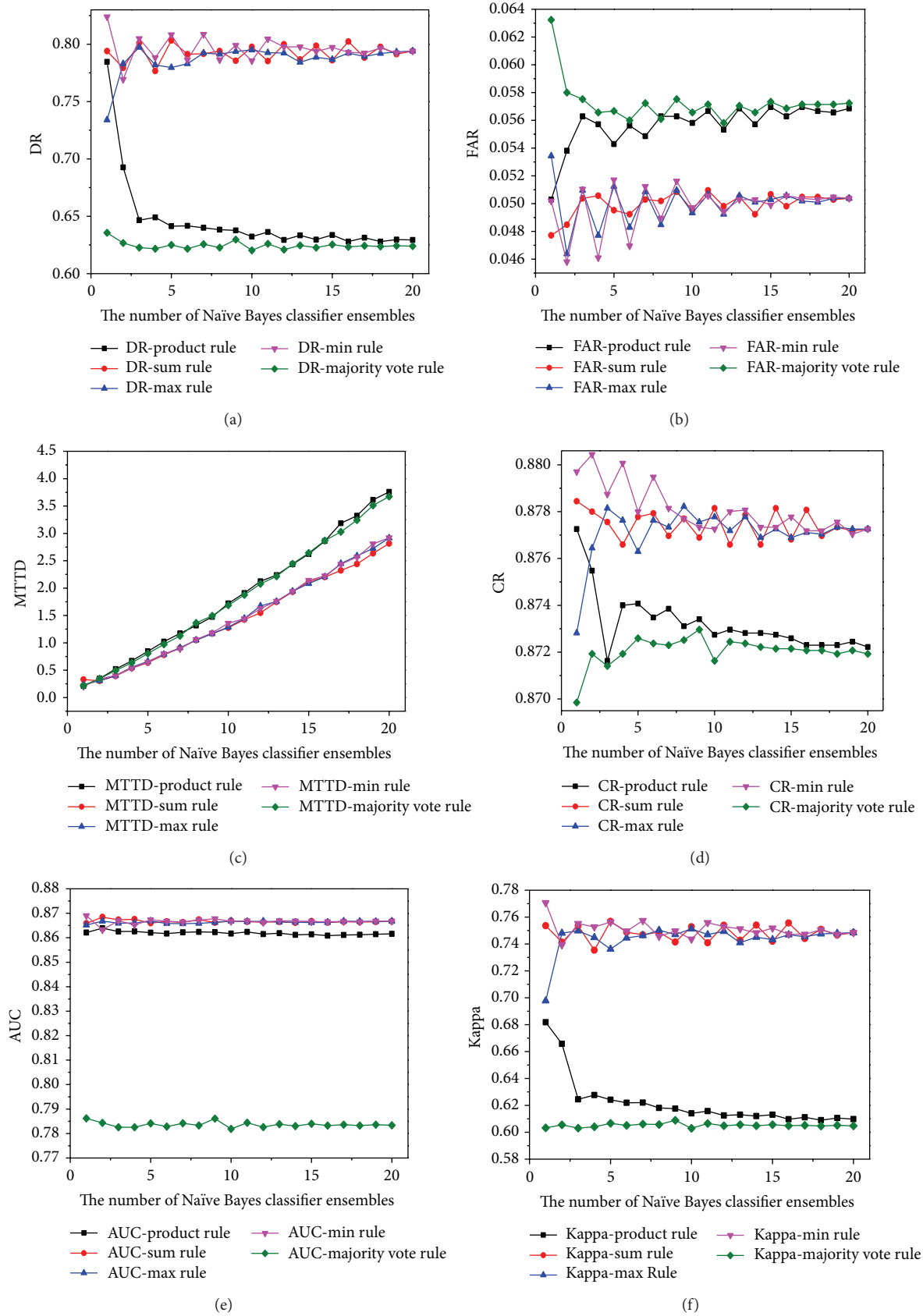


FIGURE 6: Experimental results of five rules for Naïve Bayes ensemble as applied to the AYE dataset: (a) performance with DR; (b) performance with FAR; (c) performance with MTTD; (d) performance with CR; (e) performance with AUC; (f) performance with Kappa.

