Data Science and Engineering (2026) 11:248–259
https://doi.org/10.1007/s41019-025-00336-4

RESEARCH PAPERS

Real-Time Dynamic Response Identification for Highway Structural Health Monitoring Data

Zhixin Qi[1] · Xin Su[1] · Yulin Wang[1] · Zemin Chao[1] · Zejiao Dong[1] · Hongzhi Wang[1]

Received: 28 April 2025 / Revised: 23 October 2025 / Accepted: 7 December 2025 / Published online: 29 January 2026

Abstract
The high sampling frequency of highway structural health monitoring systems brings a heavy burden on data storage. However, existing dynamic response identification approaches can guarantee either reduced data volume after identification or high accuracy of dynamic response identification. Motivated by this, we propose a real-time dynamic response identification method to filter meaningless data. Our method not only selects effective features from highway structural health monitoring data, but also designs a training data generation strategy for machine learning models within the dynamic response identification framework. Experimental results on real highway structural health monitoring data demonstrate that our proposed approach spends 0.4 ms to process the monitoring data generated in 1 s and saves around 91.63% storage space. Also, the recall value of our method achieves 0.91 on average.

Keywords Monitoring data · Real-time identification · Feature selection · Machine learning · Model training

1 Introduction

With the constant improvement of highway construction, it is an important task to monitor the long-term performance of highway service for structural health evaluation and maintenance decision making [1]. As the cornerstone of revealing the essence of highway disease, structural health monitoring (SHM, for brevity) data are collected by many regions and institutions. As Fig. 1 shows, strain sensors are built in the pavement and generating SHM data continuously [2]. When there is no vehicle load, the SHM data change mainly with the pavement temperatures [3].

To obtain the transient information of highway structure under high-speed vehicle loads, the sampling frequency of highway SHM data reaches up to 2000 Hz. Such high frequency makes the data size reach 5–6 TB each month, which brings a heavy burden on data storage.

To alleviate the storage burden of highway SHM data, researchers and practitioners in pavement monitoring community attempt to extract the key data that are generated under vehicle loads, referred to *dynamic responses*, from the entire SHM data and filter the other data. Even though many effective dynamic response identification models have been proposed [4–8], they can not guarantee the small data storage size after filtering, high identification rate of dynamic responses, and real-time performance at the same time. How can we filter meaningless data as much as possible and achieve high identification rate in real time? This problem brings the following challenges.

(i) In practice, there is no label in highway SHM data and the ratio of dynamic responses is usually $\frac{1}{200}$. The unlabeled and sparse properties of SHM data make it difficult to establish a well-trained dynamic response identification model, which is called a *cold start* of model training. Therefore, the first challenge is how to

✉ Zhixin Qi
qizhx@hit.edu.cn

Xin Su
hit_sux@outlook.com

Yulin Wang
crystalniall@outlook.com

Zemin Chao
chaozm@hit.edu.cn

Zejiao Dong
hitdzj@hit.edu.cn

Hongzhi Wang
wangzh@hit.edu.cn

[1] Harbin Institute of Technology, Huanghe Road 73, Harbin, Heilongjiang, China

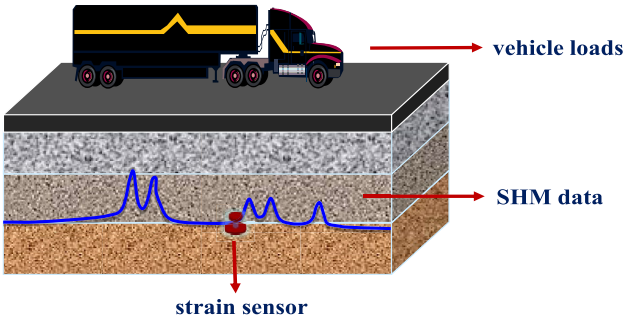


Fig. 1 The sketch of highway structural health monitoring

handle the cold start in dynamic response identification model establishment.

(ii) Since there are a lot of statistical characteristics in highway SHM data, such as the mean value, extreme value, and variance, the second challenge is how to determine which features are valuable for dynamic response identification.

(iii) Since the highway SHM data are generated with high frequency in real time, the third challenge is how to reduce the storage space of SHM data.

In light of the first challenge, we divide the historical highway SHM data into equal-length data segments and label *positive examples* for segments by crowdsourced annotation from field experts [9]. Note that in this paper, a *positive example* refers to a data segment which contains dynamic responses while a *negative example* denotes a data segment without dynamic responses. Based on the labeled segments, we construct a training data set where the ratio of positive examples to negative ones is 1:1 and input it into the machine learning model.

To address the second challenge, we select candidate statistical features which are related to the classification of positive and negative examples. Then, we use the feature selection based on mutual information to choose effective features from the candidate set [10]. In this way, we obtain the features which are most related to the classification labels, that is, most useful for dynamic response identification.

After identifying dynamic responses from SHM data, the data segments which are classified as positive examples are stored. To save the storage space, we design a real-time identification framework to classify the incoming data segments and filter the negative examples efficiently. Via this solution, the third challenge is solved.

To summarize, the contributions of this paper are threefold as follows.

(i) With the consideration of the unlabeled and sparse properties of highway SHM data, we develop a real-time dynamic response identification method based on crowdsourced labelling and machine learning.

(ii) We adopt feature selection strategies based on mutual information to determine the features which are useful for dynamic response identification.

(iii) Experimental results demonstrate that our proposed dynamic response identification method reduces around 91.63% data storage space and spends 0.4 ms to process the SHM data generated in 1 s. Also, the recall value of our approach achieves 0.91 on average.

The remainder of the paper is organized as follows. We first review the related work and propose the framework of the dynamic response identification method. After that, we discuss training data generation and dynamic response identification in detail. Then, experimental evaluations are presented and analyzed. Finally, we conclude this paper.

2 Related Work

The related literature is divided into the following main lines.

Highway SHM Data Denoising Before using measured SHM data to identify highway dynamic response, most studies have focused on noise reduction since there are many interference factors in the monitoring process [11].

