IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON BIG DATA TECHNOLOGY AND APPLICATIONS IN INTELLIGENT TRANSPORTATION

Received April 12, 2020, accepted June 17, 2020, date of publication June 24, 2020, date of current version July 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3004590

Road Anomaly Detection Through Deep Learning Approaches

DAWEI LUO[1,2], JIANBO LU[2], AND GANG GUO[1]
[1]Department of Automotive Engineering, Chongqing University, Chongqing 400044, China
[2]Research and Advanced Engineering, Ford Motor Company, Dearborn, MI 48124, USA

Corresponding author: Gang Guo (guogang@cqu.edu.cn)

This work was supported in part by the Chongqing Science and Technology Committee under Grant cstc2018jszx-cyzdX0074, and in part by Ford Motor Company.

ABSTRACT This paper addresses road anomaly detection by formulating it as a classification problem and applying deep learning approaches to solve it. Besides conventional road anomalies, additional ones are introduced from the perspective of a vehicle. In order to facilitate the learning process, the paper pays a close attention to pattern representation, and proposes three sets of numeric features for representing road conditions. Also, three deep learning approaches, i.e. Deep Feedforward Network (DFN), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN), are considered to tackle the classification problem. The detectors, with respect to the three deep learning approaches, are trained and evaluated through data collected from a test vehicle driven on various road anomaly conditions. The comparison study on the detection performances is conducted by setting key hyper-parameters to certain sets of fixed values. Also, the comparison study on performances of each detector with respect to different pattern representations is conducted. The results have shown the effectiveness of the proposed approaches and the efficiency of the proposed feature representations in road anomaly detection.

INDEX TERMS Convolutional neural network, deep feedforward network, deep learning, pattern representation, recurrent neural network, road anomaly detection.

I. INTRODUCTION

Anomaly is something that deviates from what is standard or the normal. Anomaly detection (AD) in real-world applications aims at determining if there are instances that are dramatically dissimilar to all the other instances [1]. A generic form is the so-called *outlier detection* which refers to finding patterns in data that do not conform to the expected normal behavior [2]. A more concrete concept on an outlier is given by [3], which is an observation that deviates so significantly from the other observations as to arouse suspicion that it was generated by a different mechanism.

A rich body of literature [2]–[6] reviewed the approaches for solving AD problems. As the advancement of deep learning [7], generic methods and algorithms [1], [8] have emerged by incorporating traditional AD approaches with neural networks. This marriage has prompted various novel AD methods that have been successfully applied to obstacles and anomalies detection in different application domains [9]–[18].

In this paper, AD for road condition is of interest and its main application is for intelligent transportation systems (ITS) and smart mobility. By using in-vehicle sensors and their measurements, detecting anomalous road conditions such as rough road, potholes, and speed bumps is pivotal in ITS since those anomalies can cause ride discomfort and potential damage to the vehicle's chassis systems such as wheel and tire assemblies, steering system, and suspension systems. Furthermore, for an autonomous vehicle (AV) where there are no human drivers to remember and monitor those road conditions, such a lack of self-awareness could be a limiting factor for its healthy operation. Hence, it might be desirable to monitor the road anomalies encountered by the vehicle itself. This information can be used as one of the factors to schedule on-demand maintenance so as to prevent unnecessary vehicle breakdown. This motivates us to look into using vehicle as a ''sensor'' to detect the road anomalies.

The associate editor coordinating the review of this manuscript and approving it for publication was Sabah Mohammed.

For road anomaly detection, unlike most of the research works which focus on potholes and bumps only, we consider a broader scope of anomalies. On the other hand, instead of using range sensors or image sensors, raw data are acquired by leveraging in-vehicle sensors. Those raw data are the measurements of vehicle responses. Thus, road condition will be inferred indirectly. Nowadays, lots of works focus on using smartphone sensors, however, signatures of road anomalies may be filtered out due to the fact that the smartphones are not rigidly fixed on the vehicle. Moreover, as majority of literatures rely on vertical signals, some complex types of road anomalies can not be inferred except potholes or bumps. For those cases, other vehicle response signals can be used, together with inertial signals, to form multivariate time series data. This will result in richer and more robust feature representation, such that the modeling capabilities of deep learning models can be fully explored. We tackle road anomaly detection by formulating it as a classification problem and apply deep learning approaches by leveraging multivariate time series data. The main contributions of this paper are:

- The eight types of road (pavement) anomalies are specified from the perspective of a vehicle.
- The three sets of numerical features for representing road conditions are proposed. The comparison study on the detection performances of a deep learning model with respect to (w.r.t.) the three sets of features is conducted.
- Road anomaly detection is formulated as a classification problem. A Deep Feedforward Network (DFN) is applied to solve the classification problem. In addition, a Convolutional Neural Network (CNN), and a Recurrent Neural Network (RNN) are also applied. Furthermore, comparison study on the detection performance of the three approaches is conducted by setting key hyper-parameters to certain pre-defined values.

The rest of this paper is organized as follows: section II reviews the literatures that focus on road anomaly detection; section III describes the problem formulation of road anomaly detection; section IV includes the data acquisition, pattern representation for road conditions, and dataset construction; section V models road anomaly and presents the architectures of the three deep learning approaches; section VI shows the training results and compares the performances of the three deep learning approaches; section VII concludes the paper.

II. RELATED WORK

Road anomaly detection can be tackled by defining and modeling the reference (normal behavior) through physical model based approaches [19], [20] and then finding the similarity metric between the normal and the anomaly behavior [21]. When the dynamical process of road anomaly is characterized as low level of nonlinearity, free from time delays, and less noisy in its observations, the physical model based approaches are very efficient. If the process, however, is highly nonlinear or noisy, data driven approach might be more efficient and effective.

Road anomaly detectors based on support vector machine (SVM) are studied [22]–[28], almost all of which use smartphone sensors for data collection except [22]. Robust features of road anomaly are extracted by applying signal processing techniques, such as wavelet decomposition, resampling, and thresholding, to acceleration signal or other types of inertial signals of vehicle responses. Besides SVM, deep learning approaches also investigated in [29], [30]. Reference [29] applies a conventional DFN to learn a detector by feeding with inertial signals acquired by both in-vehicle sensors and smartphone sensors. Reference [30] applies a unique deep learning algorithm which combines wavelets, neural network, and Hilbert transform, to extract robust features of an acceleration signal only. Moreover, [31] uses unsupervised learning technique (clustering), to detect road anomalies by analyzing the changes of driving behavior. For most signal based road anomaly detection approaches, lots of works rely on smartphone sensors. Thus, [32], [33] paid close attention to implement a user-friendly framework for road anomaly detection. Reference [32] focuses on developing robust sensor measurements as well as sharing information among users. Reference [33] proposes a smartphone probe car (SPC) system to objectively assess and monitor road conditions, which endows the system with awareness of mounted posture of smartphones. It also performs lots of signal processing in device level as well as in system level.

For all the above research works, vehicle response data acquired by either smartphone sensors or in-vehicle sensors. Instead of using vehicle response signals, researchers also take advantage of image data to perform road anomaly detection directly [34]–[38]. Conventional image processing techniques are applied extensively in those works. In addition, machine learning approaches, such as SVM [36], special designed CNN [37], and conventional CNN [38] are also investigated and integrated with traditional image process methods.

Although image based approaches can detect road anomalies in advance, they may fail due to potential issues with lighting and weather conditions. On the other hand, the smartphone sensors can pick up information which are not necessarily related to vehicle responses and not all the vehicle responses of the eight types of road anomalies defined in this paper can be inferred from the smartphone sensors. Those motivate us to use in-vehicle sensors for data acquisition and propose three sets of numerical features for pattern representation of normal and anomalous road conditions. As few research works apply CNN along with time series data in road anomaly detection domain, we attempt to learn a CNN detector and compare its performance with DFN detector. Also, there are few approaches based on RNN, we develop a RNN detector which is capable of learning the complex dynamics

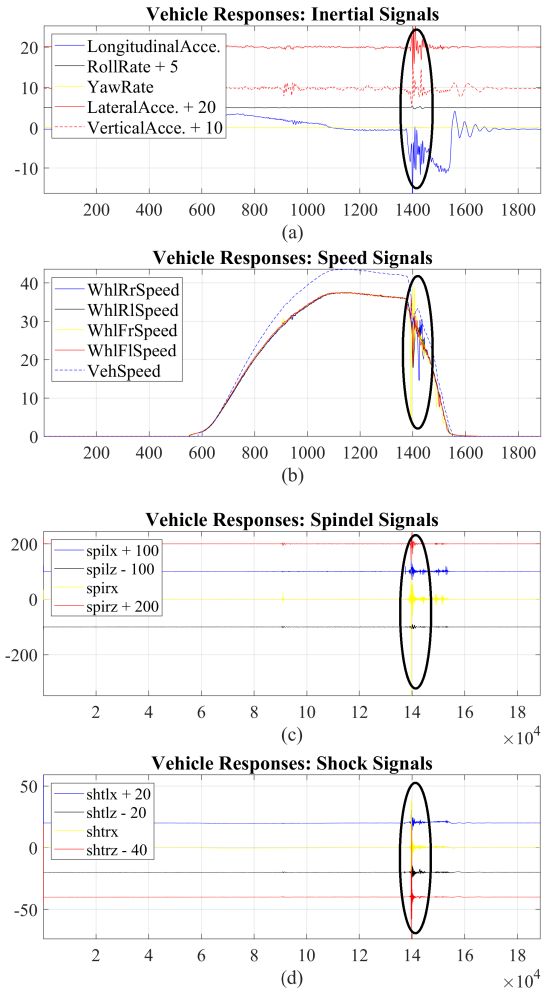

FIGURE 1. Example plot of vehicle responses of a pothole test. The patterns inside the ovals corresponds to the pothole negotiation.

of vehicle responses when encountering road anomalies, and differentiate them from the normal road conditions. The performance of the learned RNN detector is compared with the other two detectors we investigated.