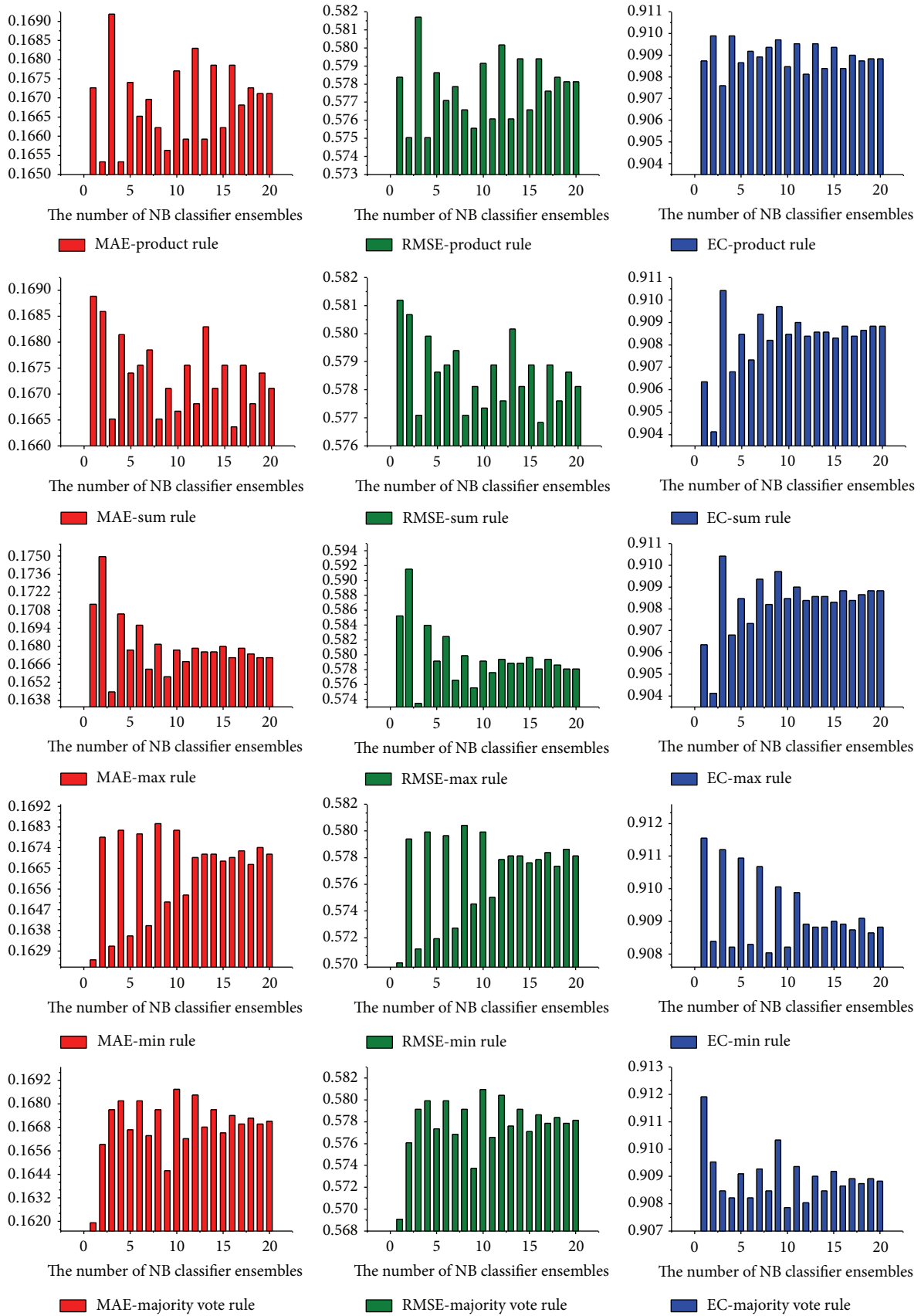


FIGURE 7: Bar chart comparison of five rules for Naïve Bayes classifier ensembles as applied to the AYE dataset. The red bar is MAE, the green bar is RMSE, and the blue bar is EC.

is observed with RMSE. Comparing Figure 4 and Figure 5, we find that the performances of all five rules are reduced in Figure 5. The reason for this is that the noisy data involved in the process of training a single classifier result in the final output of the NB ensemble being worse. However, the performances of the sum rule, the max rule, and the min rule are reduced, which indicates that the sum rule, the max rule, and the min rule have a better ability to tolerate the noisy data among the five rules.