In the Long-Term Pavement Performance (LTPP) program, researchers conducted statistical analyses on missing values, outliers, and redundant values in SHM data for noise reduction [12]. Bismor used the Butterworth low-pass filter to denoise the acceleration data collected by MEMS sensors and adopted an average filtering approach to process the magnetic field data [13]. Ma et al. divided SHM data into three parts, including the pavement dynamic response under loads, the baseline drifting with temperature, and the unavoidable noise [14]. They leveraged the moving average filtering method to reduce the noise created by temperature. However, when SHM data are denoised by filtering methods, the peak value was reduced. To weaken the influence on peak values, Sencer et al. recommended using linear interpolation before processing data with a moving average filtering approach [15]. Compared with non-stationary signal noise reduction methods, such as median filtering, stationary wavelet filtering, and empirical mode decomposition, Bian et al. observed that the Butterworth low-pass filter integrated with second-order difference outperformed the other approaches in terms of noise reduction level and algorithm cost [8].

Various kinds of filtering techniques have been widely applied in SHM data denoising. Even though the commonly-used noise reduction methods, such as median filtering [16],

band-pass filtering [17], synchronous accumulation [18], correlation detection [19], and lock detection [20], perform well in non-linear vibration signal denoising tasks, they need to consume a great amount of computing resources in the running process [21]. Therefore, these approaches are not suitable to denoise high-frequency and large-scale highway SHM data. Considering the properties of highway SHM data, we adopt the noise reduction method which combines linear interpolation and moving average filtering in this paper.

Highway Dynamic Response Identification Most approaches of highway dynamic response identification are based on threshold [4–6], information entropy [7], or energy [7, 8].

In earlier years, researchers set the threshold value manually based on the characteristics of a SHM data segment and compared all data points with the determined threshold value [4]. If there exists a point which is larger than the threshold value, there may be dynamic response in the data segment. However, in practice, the selection of threshold value is related to the traffic loads, environmental conditions, and researcher experience. The day-night changes may also cause the fluctuation of SHM data collected by sensors. Therefore, the threshold value method is hard to guarantee the performance of dynamic response identification for high-frequency and large-scale highway SHM data. Motivated by this, Abed et al. considered that the highway SHM data without vehicle loads were Gaussian distributed [5]. They selected the data range of Gaussian distribution as the threshold range and identified the dynamic responses based on it. Han et al. used the low-pass filter to obtain extreme data points and extracted the dynamic response data based on them [6].

To improve the effectiveness of dynamic response identification, Hu et al. introduced the concept of information entropy and the endpoint detection method based on short-time energy into this field [7]. They computed the information entropy and signal energy for normalized series data with equal step length and identified the dynamic responses. Based on the experimental evaluation, they observed that the information-entropy-based identification method was suitable for the SHM data whose signal-to-noise ratio was high. In addition, there was a big difference between SHM data and other data in terms of energy. Bian et al. computed the short-time average energy, long-time average energy, and the ratio of them [8]. They set the ratio value as athreshold and treated the short-term data segments whose ratio was larger than the threshold as dynamic responses.

While the above-mentioned work proposed a lot of effective methods to identify dynamic responses for highway SHM data, they could not guarantee reduced data storage requirements after filtering, high dynamic response identification accuracy , and real-time performance at the same time. To make up for the deficiencies, we take feature selection and machine learning into account and develop a real-time dynamic response identification method for highway SHM data.

3 Framework Overview

In this section, we introduce the framework of our proposed dynamic response identification method.

The design of this framework has two goals. One is to obtain sufficient and effective data for model training. To achieve this goal, we add the training data generation module in our framework. The other is to establish a real-time machine-learning-based model that is able to filter meaningless information of incoming SHM data as much as possible. Therefore, we add the module of dynamic response identification to our framework. The framework is illustrated in Fig. 2.

3.1 SHM Data Pre-processing

Our goal in this module is to clean the highway SHM data and prepare data segments for training data generation or dynamic response identification.

For the historical SHM data D, the first step is to adopt the noise reduction method which combines linear interpolation and moving average filtering to denoise D. Then, we divide D into equal-length segments D_i $(1 \leq i \leq \frac{|D|}{100})$ which contain 100 data points, that is, $|D_i| = 100$. Finally, we send the data segments to the next module for training data generation.

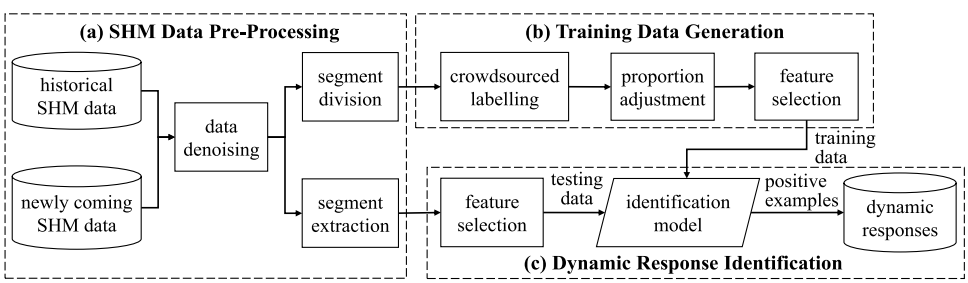


Fig. 2 Framework of dynamic response identification method for highway SHM data: **a** SHM data pre-processing, **b** training data generation, **c** dynamic response identification

For the incoming SHM data D_N, we first use the data denoising approach to reduce the noise of D_N. Then, we extract data segment D_s whose length is 100 from D_N. Finally, we send each D_s to the dynamic response identification module.