III. PROBLEM FORMULATION

Vehicle dynamic responses such as roll and yaw rate, accelerations, and wheel speeds, can be sensed by existing in-vehicle sensors. When it encounters anomalous road conditions such as potholes and bumps, special patterns will show up in the responses, which are captured by intervals of time sequences through the measured signals. For example, Fig. 1 shows time sequences of the measured signals and certain patterns showing inside the ovals corresponds to a road anomaly. This change of pattern from data point of view prompts us using the responses to infer if a road anomaly has been negotiated, so as to declare that the road conditions as an anomaly. Instead of using physical model based methods, deep learning approaches are proposed. More specifically, the anomaly detection is formulated as a binary classification problem, which differentiates the normal road condition from an anomalous road condition of any of the eight types in set $\mathbb{RA}$:

$\mathbb{RA} = \{$
pothole,
bump,
gravel,
cobblestone,
broken concrete,
curb impact,
the road condition that causes high wheel impact,
the road condition that causes severe vehicle body twist
$\}$.

All types of road anomalies in $\mathbb{RA}$ are used to describe the status of anomalous road pavement. It is rarely to see road anomalies, such as curb impact, vehicle body twist, and vehicle wheel impact, showing up in literatures. Those three types are introduced herein as we categorize road anomalies from a vehicle's perspective, which are unsatisfactory behaviors to a vehicle.

Let $X_R = \{x_R{}^{(k)} \in \mathbb{R}^m : k \in \{1, 2, \ldots, N\}\}$ be a multivariate time series which is a sequence with length N. It includes the collection of the *raw samples* corresponding to the natural samples obtained through real-time sampling of the selected m vehicle response signals, which are assumed to be sampled at the same frequency. Obviously, the *raw sample* $x_R{}^{(k)}$ is a m-tuple, and sampled at time instant k, which is also regarded as a column vector in this paper:

$$x_R{}^{(k)} = [x_{1R}{}^{(k)}, x_{2R}{}^{(k)}, \ldots, x_{mR}{}^{(k)}]^T, \quad k \in \{1, \ldots, N\} \tag{1}$$

We define the ground truth label of $x_R{}^{(k)}$ as

$$y_R{}^{(k)} = L_R[x_R{}^{(k)}] \tag{2}$$

where the labeling function L_R will be addressed in Section IV. Hence, the raw label set corresponds to X_R follows:

$$Y_R = \{y_R{}^{(k)} : k \in \{1, 2, \ldots, N\}\} \tag{3}$$

For time series data, sliding window techniques are usually used to divide the time-series into discrete segments in order to reveal the underlying properties of its source. Hence, a sliding window with fixed length, TR, is applied to X_R. Let $X_T = \{x_T{}^{(i)} \in \mathbb{R}^{m \times TR} : i \in \{0, 1, \ldots, (N_T - 1)\}\}$ be the set resulting from applying the sliding window, whose elements are the segments of X_R. In this paper, we denote segment $x_T{}^{(i)}$ as the *raw training sample*. Thus, we refer TR as the training-sample-to-raw-sample ratio since one (raw) training sample for the learning models originates from TR consecutive *raw samples*:

$$x_T{}^{(i)} = [x_R{}^{(i*(TR-N_{ors})+1)}, x_R{}^{(i*(TR-N_{ors})+2)}, \ldots, x_R{}^{(i*(TR-N_{ors})+TR)}] \tag{4}$$

where $TR \leq N$, and N_{ors} is the number of overlapped *raw samples* between $x_T{}^{(i)}$ and $x_T{}^{(i+1)}$. Therefore, the total

Algorithm 1 Construction and Labeling of Training Samples

Input: X_R, TR, N_{ors}
Output: Dataset $\mathbb{SL}$

```
Initialization: rawLabels = [ ], SL = {}
for k = 1 to N do
    y_R^(k) ← apply (2) to x_R^(k)
    rawLabels ← append y_R^(k) to rawLabels
end for
for i = 0 to (N_T − 1) do
    x_T^(i) ← apply (4) to X_R
    x_T^(i)^f ← apply (6) to x_T^(i)
    y_T^(i) ← apply (8) to rawLabels
    SL ← add ⟨x_T^(i)^f, y_T^(i)⟩ to SL
end for
return SL
```

number of *raw training samples* N_T can be computed from the total number of the *raw samples* N:

$$N_T = \lfloor \frac{N - TR}{TR - N_{ors}} \rfloor + 1 \tag{5}$$

where $\lfloor \bullet \rfloor$ is the floor function.

To generate the final *training sample set* ${X_T}^f$ for the learning algorithms, a function *modelDependConversion()* is used, such that

$${x_T}^{(i)^f} = modelDependConversion({x_T}^{(i)}) \tag{6}$$

$${X_T}^f = \{{x_T}^{(i)^f} : i \in \{0, 1, \ldots, (N_T - 1)\}\} \tag{7}$$

where ${x_T}^{(i)^f} \in {X_T}^f$ is interpreted as the pattern representation of road conditions, the details of the model dependent function will be addressed in Section V.

We define the corresponding ground truth label of ${x_T}^{(i)^f}$ as

$${y_T}^{(i)} = L_T[{y_R}^{(i*(TR-N_{ors})+1)}, {y_R}^{(i*(TR-N_{ors})+2)}, \ldots, {y_R}^{(i*(TR-N_{ors})+TR)}] \tag{8}$$

where L_T is a projection function and will be discussed in section IV. Thus, the labeled set associated with the final training sample set ${X_T}^f$ is defined as Y_T, called the *training label set*:

$$Y_T = \{{y_T}^{(i)} : i \in \{0, 1, \ldots, (N_T - 1)\}\} \tag{9}$$

Up to this point, the process of training sample construction and labeling from time series data has been complete. The resulting dataset $\mathbb{SL}$ is then a set of pairs as follows:

$$\mathbb{SL} = \{\langle {x_T}^{(i)^f}, {y_T}^{(i)} \rangle : i \in \{0, 1, \ldots, (N_T - 1)\}\} \tag{10}$$

The pseudocode of the process is shown in Algorithm 1. For the binary classification of road anomaly detection considered in this paper

$$Y_T = Y_R = \{0, 1\} \tag{11}$$

where label 0 means normal road condition and 1 means anomalous road condition (road anomaly).

The problem to be solved here is to find a parameterized mapping with parameters θ taking values from a parameter set Ξ:

$$h_\theta : {X_T}^f \rightarrow Y_T, \theta \in \Xi \tag{12}$$

where h_θ has a given structure, such that the cost function

$$J(\theta) = \sum_{i=0}^{N_T - 1} H[h_\theta({x_T}^{(i)^f}), {y_T}^{(i)}] \tag{13}$$

where H is a function to measure the dissimilarity between the estimated label $h_\theta({x_T}^{(i)^f})$ and ground truth ${y_T}^{(i)}$ is minimized. The optimal θ^* can be found by:

$$\theta^* = \arg\min_{\theta \in \Xi} J(\theta) \tag{14}$$

The dissimilarity function H used in this paper is the cross entropy between the estimated label and the ground truth label. The resulting mapping h_{θ^*}, selected by holdout cross validation, gives a low empirical classification error over the training sample set for a given h_θ. Hence for a new observed sample x outside the training sample set, the estimated label $\hat{y}$ computed as following

$$\hat{y} = \begin{cases} 1, & h_{\theta^*}(x) \geq 0.5 \\ 0, & h_{\theta^*}(x) < 0.5 \end{cases} \tag{15}$$

should be a good estimation of its ground truth label. h_θ can fall within a much broader space including linear mapping, logistic regressions, and supported vector machines. In this paper, we focus on limiting h_θ in the deep learning domain, where hypotheses are represented by computational graphs and parameters θ are learned by backpropagation method [39]. We will discuss the deep learning models for road anomaly detection in section V.

IV. DATASET PREPARATION

In order to learn a good classification mapping h_θ, an appropriate pattern representation of road condition and the completeness of data for the eight types of road anomalies are demanded.

A. DATA ACQUISITION

For the sake of data completeness, eight types of road anomalies along with normal road conditions should be included in our training dataset. Hence, testing scenarios have been designed, and field tests are carried out for each type of the road anomalies at Dearborn Development Center of Ford Motor Company. A 2013 Lincoln MKS passenger car, as shown in Fig. 2 (a), is used for the field testing.