In Tables 3 and 4, the dataset we used is extremely unbalanced. A very high CR, even exceeding 90%, does not indicate good detection performance. In Table 4, the average DR value of the Naïve Bayes algorithm is 71.12% and the average DR value of the NBTree algorithm is 72.75%. Both values are less than 75%, which is unsatisfactory for practical applications. The DR values of the product rule and the majority rule are even lower (64.57% and 62.46%, resp.). These results indicate that if the average accuracy of the individual NB classifiers is lower, the average accuracy of the ensemble classifiers, which are constructed using these individual classifiers, will become even lower than the average accuracy of the individual NB classifiers in certain combination rules.

Therefore, we should avoid drawing noisy data into the NB ensemble. The standard Naïve Bayes and NBtree are both individual classifiers, and they only need to train one time. In contrast to standard Naïve Bayes and NBtree, NB ensemble needs to train many individual NB classifiers to construct the NB ensemble. The training time of the NB ensemble is relatively long. From Figures 4(a)–4(f), we can see that in order to obtain relatively better performance, the NB ensemble needs approximately 15 individual NB to construct the NB ensemble; that is, the NB ensemble algorithm needs to train 15 times. Thus, compared with NB ensemble, the NBtree algorithm saves a large amount of time cost.

5. Conclusions

The Naïve Bayes classifier ensemble is a type of ensemble classifier based on Naïve Bayes for AID. In contrast to Naïve Bayes, the NB classifier ensemble algorithm trains many individual NB classifiers to construct the classifier ensemble and then uses this classifier ensemble to detect the traffic incidents, and it avoids the burden of choosing the optimal threshold and tuning the parameters. In our research, we take the traffic incident detection problem as a binary classification problem based on the ILD data and use the NB ensemble to divide the traffic patterns into two groups: an incident traffic pattern and nonincident traffic pattern. In this paper, we have performed two groups of experiments to evaluate the performances of the three algorithms: standard Naïve Bayes, NB ensemble, and NBTree. In the first group of experiments, we used all the three algorithms as applied to the I-880 dataset without noisy data. The results indicate that the performances of the five rules of the NB ensemble are significantly better than those of standard Naïve Bayes and slightly better than those of NBTree in terms of some indicators. More importantly, the NB ensemble performance is very stable. To

further test the stability of the three algorithms, we applied the three algorithms to the AYE dataset with noisy data in the second group of experiments. The experimental results indicate that the NB ensemble has the best ability to tolerate the noisy data among the three algorithms. After analyzing the experimental results, we found that if the average accuracy of the individual NB classifiers is lower, the average accuracy of the ensemble classifiers constructed by these individual classifiers will become even lower than the average accuracy of the individual NB classifiers. To obtain good results for the NB ensemble classifier, we should avoid drawing the noisy data into the ensemble. NBTree is an individual classifier that needs to train only one time, whereas the NB ensemble needs to train many individual NB classifiers to construct the NB ensemble. As a result, compared with the NBTree algorithm, the NBTree algorithm reduces the time cost. The contribution of this paper is that it presents the development of a freeway incident detection model based on the Naïve Bayes classifier ensemble algorithm. The NB ensemble not only improves the performances of traffic incidents detection but also enhances the stability of the performances with an increased number of classifiers. The advantage of NBTree is that the MTTD value is better than that of the NB ensemble algorithm. We believe that the NB ensemble algorithm and NBTree can be successfully utilized in traffic incidents detection and the other classification problems. In a future study, we will concentrate on constructing an NB ensemble and scaling up the accuracy of NBTree to detect traffic incidents.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is part of an ongoing research Supported by National High Technology Research and Development Program of China under Grant no. 2012AA112304 and Research and Innovation Project for College Graduates of Jiangsu Province no. CXZZ13_0119. Qingchao Liu would like to make a grateful acknowledgment to all the members of Traffic Engineering Lab in Southeast University, China, for their help and many useful suggestions in this study.