Note that the length of a data segment is typically influenced by the data sampling frequency, vehicle axle spacing, and traveling speed. A shorter data segment may fail to adequately capture the dynamic response characteristics, while a longer segment could reduce the efficiency of data processing. For instance, the time interval for a vehicle with a 2.5-m axle spacing to pass one sensor at 100 km/h can be calculated as around 0.090 s. Thus, at a sampling frequency of 1000 Hz, only 90 data points are effective to record the complete passing information of the vehicle. Accordingly, we set the length of a data segment to 100 points. In addition, via our proposed framework, even if a complete dynamic response is split across two adjacent data segments, both segments can be accurately identified.

3.2 Training Data Generation

In order to generate training data for the dynamic response identification model with the given SHM data segments, we apply crowdsourced labelling and feature selection in this module.

First, we label the historical segments by crowdsourcing and differentiate positive examples and negative examples. Since the original proportion of positive and negative examples is around 1:200, we select the labelled segments by adjusting the proportion to 1:1 for a high-performance training data set. Finally, we extract effective features from the selected segments to construct training data for the dynamic response identification model.

We will discuss this module in detail in the section of *Training Data Generation*.

3.3 Dynamic Response Identification

For the generated training data, we train the dynamic response identification model in this module.

For the given newly coming data segments, the first step is to select effective features from them. With the selected features as testing data in the identification model, we identify positive examples as dynamic responses.

We will explain this module in detail in the section of *Dynamic Response Identification*.

4 Training Data Generation

In this section, we discuss the detailed process of training data generation.

As Fig. 2 shows, there are three steps to generate the training data for the dynamic response identification model. The first step is crowdsourced labelling. Since each historical SHM data segment $D_i \subset D$ ($1 \leq i \leq \frac{|D|}{100}$) is unlabeled, we label each D_i with L_i by crowdsourcing [9]. For high-quality labelled data segments, we select 9 domain experts to judge whether each D_i contains dynamic responses. If there is dynamic response in D_i, the experts set the corresponding L_i as a positive example. If not, they set L_i as a negative example. Based on the results of 9 experts, we determine the final label of each L_i.

The second step is proportion adjusting. Since the original proportion of positive examples and negative examples is about 1:200, it is hard to train a high-performance model with the original proportion. Therefore, based on the labeled data segments, we construct a SHM data set whose proportion of positive examples and negative examples is 1:1 and treat it as the training data set for the dynamic response identification model.

The third step is feature selection. Since there are many candidate statistical features in the training data set, such as mean value, variance, and peak value, it is necessary for us to determine which features are valuable to identify dynamic responses, that is, are most related to the given labels. Our core idea is to adopt the concept of mutual information [10] to measure the correlation between each feature and its corresponding label. The value of mutual information *MI* of a feature *f* is computed as follows.

$$MI(f) = \alpha \times I(L; f|S) - g(L, S, f),$$

where α is a coefficient to regulate the relative significance of mutual information, L is the label set, and S is the selected feature set. The function $g(L, S, f)$ is a deviated function about f and S under L, which denotes the redundancy of f and S, and can be learned from historical data. The function I is computed as follows.

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} log \frac{p(x,y)}{p(x)p(y)},$$

where X and Y are two variables. The function $p(x, y)$ is the joint probability density function of X and Y. If X and Y are closely related to each other, the value of I will be large, and $I = 0$ means that these two variables are independent.

In this step, our candidate features are mean value, variance, skewness, peakness, maximum value, and minimum

value. Based on the computation of mutual information, we select the features whose mutual information values are large as effective features for the dynamic response identification model. Then, we encode the effective features and the corresponding label of each selected data segment as training data and send them to the module of dynamic response identification.

Since the selected effective features depend on the data characteristics [10], different SHM data sets may produce different effective features. We will show the practical feature selection process in our experimental section.

5 Dynamic Response Identification

In this section, we explain the training and testing process in the dynamic response identification module.

We denote the selected effective features as F_s and the encoded feature values of each training data segment D_i as F_i. As illustrated in Fig. 2, for the training process, the goal is to train a dynamic identification model M, where $L_i = M(F_i)$.

Note that the selection of the machine learning model relies on the training data characteristics [22, 23], we will discuss the practical model selection in our experimental section.

For each data segment D_s extracted from incoming SHM data D_N, we first select the same features F_s as those in the training data generation module from D_s. Then, the values of F_s are encoded as a vector V_s. We treat V_s as testing data and send it to the trained model M.

By computing $L_s = M(V_s)$, M produces the label L_s of V_s and judges whether the corresponding D_s is a positive example. If D_s is positive, it contains dynamic responses.

The pseudo code of identifying dynamic responses are shown in Algorithm 1. The input is the incoming SHM data D_N and the output is the dynamic response data set D_R.

Input: newly coming SHM data D_N
Output: dynamic response data set D_R
1 $D_R \leftarrow \emptyset$;
2 **foreach** $D_s \in D_N$ **do**
3 $F_s \leftarrow$ getFeatures(D_s) ;
4 $V_s \leftarrow$ encode(F_s) ;
5 $L_s \leftarrow M(V_s)$;
6 **if** L_s==*1* **then**
7 $D_R \leftarrow$ append(D_R, D_s) ;
8 **if** L_s==*0* **then**
9 continue;
10 **return** D_R;

Algorithm 1 Dynamic Response Identification Algorithm

We first initialize an empty set D_R to store the dynamic responses (Line 1). For each data segment D_s in D_N, we extract features F_s from it and encode F_s to a vector V_s (Lines 2–4). Then, we input V_s into the trained identification model M and obtain the corresponding label L_s (Line 5). If the value of L_s is 1, D_s is a positive example and we append it to the set D_R (Lines 6–7). If the value of L_s is 0, D_s is a negative example and we filter it. The process is iterated until there are no remaining data segements (Lines 8–9). Finally, we output the dynamic response set D_R (Line 10).

Time Complexity Analysis The time complexity of our algorithm is linear with respect to the number of data segments, denoted as n, in the incoming SHM data. Formally, the complexity can be expressed as $O(n)$, as the algorithm processes each segment exactly once in order. The identification operation for each segment executes in constant time, that is $O(1)$. Therefore, the overall time complexity remains $O(n)$.