Let's set the road anomaly, pothole, as an example. The following 10 testing designs are considered:

- Braking applied when negotiating potholes:
  - Keep speed around 15 mph, before and after;
  - Keep speed around 20 mph, before and after;
  - Keep speed around 25 mph, before and after;
  - Keep speed around 30 mph, before and after;

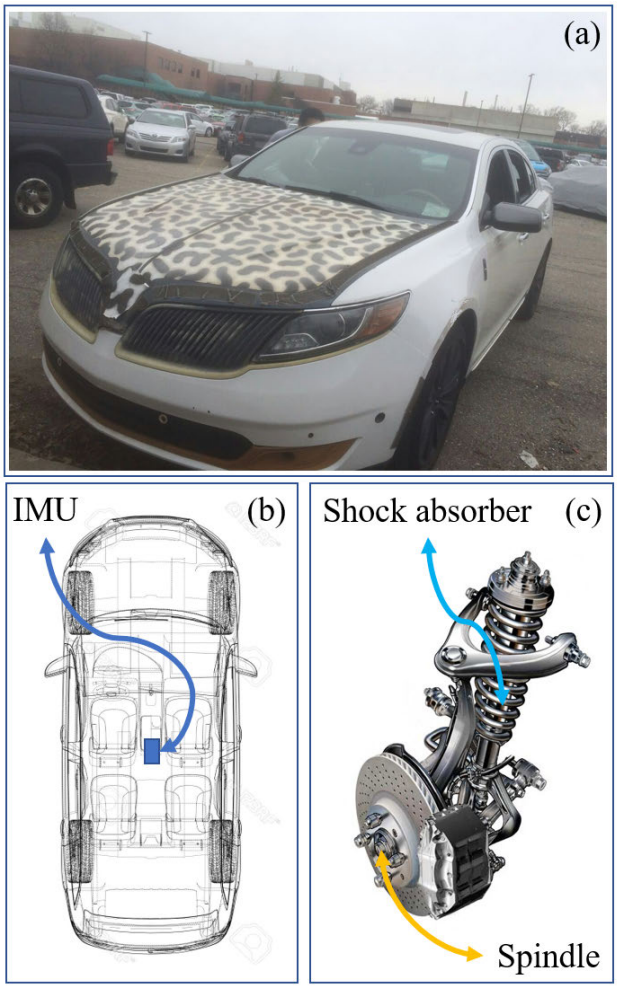

FIGURE 2. Testing Vehicle and Sensor Locations: (a) the vehicle used for testing, (b) the location of in-vehicle IMU, (c) the location of spindle and shock absorber.

  - Keep speed around 35 mph, before and after;
- No braking applied when negotiating potholes:
  - Keep speed around 15 mph always;
  - Keep speed around 20 mph always;
  - Keep speed around 25 mph always;
  - Keep speed around 30 mph always;
  - Keep speed around 35 mph always;

As we can see, the first five testing scenarios are similar to the driving behaviors of most human drivers, where braking is applied, when road anomalies are observed in roadway. However, the last five testing scenarios are used to imitate the driving behaviors of aggressive drivers or mild drivers who do not observe the anomalies. As the patterns of the normal road conditions can be extracted directly from the road segment excluding road anomalies of the testings, there is no need to conduct dedicated vehicle field testings for collecting data w.r.t. the normal road conditions.

Therefore, there are 80 tests totally. During each test, vehicle response data from high-speed CAN (control area network) are collected via On-Board Diagnostics (OBD) interface.

B. PATTERN REPRESENTATION

An appropriate pattern representation of road conditions might improve a deep learning model's performance. In this paper, we treat signals as the features of road conditions. Three sets of signals (numerical features) representing road conditions are proposed:

1) Standard set (6 signals)

The set includes sensor signals of

$\mathbb{S} = \{$
Vehicle Longitudinal Acceleration,
Vehicle Vertical Acceleration,
Vehicle Lateral Acceleration,
Vehicle Roll Rate,
Vehicle Yaw Rate,
Vehicle Speed
$\}$.

2) Performance set (10 signals)

It is the union of Standard set and wheel speeds (rotational speeds of four wheels), i.e.

$\mathbb{P} = \{$
Standard set,
Rotation speeds of the 4 wheels
$\}$.

3) All-signal set (18 signals)

It is the union of Performance set and other eight signals, i.e.

$\mathbb{A} = \{$
Performance set,
Spindle responses of the 4 spindles,
Shock absorber responses of the 4 absorbers
$\}$.

The signals in $\mathbb{S}$ are the most common ones that are used extensively in road anomaly detection domain. Therefore, we refer it as the standard set. Those, except vehicle speed, are the measurements of the in-vehicle Inertial Measurement Unit (IMU), which is located at the vehicle body as shown in Fig. 2 (b). In order to tackle complex anomalous road conditions, however, numerical features given by sets $\mathbb{P}$ and $\mathbb{A}$ are used, where response signals of suspension components, see Fig. 2 (c), are introduced.

C. DATASET CONSTRUCTION

Data preprocessing is required since sampling rates of in-vehicle sensors are different. Therefore, all the signals of interest are resampled to 100 Hz. In Section III, the process of constructing and labeling the training samples from the segments of a multivariate time-series X_R is given, where two labeling functions, i.e. L_R and L_T, are applied.

- Raw sample labeling L_R:

For each field test, not all the elements of the m-tuple $x_R{}^{(k)}$ will reach the anomaly conditions at the exact same time instant as the result of potential time delays of the vehicle

Algorithm 2 Dataset Construction

Input: *testData*, *TR*, *signalSet*
Output: $\mathbb{SL}_{train}, \mathbb{SL}_{dev}, \mathbb{SL}_{test}$

```
Initialization: SL_all = {}, N_ors = TR − 1
for idxTest = 1 to 80 do
    X_R ← get the idxTest-th testing data from testData
    X_R ← resample X_R
    SL ← call Algorithm 1 by feeding [X_R, TR, N_ors]
    SL_all ← add all elements of SL to SL_all
end for
SL_train, SL_dev, SL_test ← shuffle and split SL_all
return SL_train, SL_dev, SL_test
```

dynamic system. Therefore, the rule we proposed is:

$$L_r[x_R{}^{(k)}] = \begin{cases} 1, & \exists j,\ s.t.\ x_{jR}{}^{(k)} \textit{ is different from norm} \\ 0, & \textit{otherwise} \end{cases} \tag{16}$$

where j indexes the j-th element of the tuple. By applying such a rule, we label $x_R{}^{(k)}$ as anomaly if there exists one of its elements behaves abnormally. The labeling of each segment is done, manually, by observing the time-series X_R.

• Training sample labeling L_T:

Recalling (8), the label of a training sample $x_T{}^{(i)^f}$ depends on the labels of its corresponding collection of raw samples, which is given by (4). For a given length (time horizon), *TR*, not all the raw samples in (4) of the segment $x_T{}^{(i)}$ may exhibit anomalous behaviors. Hence, the labeling rule we proposed is:

$$\begin{aligned} y_T{}^{(i)} &= L_T[y_R{}^{(i*(n-N_{ors})+1)}, \ldots, y_R{}^{(i*(n-N_{ors})+n)}] \\ &= \begin{cases} 1, & \textit{majority of the raw labels } y_R \textit{ is } 1 \\ 0, & \textit{otherwise} \end{cases} \end{aligned} \tag{17}$$

As we go through the labeling process, we see that the segments of X_R are of great importance. That means the choices of *TR* and N_{ors} are significant since those parameters effect the time span of a segment $x_R{}^{(k)}$. Hence, we set N_{ors}

$$N_{ors} = TR - 1 \tag{18}$$

and treat *TR* as a hyper-parameter owing to their coupled and cooperative behavior. In this paper, we explore the performances of deep learning models w.r.t. the following *TR* values:

$$\mathbb{TR} = \{1, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56\} \tag{19}$$

By such a setting, we expect to see how the length of a sliding window, *TR*, effects the performance of a model.

After all, by given the length, *TR*, and the set of features, we are able to construct the dataset $\mathbb{SL}_{all}$ including the training samples of the eight types of road anomalies by iterating through the 80 field tests. For model training and evaluation purpose, dataset $\mathbb{SL}_{all}$ is randomly shuffled, and split with 80% of which as training set $\mathbb{SL}_{train}$, 10% as developing set $\mathbb{SL}_{dev}$, and 10% as testing set $\mathbb{SL}_{test}$. The pseudocode of dataset construction is shown in Algorithm 2. Here, the functionality of $\mathbb{SL}_{dev}$ is the same as holdout cross validation set, which is used to tune the hyper-parameters. And $\mathbb{SL}_{test}$ is used for model evaluation.

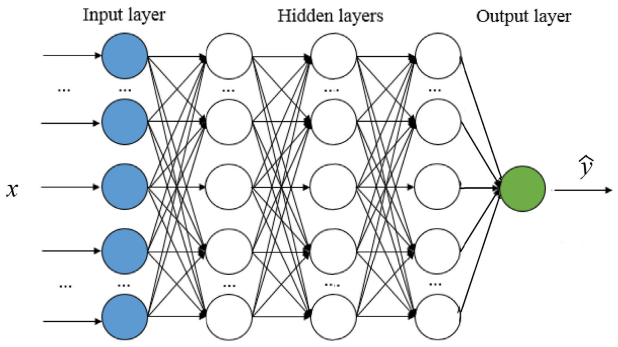

FIGURE 3. Architecture of road anomaly detector: deep feedforward network.

V. MODELING ROAD ANOMALY

Three popular supervised deep learning methods are explored in this paper for the road anomaly detection purpose, i.e., DFN, CNN, and RNN. The structure and hyper-parameters of each model are determined based on experience in practice. Learning rate is carefully selected, for the sake of convergence and learning efficiency, of each model. Since the influences of *TR* and feature representation of road conditions are paid prime attention, each of the three approaches is evaluated at the given *TR* and sets of signals.

A. DEEP FEEDFORWARD NETWORK

A DFN has multiple hidden layers between the input layer and the output layer. The structure of the DFN used here is shown in Fig. 3. For such an architecture, a training sample $x_T{}^f$ is obtained by averaging along the time axis. Hence, the output of *modelDependConversion*() is governed by:

$$x_T{}^{(i)^f} = \frac{\sum_{i=0}^{TR-1} x_T{}^{(i)}[:, k]}{TR} \tag{20}$$

where $x_T{}^{(i)}[:, k]$, i.e. $x_R{}^{(k)}$, is the k-th column vector of $x_T{}^{(i)}$.