References

- [1] Y.-S. Jeong, M. Castro-Neto, M. K. Jeong, and L. D. Han, "A wavelet-based freeway incident detection algorithm with adapting threshold parameters," *Transportation Research C: Emerging Technologies*, vol. 19, no. 1, pp. 1–19, 2011.
- [2] I. Eleni and G. Matthew, "Fuzzy-entropy neural network freeway incident duration modeling with single and competing uncertainties," *Computer-Aided Civil and Infrastructure Engineering*, vol. 28, no. 6, pp. 420–433, 2013.
- [3] F. Yuan and R. L. Cheu, "Incident detection using support vector machines," *Transportation Research C: Emerging Technologies*, vol. 11, no. 3–4, pp. 309–328, 2003.

- [4] J. Xiao and Y. Liu, "Traffic incident detection using multiple kernel support vector machine," *Transportation Research Record*, vol. 2324, pp. 44–52, 2012.
- [5] X. Jin, D. Srinivasan, and R. L. Cheu, "Classification of freeway traffic patterns for incident detection using constructive probabilistic neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 5, pp. 1173–1187, 2001.
- [6] D. L. Han and A. D. May, *Automatic Detection of Traffic Operational Problems on Urban Arterials*, Institute of Transportation Studies, University of California, Berkeley, Calif, USA, 1989.
- [7] Y. J. Stephanedes and G. Vassilakis, "Intersection incident detection for IVHS," in *Proceedings of the 74th Annual Meeting of the Transportation Research Board*, Washington, DC, USA, 1994.
- [8] N. Hoose, M. A. Vicencio, and X. Zhang, "Incident detection in urban roads using computer image processing," *Traffic Engineering and Control*, vol. 33, no. 4, pp. 236–244, 1992.
- [9] J. Lu, S. Chen, W. Wang, and H. van Zuylen, "A hybrid model of partial least squares and neural network for traffic incident detection," *Expert Systems with Applications*, vol. 39, no. 5, pp. 4775–4784, 2012.
- [10] N.-E. E. Faouzi, H. Leung, and A. Kurian, "Data fusion in intelligent transportation systems: progress and challenges—a survey," *Information Fusion*, vol. 12, no. 1, pp. 4–10, 2011.
- [11] B. Twala, "Multiple classifier application to credit risk assessment," *Expert Systems with Applications*, vol. 37, no. 4, pp. 3326–3336, 2010.
- [12] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.
- [13] Y. Bi, J. Guan, and D. Bell, "The combination of multiple classifiers using an evidential reasoning approach," *Artificial Intelligence*, vol. 172, no. 15, pp. 1731–1751, 2008.
- [14] T. G. Dietterich, "Machine-learning research: four current directions," *The AI Magazine*, vol. 18, no. 4, pp. 97–136, 1997.
- [15] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.
- [16] R. Kohavi, "Scaling up the accuracy of Naive-Bayes classifiers: a decision tree hybrid," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pp. 202–207, Portland, Ore, USA, 1996.
- [17] G. H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," in *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (UAI '95)*, pp. 338–345, Montreal, Canada, 1995.
- [18] E. Parkany and C. Xie, "A complete review of incident detection algorithms and their deployment: what works and what doesn't," Tech. Rep. NETCR37, Transportation Center, University of Massachusetts.
- [19] S. Chen and W. Wang, "Decision tree learning for freeway automatic incident detection," *Expert Systems with Applications*, vol. 36, no. 2, pp. 4101–4105, 2009.
- [20] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [21] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.
- [22] D. J. Hand and R. J. Till, "A simple generalisation of the area under the ROC curve for multiple class classification problems," *Machine Learning*, vol. 45, no. 2, pp. 171–186, 2001.
- [23] R. L. Cheu, D. Srinivasan, and W. H. Loo, "Training neural networks to detect freeway incidents by using particle swarm optimization," *Transportation Research Record*, vol. 1867, pp. 11–18, 2004.
- [24] D. Srinivasan, X. Jin, and R. L. Cheu, "Adaptive neural network models for automatic incident detection on freeways," *Neurocomputing*, vol. 64, pp. 473–496, 2005.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