Space Complexity Analysis The space complexity is determined by the storage of identified positive segments. Let p represent the proportion of segments containing dynamic responses, where $p \ll 1$. Suppose the total number of segments is n, the space complexity is $O(p \cdot n)$.

6 Experimental Evaluation

In this section, we discuss the experiments of our proposed framework and analyze the experimental results. First, we conduct feature selection to determine the effective features for the identification model. Then, we compare our approach with existing dynamic response identification methods and test the performance of different machine learning models in our framework. Next, we verify the efficiency of our identification method. Finally, we tune the parameters in machine learning models to select the best parameter values.

Data Sets In our experiments, we use real SHM data collected by 8 single-point fiber grating sensors from Jilin Tonghua Highway. The sampling frequency of sensors is 250 Hz and the time range of highway SHM data is 7 days. The training and validation data sets of our dynamic response identification model are shown in Table 1.

Metrics Since our first goal is to filter the meaningless SHM data as much as possible to save storage space, we use the sum of TP (True Positive) and FP (False Positive) to measure the effectiveness of the proposed dynamic response identification method. TP is the number of positive examples that are identified correctly and FP is the number of negative examples that are identified as positive examples. The sum of TP and FP represents the data segments which

Table 1 Training and validation data sets information

Data set type	Sensor ID	#-Positive examples	#-Negative examples
Training set	1	300	300
Validation set	1	117	23,283
Training set	2	200	200
Validation set	2	76	15,124
Training set	3	200	200
Validation set	3	86	17,114
Training set	4	200	200
Validation set	4	116	23,084
Training set	5	200	200
Validation set	5	106	21,094
Training set	6	200	200
Validation set	6	98	19,502
Training set	7	200	200
Validation set	7	68	13,532
Training set	8	200	200
Validation set	8	82	16,318

Table 2 Mutual information values of candidate features

Feature	Mutual information
Mean value	0.1081
Variance	0.5065
Skewness	0.4646
Peakness	0.3069
Maximum value	0.2135
Minimum value	0.0709
Range	**0.5222**
Peak to average ratio (PAR)	**0.5279**

are not filtered by our model, that is, those that are preserved in our storage space.

Also, since the second goal of our framework is to preserve positive examples as much as possible, we use recall to measure the ratio of dynamic responses which are identified and preserved. The value of recall is computed as follows.

$$recall = \frac{TP}{TP + FN},$$

where FN is the number of positive examples which are filtered wrongly.

Since the other goal of our framework is to identify dynamic responses in real time, we use running time to measure the efficiency of our proposed method. We test each result 5 times and report the average time costs.

Setup All experiments are conducted on a computer with an AMD Ryzen 7 5800HS CPU@3.20 GHz, 16 GB memory. All algorithms are implemented in Python.

6.1 Feature Selection

Since feature selection is an essential step in both training data generation and dynamic response identification modules, we test the effectiveness of feature selection from highway SHM data. The candidate features are mean value, variance, skewness, peakness, maximum value, minimum value, range, and peak to average ratio. We utilize a SHM data set, which contains 500 data segments from Sensor 1, to compute the mutual information value between each candidate feature and the data segment labels. All data segments are labeled as 0 or 1 by crowdsourcing. The mutual information values are depicted in Table 2.

As Table 2 shows, the mutual information values of range and peak to average ratio (PAR) are the highest. Therefore, we treat range and PAR as effective features in our SHM data sets. Since different SHM data sets may produce different effective features, we need to recompute the mutual information values of the candidate features when the given data change.

6.2 Comparison Experiments

To verify the effectiveness of our proposed framework, we test the sum of TP and FP and the recall value compared with 3 commonly-used dynamic response identification methods, that are threshold method [6], entropy method [7], energy method [8], and 6 classical machine learning models in our framework, that are decision tree, k-nearest neighbors, naive bayes, support vector machine, logistic regression, and random forest [22, 23]. All validation data segments from the 8 sensors listed in Table 1 are grouped into datasets, with each dataset consisting of 400 continuous segments. The average experimental results obtained on these datasets are presented in Tables 3 and 4.

As shown in Tables 3 and 4, the three commonly-used methods identify fewer dynamic responses, and their performance exhibits considerable variation across different sensors. In contrast, the machine learning methods in our framework achieve recalls higher than 0.8 for dynamic response identification in most cases, while also preserving a reasonable number of data segments. We conduct a comprehensive statistical analysis of the performance of each method, and the results are presented in Table 5.

As depicted in Table 5, the method achieving the highest average recall is the support vector machine model, with 29.3391% preserved data; the method with the second highest average recall is the k-nearest neighbors model, with 8.3718% preserved data. As the goal of our framework is not only to achieve the dynamic response identification, but also to filter meaningless data segments, we recommend

Table 3 TP + FP values of different methods on different sensors

Method	Sensor 1	Sensor 2	Sensor 3	Sensor 4
Threshold method	7.3590	6.2632	2.8605	4.5345
Entropy method	2.0513	0.9474	0.2791	2.2586
Energy method	2.4615	1.2895	0.3023	5.8103
Decision tree	44.2307	58.4736	12.3488	14.9827
k-nearest neighbors	52.0769	65.4211	19.4419	23.0344
Naive Bayes	24.5641	35.2895	8.6512	11.1034
Support vector machine	166.8205	149.3421	114.2558	113.0345
Logistic regression	44.2308	58.4737	12.3488	14.9828
Random forest	41.8462	54.1316	14.8140	18.1379
Method	Sensor 5	Sensor 6	Sensor 7	Sensor 8
Threshold method	5.2830	5.2653	3.7353	2.1463
Entropy method	148.1698	0.4490	28.9118	0.1463
Energy method	222.8870	0.6327	25.6765	0.0488
Decision tree	18.2264	38.4286	32.5000	15.9756
k-nearest neighbors	26.5471	45.3265	37.6765	23.0000
Naive Bayes	13.0377	22.2653	17.6471	8.3415
Support vector machine	117.6038	119.3469	115.6765	104.5366
Logistic regression	18.2264	38.4286	32.5000	15.9756
Random forest	20.7925	36.6939	30.7059	16.4390