For instance, if we set $TR = 12$, use signals in $\mathbb{S}$ as the features of interest, and apply sliding window to X_R, then, it ends up with segments all of which have the same size (6, 12). By averaging along the time axis (the second dimension), the resulting training sample $X_T{}^f$ for DFN will have dimension (6, 1). After the training set $\mathbb{SL}_{train}$, the developing set $\mathbb{SL}_{dev}$, and the testing set $\mathbb{SL}_{test}$ are constructed, DFN will be trained and evaluated for the given *TR* and the given set of signals. In this paper, the three-hidden-layer DFN in Fig. 3 is used. There are 75 nodes in the first hidden layer, 50 nodes in the second hidden layer, and 15 nodes in the last hidden layer. The output layer has one single node. The dimension of the input layer is governed by the number of signals (features)

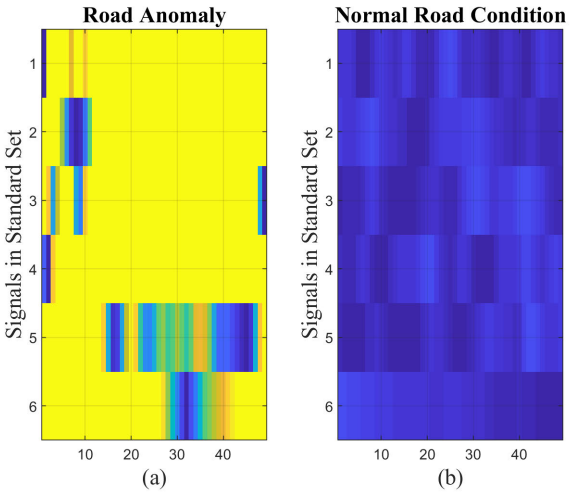

FIGURE 4. Pesudo images: Road anomaly vs. Normal road condition. Image displayed from data array constructed from signals in Standard set.

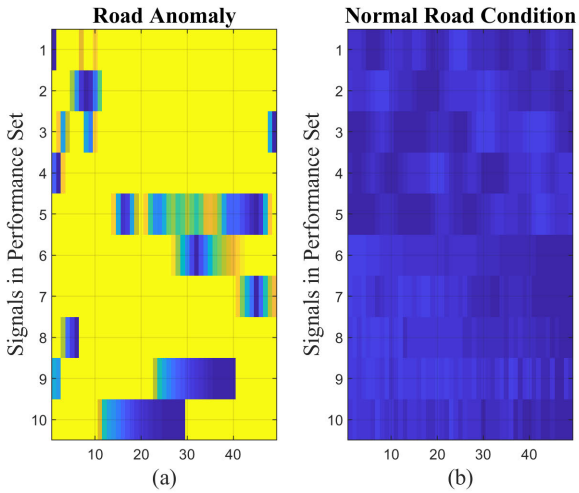

FIGURE 5. Pesudo images: Road anomaly vs. Normal road condition. Image displayed from data array constructed from signals in Performance set.

used. More specifically, the computation of any hidden node is given below, except the output node:

$$node_{out} = \max(0, W^T node_{inp} + b) \tag{21}$$

where dedicated parameters W (vector) and b (scalar) are associated with the node, $node_{inp}$ is the input of the node, and $node_{out}$ is the output of the node. For the output node, its computation is given by

$$\hat{y} = node_{out} = \begin{cases} 1, & sigmoid({W_0}^T node_{inp} + b_0 \geq 0.5) \\ 0, & sigmoid({W_0}^T node_{inp} + b_0 < 0.5) \end{cases} \tag{22}$$

where W_0 and b_0 are the parameters associated with the output node.

The aforementioned network is implemented by leveraging the open sourced tool of Tensorflow [40]. Adam [41] optimizer is used for the training, with a learning rate of 0.00015. The performances of the model will be discussed in the next section.

B. CONVOLUTIONAL NEURAL NETWORK

CNN is a class of artificial neural networks that has been successfully applied to deal with image data in computer vision. Although a time sequence is not like an image, they share some common attributes. Considering a segment ${x_T}^{(i)}$, its elements are likely to have some connections in term of changes, as vehicle responses are sampled vehicle dynamics. This regional and local behavior is somehow similar to the relationship between a pixel and its surrounding pixels of an image. Hence, a raw training sample ${x_T}^{(i)}$, which is a 2 dimensional (2D) array, can be treated and displayed as pseudo images with just one channel as shown in Fig. 4, Fig. 5, and Fig. 6, where pseudo images (a) and (b) of each figure are displayed from the normalized segments corresponding to road anomaly and normal road condition, respectively.

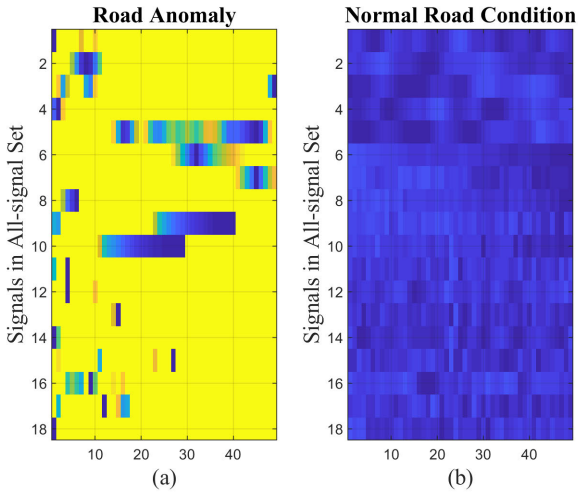

FIGURE 6. Pesudo images: Road anomaly vs. Normal road condition. Image displayed from data array constructed from signals in All-signal set.

As the patterns showing by the pseudo images are quite different, it is plausible for us to make decisions, about the road conditions, by leveraging the segments of X_R. This prompts us to treat ${x_T}^f$ as a one channel pseudo image and take advantages of the properties of CNN. Therefore, the training samples are quite different from that of DFN model. In order to obtain 2D array like training samples for CNN, we do not extract any statistical features from segment x_T. Moreover, reshaping is required by extending its dimension from (No. of *signals*, *TR*) to (No. of *signals*, *TR*, 1), such that pseudo image with one channel is guaranteed. Hence, the output of *modelDependConversion*() is given:

$${x_T}^{(i)^f} = reshape({x_T}^{(i)}) \tag{23}$$

In this paper, the CNN structure we proposed is shown in Fig. 7 where a two-convolutional-layer CNN is used, and

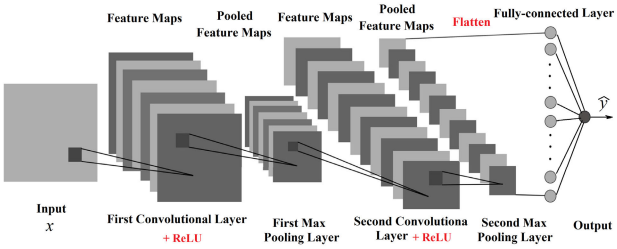

FIGURE 7. Architecture of road anomaly detector: convolutional neural network.

TABLE 1. CNN: size of each layer.

Layer	Size
First Convolutional Layer	[4, 4, 1, 8]
First Max Pooling Layer	[1, 4, 4, 1]
Second Convolutional Layer	[2, 2, 8, 16]
Second Max Pooling Layer	[1, 2, 2, 1]

TABLE 2. CNN: size of each layer when *TR* is 1 or 4.

Layer	Size
First Convolutional Layer	[1, 1, 1, 8]
First Max Pooling Layer	[1, 1, 1, 1]
Second Convolutional Layer	[1, 1, 8, 16]
Second Max Pooling Layer	[1, 1, 1, 1]

each convolutional layer is followed by max pooling layer. A flatten operator applied to the second max pooling layer results in a fully-connected layer, as shown in Fig. 7. The dimension of each layer is given in Table 1. Especially, if $TR = 1$ and $TR = 4$, the size of each layer is given by Table 2.

The computations of convolutional layers can be fully expressed by Fig. 7. The computation of output layer is the same as (22), where the $node_{inp}$ is substituted by the fully-connected layer.

The CNN is implemented using the open source tool of Tensorflow by applying Adam optimizer with a learning rate of 0.001. The performance of the model will be discussed in the next section.

C. RECURRENT NEURAL NETWORK

Recurrent neural network is a class of artificial neural network where connections between units form a directed graph along a sequence. This allows the network to represent dynamic temporal behavior for a time sequence.

There are various recurrent cells available for implementation, e.g., standard RNN cell, GRU (Gated Recurrent Unit) cell, and LSTM (Long Short-term Memory) cell. Exploring the performances of various RNN cells on the same dataset is beyond the scope of this paper. For the task at hand, we choose LSTM cell [42] for the sake of its good performance in many applications.

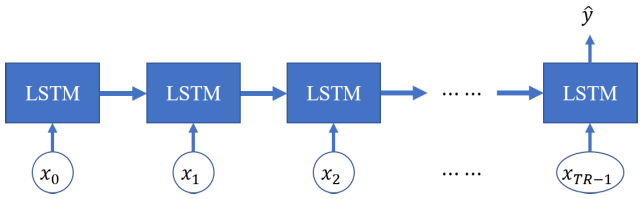

FIGURE 8. Architecture of road anomaly detector: recurrent neural network.