Table 4 Recall of different methods on different sensors

Method	Sensor 1	Sensor 2	Sensor 3	Sensor 4
Threshold method	0.8632	0.9605	0.3256	0.8966
Entropy method	0.6068	0.4605	0.1279	0.8448
Energy method	0.7949	0.6316	0.1279	0.8966
Decision tree	0.9145	1.0000	0.8023	0.9397
k-nearest neighbors	0.9402	1.0000	0.8023	0.9655
Naive Bayes	0.9145	1.0000	0.6512	0.9397
Support vector machine	0.9915	1.0000	0.9070	0.9914
Logistic regression	0.9145	1.0000	0.8023	0.9397
Random forest	0.9145	1.0000	0.7907	0.9483
Method	Sensor 5	Sensor 6	Sensor 7	Sensor 8
Threshold method	0.9057	0.7857	0.3676	0.3537
Entropy method	0.9811	0.2449	0.7353	0.0488
Energy method	0.9811	0.2245	0.5294	0.0244
Decision tree	0.9811	0.9082	0.8676	0.8537
k-nearest neighbors	0.9623	0.9184	0.8676	0.8537
Naive Bayes	0.9811	0.8980	0.6912	0.7561
Support vector machine	0.9906	0.9286	0.9265	0.8537
Logistic regression	0.9811	0.9082	0.8676	0.8537
Random forest	0.9811	0.9082	0.8529	0.8415

users adopt the k-nearest neighbors model when using our framework.

Comparison results further confirm that our proposed method not only reduces the storage space dramatically, but also guarantees s a high dynamic response identification accuracy.

6.3 Efficiency Study

To verify the efficiency of our approach, we test the total time costs and the time costs per data segment. The results of all data from 8 sensors are depicted in Table 6.

As Table 6 shows, the total time costs are directly proportional to the number of data segments that are processed. We observe that the three commonly-used methods process each data segment in 0.02 ms and the six machine learning methods in our framework process each data segment on average 0.037 ms.

Table 5 Comprehensive performance of different methods

Method	Average recall	Total TP + FP	Ratio of preserved data (%)
Threshold method	0.6823	1664	1.1108
Entropy method	0.5037	9123	6.0901
Energy method	0.5263	13,214	8.8211
Decision tree	0.9084	9956	6.6462
k-nearest neighbors	0.9137	12,541	8.3718
Naive Bayes	0.8540	6039	4.0314
Support vector machine	0.9486	43,950	29.3391
Logistic regression	0.9084	9956	6.6462
Random forest	0.9047	9996	6.6729

Table 6 Time costs of different models

Method	Total time costs (ms)	Time costs per data segment (ms)
Threshold method	2835.3593	0.0189
Entropy method	2280.3950	0.0152
Energy method	1470.3634	0.0098
Decision tree	978.3792	0.0065
k-nearest neighbors	5347.4977	0.0357
Naive Bayes	924.1548	0.0062
Support vector machine	23816.1831	0.1590
Logistic regression	957.5059	0.0064
Random forest	1308.3177	0.0087

Although the commonly-used methods process data in a shorter time, as shown in the experimental results presented in Section 6.2, their model performance is unsatisfactory. For our framework, with the exception of the support vector machine and the k-nearest neighbors methods, all other methods are able to complete the identification of a single data segment within 0.01 ms. Although the k-nearest neighbors method takes 0.0357 ms to process a single data segment, it is still able to perform real-time identification of dynamic responses for SHM data at a sampling frequency of 1000 Hz. Therefore, our proposed approach is able to identify dynamic responses in real time.

Evaluation results further demonstrate the high efficiency of our proposed method.

Table 7 Default values in different methods

Method	Parameter	Default value
Decision tree	random_state	136
Decision tree	max_features	2
k-nearest neighbors	weights	uniform
Naive Bayes	model	GaussianNB
Support vector machine	random_state	475
Logistic regression	random_state	67
Logistic regression	penalty	l1
Logistic regression	solver	liblinear
Random forest	random_state	412
Random forest	max_features	2

6.4 Parameter Tuning

There are some parameters in machine learning models of our framework. To determine the optimal values for these parameters, we randomly choose 2000 data segments from all sensors and tune the model parameters according to their performance. By varying each parameter, we measure the average recall and TP + FP value among the 8 sensors. When we vary one parameter, the other parameters are set to their default values. The default values in different models are shown in Table 7 and the experimental results of parameter tuning are illustrated in Figs. 3, 4, 5, 6, 7, and 8.

As depicted in Fig. 3, for the decision tree model, we vary max_depth and $min\_samples\_leaf$ from 1 to 15. It can be observed that as max_depth increases, the total value of TP + FP also rises, and the average recall value shows a decreasing trend. This is because increasing max_depth enables the tree to capture more complex patterns, thereby better distinguishing positive and negative examples. When the value of max_depth is 3, the model gets the highest recall, and the value of TP + FP is relatively small. However, as $min\_samples\_leaf$ increases, each leaf node requires more samples, meaning the tree becomes more conservative. Samples are only predicted as positive when the evidence is sufficiently strong, leading to the model's performance gradually stabilizing. When the value of $min\_samples\_leaf$ is 3, the average recall value reaches its maximum and the average value of TP +

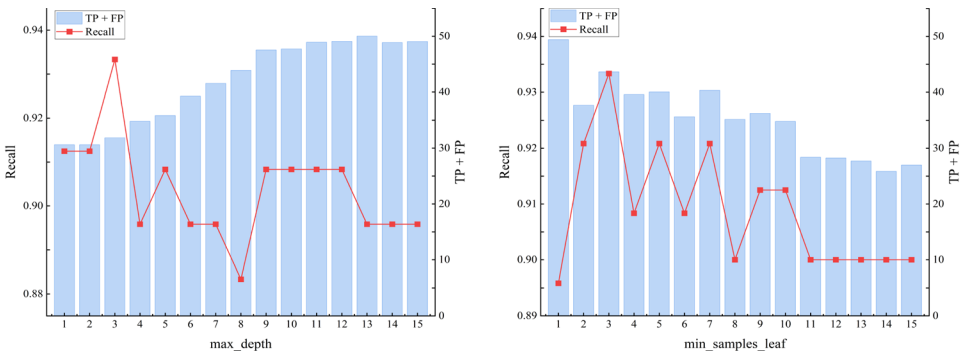


Fig. 3 Results of decision tree model in our framework

(a) Tuning of Parameter max_depth. (b) Tuning of Parameter min_samples_leaf.