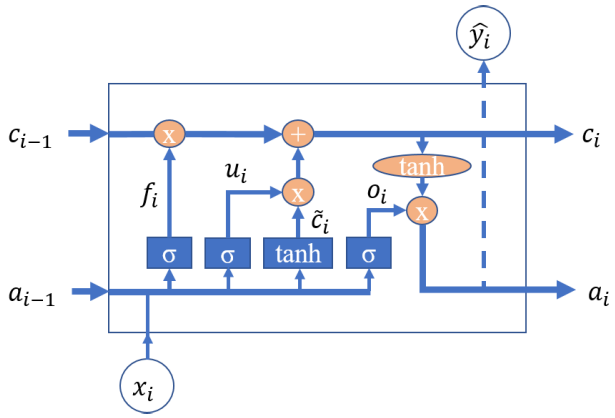

FIGURE 9. The *i*-th LSTM Cell.

The training samples of RNN are similar to that of CNN's, however, without dimension extension. There are no feature extraction operators, either. Hence, the output of *modelDependConversion*() is its argument x_T:

$$x_T{}^{(i)f} = x_T{}^{(i)} = modelDependConversion(x_T{}^{(i)}) \quad (24)$$

The structure of RNN (LSTM) we proposed is shown in Fig. 8. Considering the *i*-th LSTM cell in Fig. 9, the following computations are performed:

$$f_i = sigmoid(W_f[a_{i-1}, x_i] + b_f) \quad (25)$$
$$u_i = sigmoid(W_u[a_{i-1}, x_i] + b_u) \quad (26)$$
$$\tilde{c_i} = \tanh(W_c[a_{i-1}, x_i] + b_c) \quad (27)$$
$$o_i = sigmoid(W_0[a_{i-1}, x_i] + b_0) \quad (28)$$
$$c_i = u_i \circ \tilde{c_i} + f_i \circ c_{i-1} \quad (29)$$
$$a_i = o_i \circ \tanh c_i \quad (30)$$

where x_i, i.e. $x_R{}^{(i)}$, is the *i*-th column vector of a training example $x_T{}^f$, a_i and c_i are the output vector and state vector of the *i*-th LSTM cell, respectively. Similarly, a_{i-1} and c_{i-1} are the output vector and state vector of the $(i-1)$-th LSTM cell, which are also the initial conditions of the *i*-th LSTM cell. Parameters W_f, W_u, W_c, W_0 and b_f, b_u, b_c, b_0 are shared by all LSTM cells. Term $[a_{i-1}, x_i]$ are stacked together such that matrix multiplication is performed correctly. Operator $\circ$ stands for the element wise multiplication of two matrix. The output of a LSTM cell, $\hat{y}_i$, is optional with or without additional operations applied to a_i, which is denoted by the directed dash line.

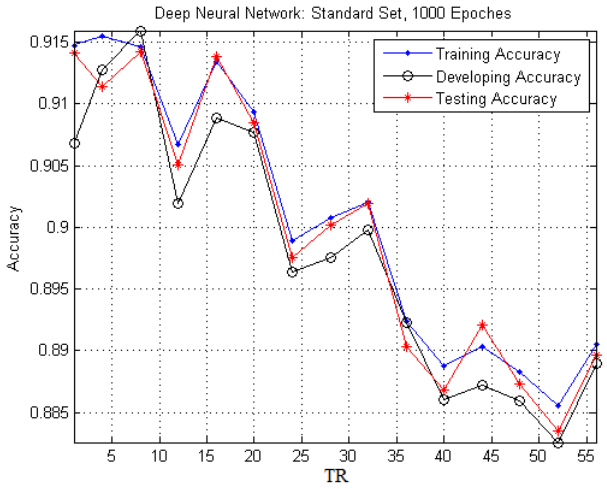

FIGURE 10. Accuracy of DFN based road anomaly detector w.r.t. set $\mathbb{S}$.

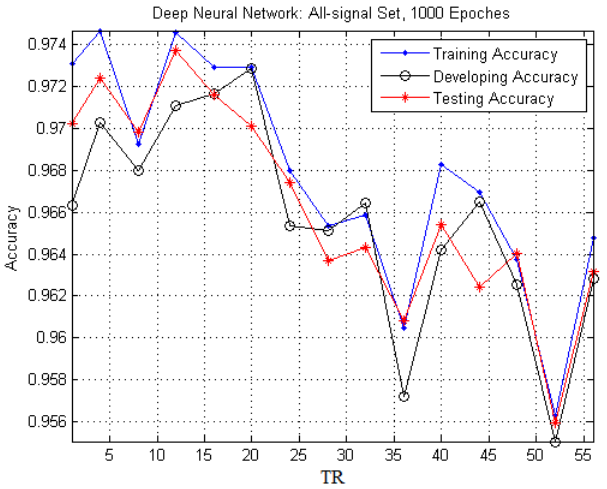

FIGURE 12. Accuracy of DFN based road anomaly detector w.r.t. set $\mathbb{A}$.

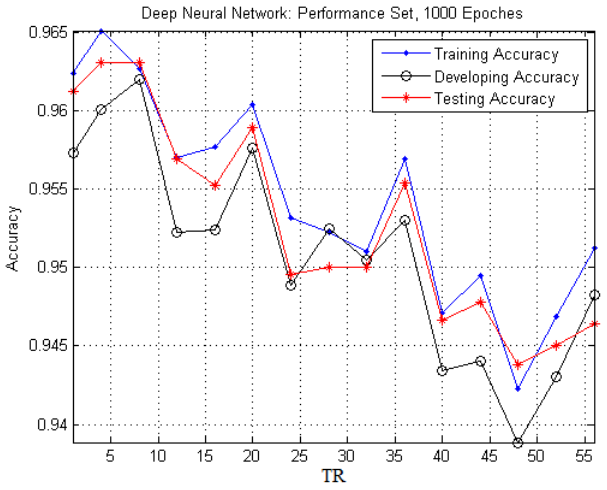

FIGURE 11. Accuracy of DFN based road anomaly detector w.r.t. set $\mathbb{P}$.

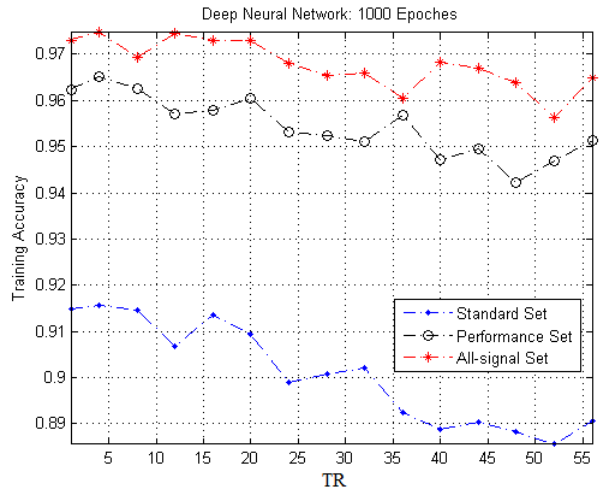

FIGURE 13. Training accuracy of DFN based road anomaly detector.

In Fig. 8, the model outputs an estimation $\hat{y}$ at the last LSTM cell, which is given:

$$\hat{y} = \hat{y}_{TR-1} = \begin{cases} 1, & sigmoid(a_{TR-1}) \geq 0.5 \\ 0, & sigmoid(a_{TR-1}) < 0.5 \end{cases} \tag{31}$$

In this paper, we choose 10 internal nodes of the LSTM cell, and apply Adam optimizer with a learning rate of 0.00085. The RNN model is implemented with the Tensorflow framework. The performance of the model will be discussed in the next section.

VI. RESULTS AND DISCUSSION

For the sake of comparison study, the value of mini-batch size is fixed to 256 and training iterations is set to 1000 of each detector. Since a model with high prediction accuracy is of interest, all training, developing, testing accuracies are recorded with respect to *TR* values and the three types of pattern representations. The prediction accuracy of a model w.r.t. a dataset is given by

$$accuracy = \frac{M_c}{M} \times 100\% \tag{32}$$

where M is the total number of samples in the dataset, and M_c is the number of correct predictions of the learned model.

A. DFN ROAD ANOMALY DETECTOR

The training results of DFN are shown in Fig. 10, Fig. 11 and Fig. 12 for datasets whose features from $\mathbb{S}$, $\mathbb{P}$, and $\mathbb{A}$ signal set, respectively. Training, developing and testing accuracies are, generally speaking, close to each other at every given *TR*. For any given set of signals (feature representation of road conditions), the performance of DFN has the tendency to degrade with the increase of *TR*. The best accuracy of DFN is summarized in Table 3.

In order to compare the accuracies of DFN corresponding to different pattern representations, the accuracy results are plotted from another view in Fig. 13, Fig. 14, and Fig. 15. The

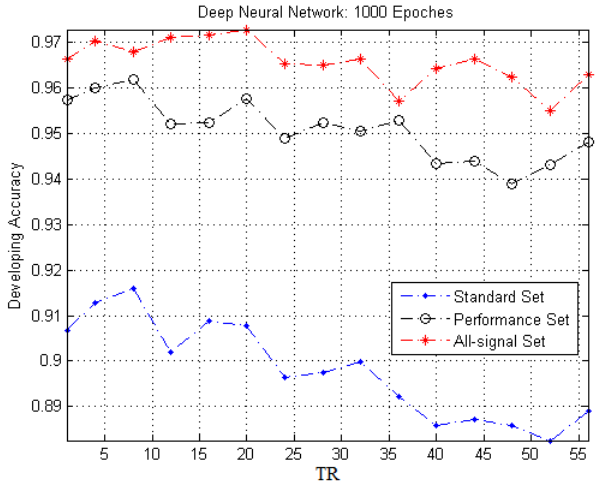

FIGURE 14. Developing accuracy of DFN based road anomaly detector.

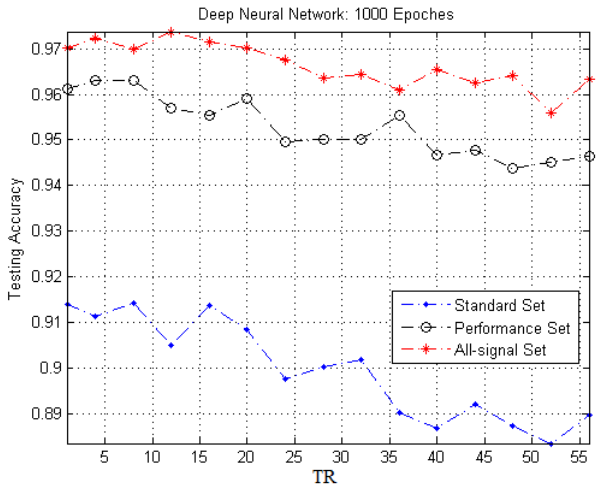

FIGURE 15. Testing accuracy of DFN based road anomaly detector.

TABLE 3. Performance information: DFN based road anomaly detector.

Set of Signals	Number of Parameters	Best (testing) Accuracy	Best *TR* w.r.t. Best Accuracy
Standard Set	5106	91.43%	8
Performance Set	5406	96.31%	4 or 8
All-signal set	6006	97.37%	12

performance information of DFN for different feature sets is summarized in Table 4.

The comparison reveals the importance of the 4 wheel speed signals for road anomaly detection. While physically this makes sense as wheels speeds are the sensor measurements that directly capture the interactions between vehicle and driveway, it is very encouraging that the learned DFN model came up with the same conclusion. Besides, vehicle's shock and spindle responses are also useful for improving the DFN model's accuracy, but not as significant as the wheel

TABLE 4. Performance information of DFN based road anomaly detector w.r.t. sets of signals.

Types of Accuracy	Standard Set ($\mathbb{S}$)	Performance Set ($\mathbb{P}$)	All-signal Set ($\mathbb{A}$)	$\mathbb{P}$ outper-formed $\mathbb{S}$ by	$\mathbb{A}$ outper-formed $\mathbb{P}$ by
Average Training Accuracy	90.08%	95.44%	96.78%	5.36%	1.34%
Average Developing Accuracy	89.80%	95.09%	96.57%	5.29%	1.48%
Average Testing Accuracy	89.98%	95.29%	96.63%	5.31%	1.34%

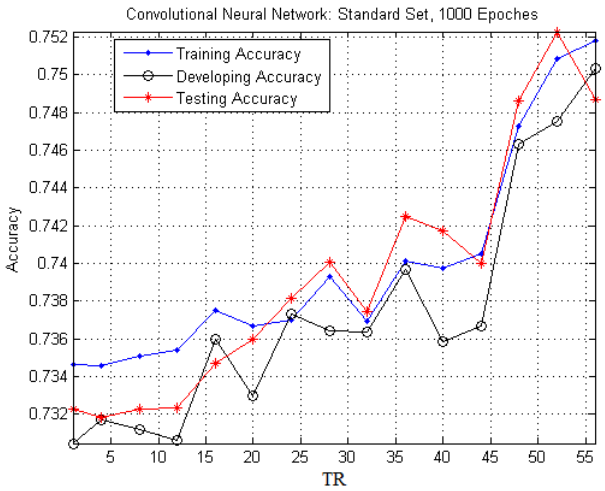

FIGURE 16. Accuracy of CNN based road anomaly detector w.r.t. set $\mathbb{S}$.

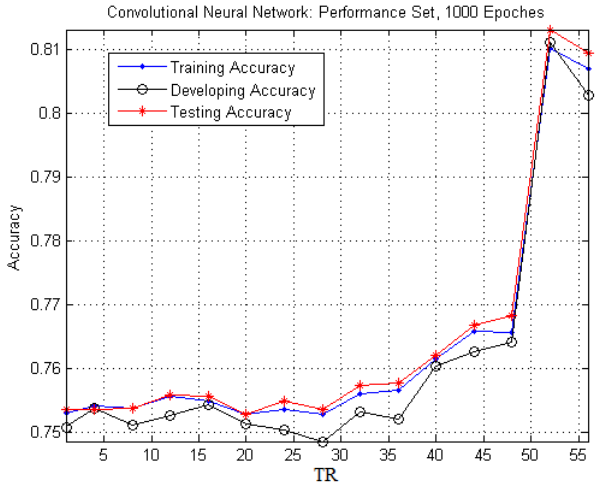

FIGURE 17. Accuracy of CNN based road anomaly detector w.r.t. set $\mathbb{P}$.

speed responses. Therefore, by introducing suspension signals, it improves the detection performance of DFN. And this might lead to a better DFN road anomaly detector compared to the others research works, which also apply DFN, however, excluding features such as wheel speeds.

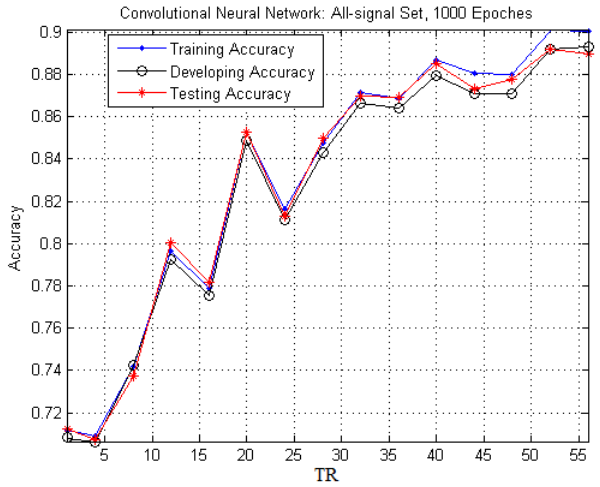

FIGURE 18. Accuracy of CNN based road anomaly detector w.r.t. set $\mathbb{A}$.

TABLE 5. Performance information: CNN based road anomaly detector.

Set of Signals	Number of Parameters w.r.t. Best Accuracy	Best (testing) Accuracy	Best *TR* w.r.t. Best Accuracy
Standard Set	777	75.23%	52
Performance Set	889	81.30%	52
All-signal set	1001	89.19%	52

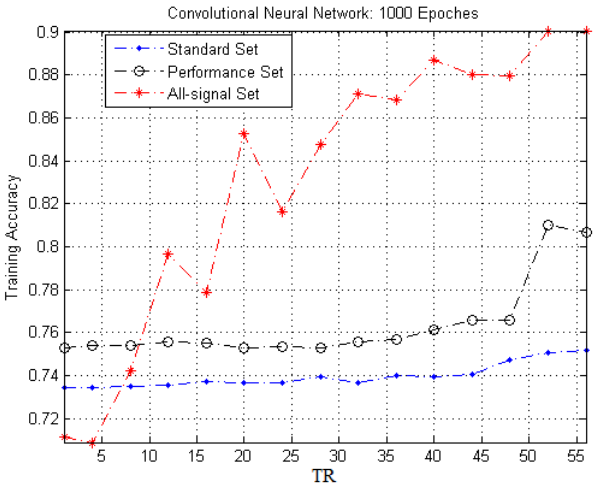

FIGURE 19. Training accuracy of CNN based road anomaly detector.

B. CNN ROAD ANOMALY DETECTOR

The results of CNN road anomaly detector are shown in Fig. 16, Fig. 17, and Fig. 18 w.r.t. datasets whose features from $\mathbb{S}$, $\mathbb{P}$, and $\mathbb{A}$ signal set, respectively. From those three figures, we can find that the CNN model's accuracy is stable and robust at every given value of *TR*. Besides, for any type of feature representation, the accuracy tends to improve as the increase of *TR*, which is different from the DFN detector. The best accuracy of CNN detector is summarized in Table 5.

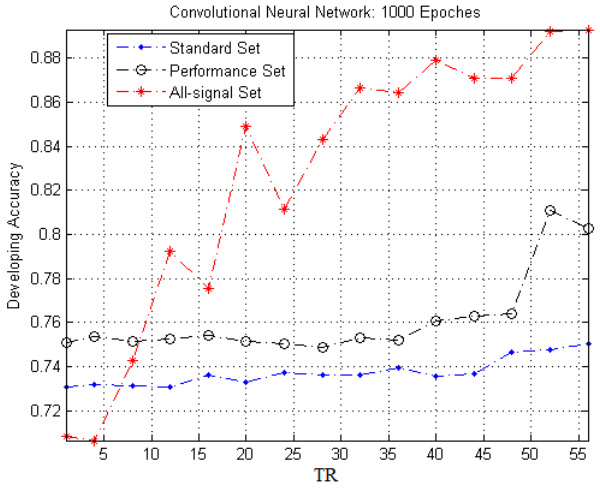

FIGURE 20. Developing accuracy of CNN based road anomaly detector.