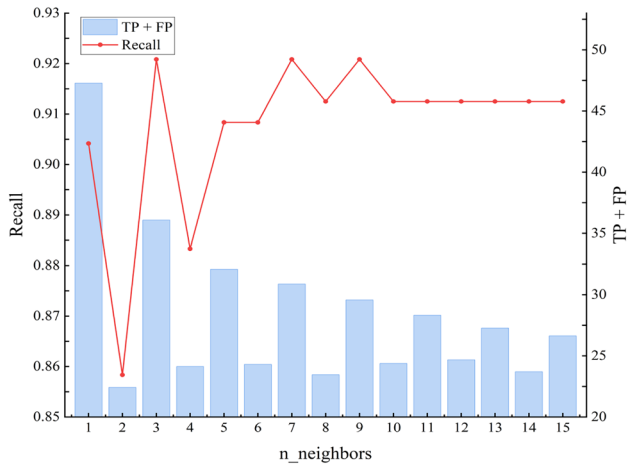


Fig. 4 Results of k-nearest neighbors model in our framework

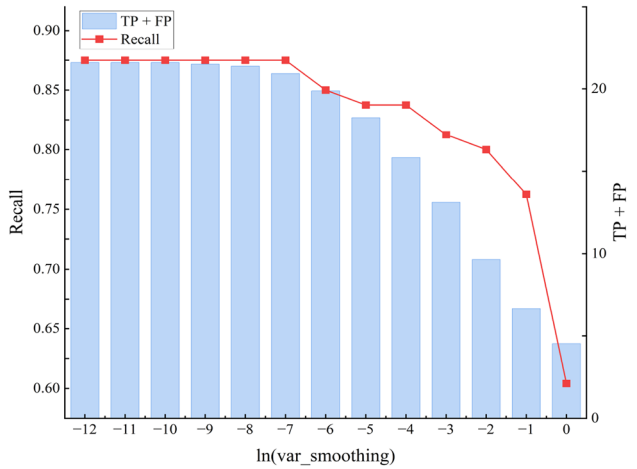
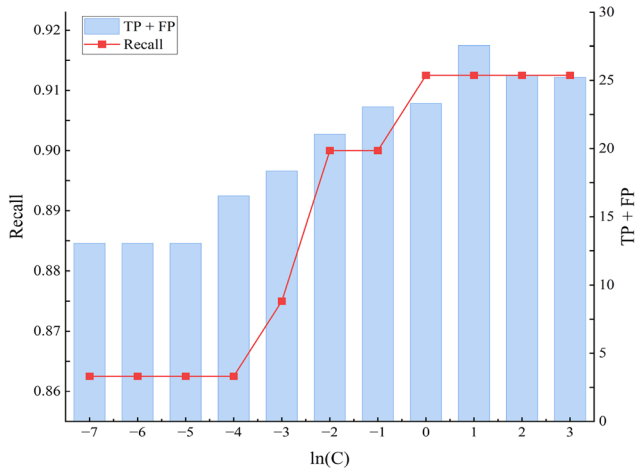
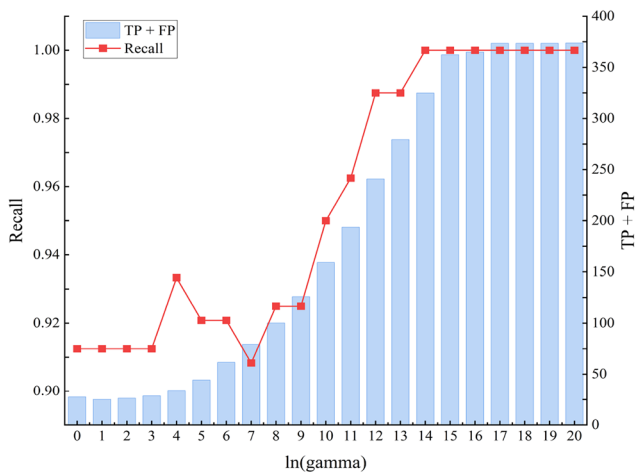


Fig. 5 Results of Naive Bayes model in our framework



(a) Tuning of Parameter C.



(b) Tuning of Parameter gamma.