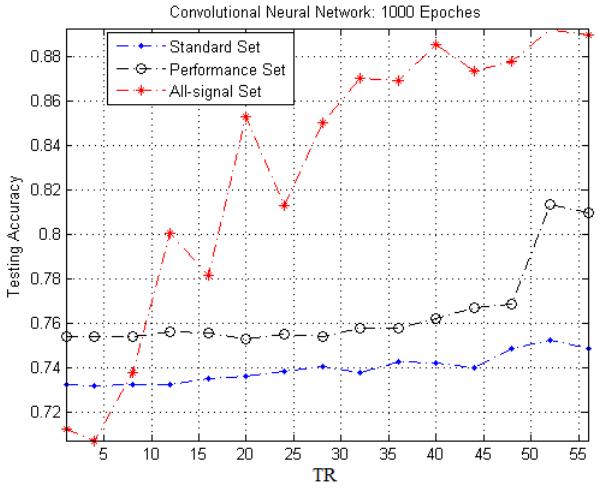

FIGURE 21. Testing accuracy of CNN based road anomaly detector.

TABLE 6. Performance information of CNN based road anomaly detector w.r.t. sets of signals.

Types of Accuracy	Standard Set ($\mathbb{S}$)	Performance Set ($\mathbb{P}$)	All-signal Set ($\mathbb{A}$)	$\mathbb{P}$ outperformed $\mathbb{S}$ by	$\mathbb{A}$ outperformed $\mathbb{P}$ by
Average Training Accuracy	73.98%	76.36%	82.95%	2.38%	6.59%
Average Developing Accuracy	73.73%	76.13%	82.43%	2.40%	6.30%
Average Testing Accuracy	73.92%	76.46%	82.74%	2.54%	6.28%

Fig. 19, Fig. 20, and Fig. 21 show the accuracies at a given *TR* w.r.t. $\mathbb{S}$, $\mathbb{P}$, and $\mathbb{A}$ signal set, respectively. We find that the prediction performance of CNN detector is getting better when *TR* is relatively larger. However, the advantage of large

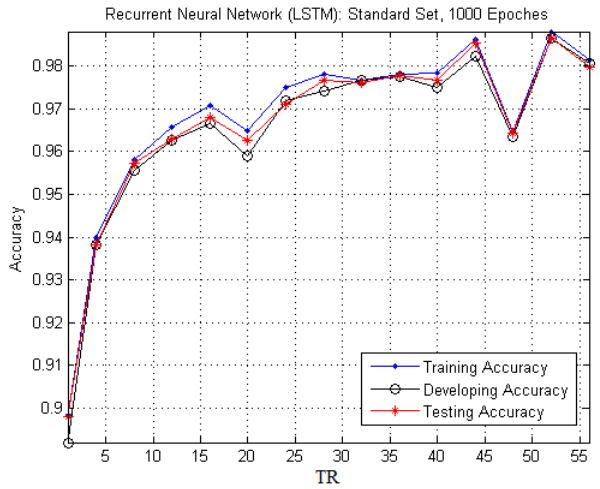

FIGURE 22. Accuracy of RNN based road anomaly detector w.r.t. set $\mathbb{S}$.

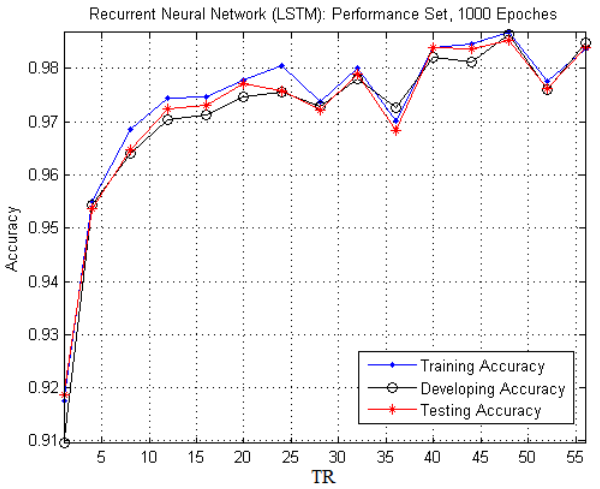

FIGURE 23. Accuracy of RNN based road anomaly detector w.r.t. set $\mathbb{P}$.

TR has been weakened by a poor feature representation of road conditions. The performance information for different feature sets is summarized in Table 6.

From the last two columns of Table 6, we can find that wheel speeds, shock and spindle responses are significant features for CNN road anomaly detector. The prediction performances (less than 90%) is relatively poor with the current configurations, especially when there are just signals available from standard set $\mathbb{S}$. It can be resulted from a variety of reasons, e.g. no more than 1000 parameters, only 2 convolutional layers, or even setting strides inappropriately.

C. RNN ROAD ANOMALY DETECTOR

The results of RNN road anomaly detector are shown in Fig. 22, Fig. 23, and Fig. 24 w.r.t. datasets whose features from $\mathbb{S}$, $\mathbb{P}$, and $\mathbb{A}$ signal set, respectively. From those three figures, we can find that for any set of signals, the prediction performance of RNN has the tendency to improve with the increase of *TR*. Furthermore, there is no overfitting problem with it. The statistical and performance information are shown in Table 7.

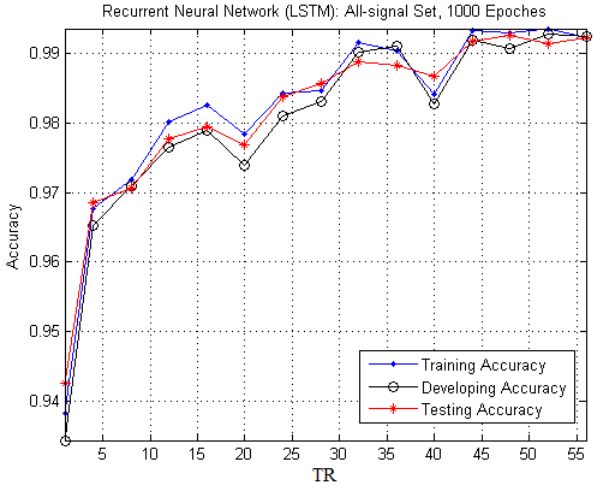

FIGURE 24. Accuracy of RNN based road anomaly detector w.r.t. set $\mathbb{A}$.

TABLE 7. Performance information: RNN based road anomaly detector.

Set of Signals	Number of Parameters	Best (testing) Accuracy	Best *TR* w.r.t. Best Accuracy
Standard Set	691	98.67%	52
Performance Set	851	98.54%	48
All-signal set	1171	99.26%	48

TABLE 8. Performance information of RNN based road anomaly detector w.r.t. sets of signals.

Types of Accuracy	Standard Set ($\mathbb{S}$)	Performance Set ($\mathbb{P}$)	All-signal Set ($\mathbb{A}$)	$\mathbb{P}$ outperformed $\mathbb{S}$ by	$\mathbb{A}$ outperformed $\mathbb{P}$ by
Average Training Accuracy	96.70%	97.26%	98.17%	0.56%	0.91%
Average Developing Accuracy	96.42%	97.03%	97.97%	0.61%	0.94%
Average Testing Accuracy	96.54%	97.12%	98.11%	0.58%	0.99%

To compare the performances corresponding to different types of feature representations, same results are plotted from another view, as shown in Fig. 25, Fig. 26, and Fig. 27. From those figures, we find that the prediction performance of RNN detector is still good even when *TR* is relatively small. However, as *TR* increases, the advantage of more signals is not obvious. The performance information for different feature representations is summarized in Table 8.

From the last two columns of Table 8, we can see the limited improvement of model's performance by adding wheel speed, shock and spindle response signals, which is quite different from that of DFN and CNN detectors. It results from

TABLE 9. Performance summary of road anomaly detectors.

Architecture	Hyper-parameters			Set of Signals	Number of Parameters (w.r.t. Best Accuracy)	Best (Testing) Accuracy	Best *TR* w.r.t. Best Accuracy
	Mini-batch Size	Epochs	Learning Rate				
Deep Feedforward Network (3 Hidden Layer)	256	1000	0.00015	Standard Set	5106	91.43%	8
				Performance Set	5406	96.31%	4 or 8
				All-signal Set	6006	97.37%	12
Convolutional Neural Network (2 Conv. Layer)	256	1000	0.001	Standard Set	777	75.23%	52
				Performance Set	889	81.30%	52
				All-signal Set	1001	89.19%	52
Recurrent Neural Network (LSTM)	256	1000	0.00085	Standard Set	691	98.67%	52
				Performance Set	851	98.54%	48
				All-signal Set	1171	99.26%	48

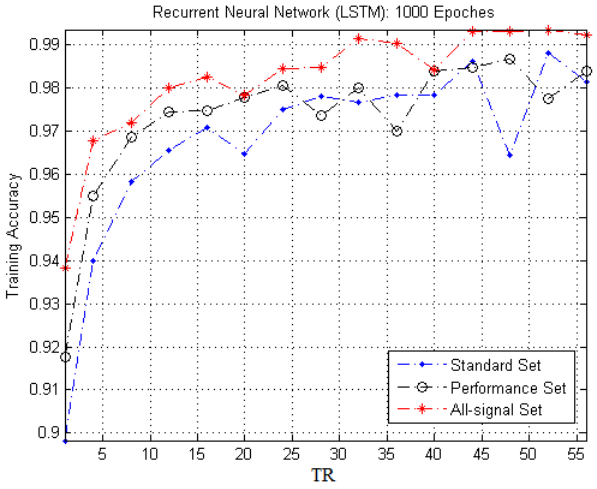

FIGURE 25. Training accuracy of RNN based road anomaly detector.