Fig. 6 Results of support vector machine model in our framework

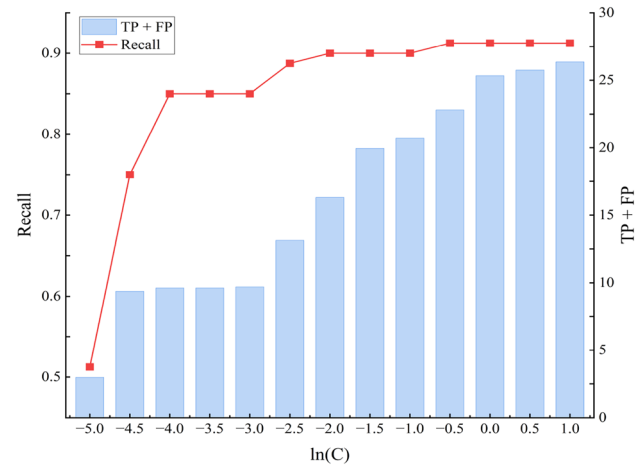
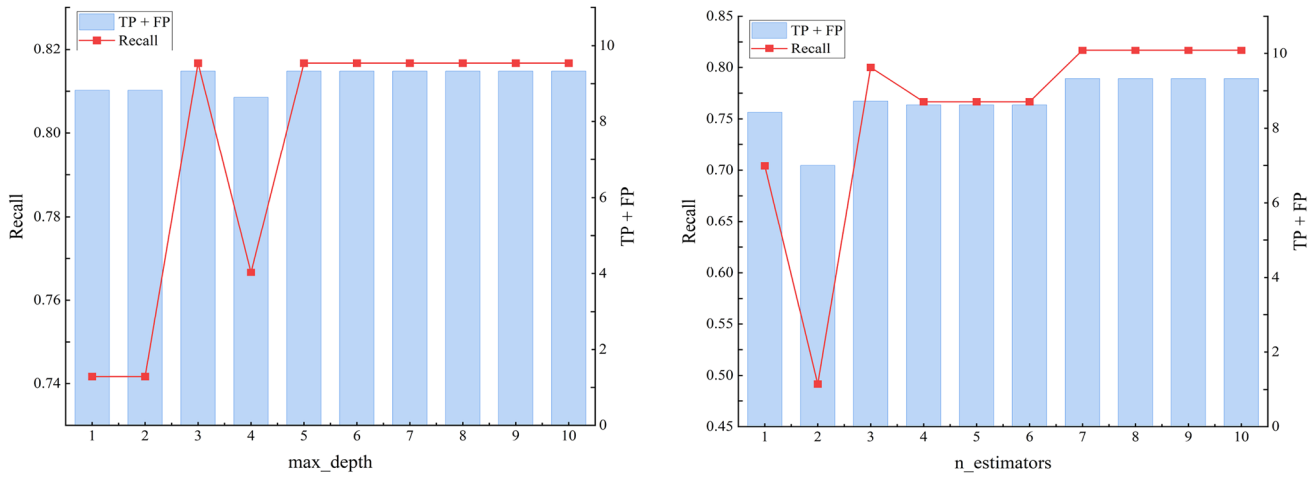


Fig. 7 Results of logistic regression model in our framework

FP is not excessively large. Therefore, to achieve good performance of the decision tree model, we set 3 as the value of max_depth and $min\_samples\_leaf$.

As shown in Fig. 4, for the k-nearest neighbors model, we vary $n\_neighbors$ from 1 to 15. It can be seen that as $n\_neighbors$ increases, the classification boundary gradually smooths out, the impact of noise diminishes, and consequently the recall gradually stabilizes. Currently, in this binary classification problem, when $n\_neighbors$ is even, the occurrence of tied votes between positive and negative examples forces many boundary samples to be classified as negative. This phenomenon does not arise when $n\_neighbors$ is odd. Therefore, as the value of $n\_neighbors$ increases, while the overall value of TP + FP exhibits a downward trend, a sawtooth-like odd-even effect emerges. When the value of $n\_neighbors$ is 9, the recall reaches the highest level and the value of TP + FP is relatively small. Thus, to achieve well performance of



(a) Tuning of Parameter max_depth. (b) Tuning of Parameter n_estimators.

Fig. 8 Results of random forest model in our framework

the k-nearest neighbors model, we set 9 as the value of $n\_neighbors$.

As illustrated in Fig. 5, for the naive bayes model, we vary the natural logarithm of $var\_smoothing$ from -12 to 0. We see that as $var\_smoothing$ increases, both recall and TP + FP exhibit a decreasing trend. This is because the number of negative examples in the dataset far exceeds that of positive examples. Increasing $var\_smoothing$ enhances distribution overlap, making decisions more dependent on prior probabilities and leading the model to predict the negative class more frequently. When the natural logarithm of $var\_smoothing$ reaches -7, the recall remains at the maximum level, while TP + FP reaches its minimum within this range. Therefore, to achieve well performance of the naive bayes model, we set -7 as the value of $ln(var\_smoothing)$, the corresponding value of $var\_smoothing$ is 9.119×10^{-4}.

As depicted in Fig. 6, for the support vector machine model, we vary the natural logarithm of C from -7 to 3 and the natural logarithm of *gamma* from 0 to 20. It can be observed that when the value of *ln*(C) is -2, the average recall reaches the highest and the value of TP + FP is the lowest. This is because when the penalty coefficient C is small, the model tends to select decision boundaries with "wide margins". Many samples near the boundary (including some positive examples) are classified as negative. As C increases, the decision boundaries better fit the training data, and more samples are correctly classified as positive. In addition, as the natural logarithm of *gamma* increases, the model transitions from near-linearity to capturing a degree of non-linearity, thereby enhancing the accurate identification of positive examples. Consequently, both the value of recall and TP + FP gradually increase and eventually stabilize. When *ln*(*gamma*) reaches 14, the recall remains at the maximum level, while TP + FP reaches its minimum value. Thus, to achieve the best performance of the support vector machine model, we set 0 as the value of *ln*(C) and 14 as the value of *ln*(*gamma*), that is, C is set to 1 and *gamma* is set to 1.2026×10^6.

As shown in Fig. 7, for the logistic regression model, we vary the natural logarithm of C from -5.0 to 1.0, which means that the value of C varies from about 0.0067 to 2.7182. It can be seen that when C is small, few samples are predicted as positive examples, as the output probability approaches the prior probability while the dataset contains a large number of negative examples. As C increases, the model begins utilizing features for classification, enabling more samples to be predicted as positive examples. This leads to an increase in TP + FP and recall, eventually reaching saturation. When the value of *ln*(C) is -0.5, the recall reaches the highest level and the value of TP + FP is the lowest. Therefore, to achieve satisfactory performance of the logistic regression model, we set -0.5 as the value of *ln*(C), the corresponding value of C is 0.6065.

As illustrated in Fig. 8, for the random forest model, we respectively vary max_depth and $n\_estimators$ from 1 to 10. Similar to decision tree models, increasing max_depth and $n\_estimators$ enhance model performance from two distinct perspectives: by boosting the complexity of individual learners and by increasing the overall capacity and stability of the ensemble model. We see that both the value of recall and TP + FP gradually stabilize with some fluctuation. When max_depth is set to 5, the recall reaches its maximum value. Beyond this point, further increases in max_depth do not lead to any improvement in model performance. Similarly, the model reaches the optimal performance when $n\_estimators$ reaches 7. Thus, to achieve the best performance of the random forest model,

we set 5 as the value of max_depth and 7 as the value of $n\_estimators$.

7 Conclusion

In this paper, we propose a real-time dynamic response identification method for highway structural health monitoring data. Based on the method, we are able to reduce the storage space dramatically and guarantee a high identification rate of dynamic responses. Evaluation results show that our proposed method with the k-nearest neighbors model not only decreases the highway structural health monitoring data storage space to 8.37%, but also achieves high efficiency and high recall at the same time.

Since the characteristics of highway structural health monitoring data are continuously changing, our future work aims to design a self-adaptive dynamic response identification model. In addition, since the performance of machine-learning-based identification models can be unstable, we are also interested in developing an integrated identification framework with different base models.

Acknowledgements This paper is supported by the National Key Research and Development Program of China (2024YFE0214600) and the National Natural Science Foundation of China (52308449).

Author Contributions Zhixin Qi: writing the original draft, methodology; Xin Su: experiments; Yulin Wang: data curation; Zemin Chao: methodology supervision; Zejiao Dong: application supervision; Hongzhi Wang: project administrator.

Data Availability The data that support the findings of this study are partially available from the corresponding author upon reasonable requests.

Declarations

Conflict of interest Authors declare that they have no conflict of interest.

References

1. Di Graziano A, Marchetta V, Cafiso S (2020) Structural health monitoring of asphalt pavements using smart sensor networks: a comprehensive review. J Traffic Transp Eng (Engl Ed) 7(5):639–651
2. Liu Z, Gu X, Wu C, Ren H, Zhou Z, Tang S (2022) Studies on the validity of strain sensors for pavement monitoring: a case study for a fiber Bragg grating sensor and resistive sensor. Constr Build Mater 321:126085
3. Wang H-P, Xiang P, Jiang L-Z (2020) Optical fiber sensing technology for full-scale condition monitoring of pavement layers. Road Mater Pav Des 21(5):1258–1273
4. Tsai Y-C, Kaul V, Mersereau RM (2010) Critical assessment of pavement distress segmentation methods. J Transp Eng 136(1):11–19
5. Abed A, Thom N, Neves L (2019) Probabilistic prediction of asphalt pavement performance. Road Mater Pav Des 20(sup1):247–264
6. Han H, Han W, Ma S, Hu G (2021) Enhanced processing of low signal-to-noise-ratio dynamic signals from pavement testing. Measurement 182:109697
7. Hu W, Zhao Q, Liu Y, Li Z, Kong X (2020) Damage evaluation of the paving around manholes under vehicle dynamic load. Adv Mater Sci Eng 2020:1–11
8. Bian Z, Zhao H, Peng K, Zeng M, Guo M (2023) Vehicle axle load sensing based on distributed vibration fiber technology. J Tongji Univ 51(8):1157–1167
9. Hossain M, Kauranen I (2015) Crowdsourcing: a comprehensive literature review. Adv Mater Sci Eng 8(1):2–22
10. Qi Z, Wang H, He T, Li J, Gao H (2020) Friend: feature selection on inconsistent data. Neurocomputing 391:52–64
11. Yang K, Kim S, Harley JB (2023) Guidelines for effective unsupervised guided wave compression and denoising in long-term guided wave structural health monitoring. Struct Health Monit 22(4):2516–2530
12. Jia Y, Wang J, Gao Y, Yang M, Zhou W (2020) Assessment of short-term improvement effectiveness of preventive maintenance treatments on pavement performance using ltpp data. J Transp Eng Part B 146(3):04020048
13. Bismor D (2019) System for vehicle sound and vibration monitoring using mems sensors. In: 2019 signal processing: algorithms, architectures, arrangements, and applications (SPA). IEEE, pp 50–55
14. Ma X, Dong Z, Dong Y (2021) Toward asphalt pavement health monitoring with built-in sensors: a novel application to real-time modulus evaluation. IEEE Trans Intell Transp Syst 23(11):22040–22052
15. Sencer B, Kakinuma Y, Yamada Y (2020) Linear interpolation of machining tool-paths with robust vibration avoidance and contouring error control. Precis Eng 66:269–281
16. Noor A, Zhao Y, Khan R, Wu L, Abdalla FY (2020) Median filters combined with denoising convolutional neural network for Gaussian and impulse noises. Multimedia Tools Appl 79:18553–18568
17. Nimehvari Varcheh H, Rezaei P (2022) Low-loss x-band waveguide bandpass filter based on rectangular resonators. Microw Opt Technol Lett 64(4):701–706
18. Lu W, Shi X, Sun J, Ge W, Zhang R (2022) Improving the signal-to-noise ratio of GM-APD coherent lidar system based on phase synchronization method. Opt Laser Technol 150:107994
19. Nie D, Xie K, Zhou F, Qiao G (2020) A correlation detection method of low SNR based on multi-channelization. IEEE Signal Process Lett 27:1375–1379

20. Kishore K, Akbar S (2020) Evolution of lock-in amplifier as portable sensor interface platform: a review. IEEE Sens J 20(18):10345–10354
21. Ma X, Dong Z, Chen F, Xiang H, Cao C, Sun J (2019) Airport asphalt pavement health monitoring system for mechanical model updating and distress evaluation under realistic random aircraft loads. Constr Build Mater 226:227–237
22. Qi Z, Wang H (2021) Dirty-data impacts on regression models: an experimental evaluation. In: International conference on database systems for advanced applications, vol 1, pp 88–95
23. Qi Z, Wang H, Wang A (2021) Impacts of dirty data on classification and clustering models: an experimental evaluation. J Comput Sci Technol 36:806–821