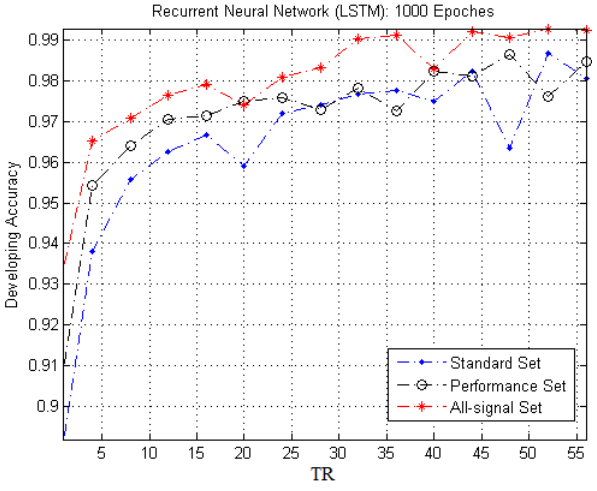

FIGURE 26. Developing accuracy of RNN based road anomaly detector.

the fact that RNN is capable of learning dynamical behaviors of RNN, whereas, DFN and CNN could not extract dynamical information.

D. APPROACH COMPARISON

In this subsection, we compare the performances of the three approaches. In order to do so, some of the hyper-parameters are fixed. As aforementioned, the mini-batch size is set to 256, total iterations is set to 1000 epochs. Moreover, the same optimizer and cost function are used. Table 9 summarizes the performances of all the three road anomaly detectors.

Generally speaking, it is difficult to conduct such a comparison since the internal structures of the three deep learning approaches are completely different. In order to have a fair comparison, we use the number of parameters (weights and bias) of a model as the metric for model complexity.

From Table 9, we can find that RNN detector outperforms DFN with higher prediction accuracy, and with much less parameters. Even though the number of parameters is

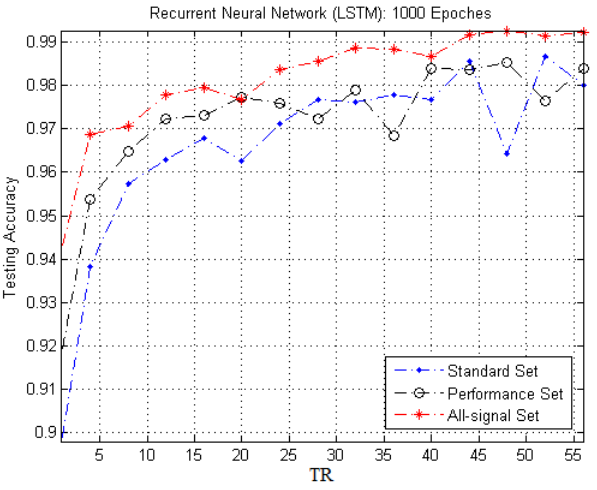

FIGURE 27. Testing accuracy of RNN based road anomaly detector.

comparable, the performance of CNN detector is far behind that of RNN. We can also find that a good road anomaly

detector from DFN or CNN architecture relies on additional signals (features). However, the performance of RNN detector is much more stable with respect to different sets of signals.

From the last column of Table 9, the best performance of RNN carried out along with large *TR*, nevertheless, it does not mean that the prediction performance of RNN is poor when *TR* becomes small, it just indicates that RNN with small *TR* is not as good as that with large *TR*, but it is still comparative to the best performance of DFN detector, and far better than the best performance of CNN detector when *TR* is small. It is not completely surprising that RNN model is better than the other two deep learning methods since it captures state information of the dynamic behavior of the vehicle system.

VII. CONCLUSION

This paper firstly formulates road anomaly detection as a classification problem, and then solves it by applying three types of road anomaly detectors based on the deep learning methods, i.e. DFN, CNN, and RNN. Also, eight types of road anomalies are specified, and three types of feature representations are proposed for dataset construction. Raw data are acquired by field tests, and the road anomaly detectors are trained. The prediction performances of the detectors are compared among the deep learning approaches and among the three feature sets. Our study shows that the RNN detector outperforms the DFN detector with much less model parameters. It also outperforms the CNN detector with much higher prediction accuracy with the similar number of model parameters. The RNN road anomaly detector is much stabler w.r.t. the different sets of feature representations than the other two detectors. For DFN or CNN road anomaly detector, wheel speed signals are strongly recommended for the sake of prediction accuracy. The RNN detector, with a small set of parameters and high prediction performance, seems to be the clear winner among the three detectors for road anomaly detection.

REFERENCES

[1] R. Chalapathy, A. K. Menon, and S. Chawla, "Robust, deep and inductive anomaly detection," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Cham, Switzerland: Springer, 2017, pp. 36–51.

[2] V. Chandola, A. Banerjee, and V. Kumar, "Outlier detection: A survey," *ACM Comput. Surv.*, vol. 14, p. 15, Aug. 2007.

[3] D. M. Hawkins, *Identification of Outliers*, vol. 11. London, U.K.: Springer, 1980.

[4] M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Process.*, vol. 99, pp. 215–249, Jun. 2014.

[5] P. Jinka and B. Schwartz, *Anomaly Detection for Monitoring*. Newton, MA, USA: O'Reilly Media, Inc., 2015.

[6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, p. 15, 2009.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[8] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognit.*, vol. 58, pp. 121–134, Oct. 2016.

[9] P. Christiansen, L. Nielsen, K. Steen, R. Jørgensen, and H. Karstoft, "DeepAnomaly: Combining background subtraction and deep learning for detecting obstacles and anomalies in an agricultural field," *Sensors*, vol. 16, no. 11, p. 1904, Nov. 2016.

[10] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Comput. Netw.*, vol. 51, no. 12, pp. 3448–3470, Aug. 2007.

[11] J. Raiyn and T. Toledo, "Real-time road traffic anomaly detection," *J. Transp. Technol.*, vol. 4, no. 03, p. 256, 2014.

[12] W. Lu, Y. Li, Y. Cheng, D. Meng, B. Liang, and P. Zhou, "Early fault detection approach with deep architectures," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 7, pp. 1679–1689, Jul. 2018.

[13] G. Kang, S. Gao, L. Yu, and D. Zhang, "Deep architecture for high-speed railway insulator surface defect detection: Denoising autoencoder with multitask learning," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 8, pp. 2679–2690, Aug. 2019.

[14] Y. Lyu, Z. Han, J. Zhong, C. Li, and Z. Liu, "A generic anomaly detection of catenary support components based on generative adversarial networks," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 5, pp. 2439–2448, May 2020.

[15] Y. Chen, M. Zhou, Z. Zheng, and M. Huo, "Toward practical crowdsourcing-based road anomaly detection with scale-invariant feature," *IEEE Access*, vol. 7, pp. 67666–67678, 2019.

[16] Z. Zheng, M. Zhou, Y. Chen, M. Huo, and L. Sun, "QDetect: Time series querying based road anomaly detection," *IEEE Access*, vol. 8, pp. 98974–98985, 2020.

[17] X. Li, D. Huo, D. W. Goldberg, T. Chu, Z. Yin, and T. Hammond, "Embracing crowdsensing: An enhanced mobile sensing solution for road anomaly detection," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 9, p. 412, Sep. 2019.

[18] W. Wang, Y. Chi, T. Zhang, M. Deng, and W. Zeng, "Road anomaly detection with group intelligence perception," in *Proc. Int. Conf. Commun., Inf. Syst. Comput. Eng. (CISCE)*, Jul. 2019, pp. 678–682.

[19] Z. Li, I. V. Kolmanovsky, U. V. Kalabic, E. M. Atkins, J. Lu, and D. P. Filev, "Optimal state estimation for systems driven by jump–diffusion process with application to road anomaly detection," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 5, pp. 1634–1643, Sep. 2017.

[20] Z. Li, D. P. Filev, I. Kolmanovsky, E. Atkins, and J. Lu, "A new clustering algorithm for processing GPS-based road anomaly reports with a mahalanobis distance," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1980–1988, Jul. 2017.

[21] D. Shu, C. Lagoa, and T. Cleary, "A sum-of-squares polynomial approach for road anomaly detection using vehicle sensor measurements," in *Proc. Mechatronics, Estimation Identificat., Uncertain Syst. Robustness, Path Planning Motion Control, Tracking Control Syst., Multi-Agent Netw. Syst., Manuf., Intell. Transp. Vehicles, Sens. Actuators, Diag. Detection, Unmanned, Ground Surf. Robotics, Motion Vibrat. Control Appl.*, Oct. 2017, Art. no. V002T17A004.

[22] F. Cong, H. Hautakangas, J. Nieminen, O. Mazhelis, M. Perttunen, J. Riekki, and T. Ristaniemi, "Applying wavelet packet decomposition and one-class support vector machine on vehicle acceleration traces for road anomaly detection," in *Proc. Int. Symp. Neural Netw.* Berlin, Germany: Springer, 2013, pp. 291–299.

[23] M. Saiful, S. Mandal, and S. Islam, "A support vector method for automated road anomaly detection using mobile device," *Int. J. Comput. Appl.*, vol. 127, no. 9, pp. 16–19, Oct. 2015.

[24] N. Silva, J. Soares, V. Shah, M. Y. Santos, and H. Rodrigues, "Anomaly detection in roads with a data mining approach," *Procedia Comput. Sci.*, vol. 121, pp. 415–422, Jan. 2017.

[25] F. Seraj, B. J. van der Zwaag, A. Dilo, T. Luarasi, and P. Havinga, "RoADS: A road pavement monitoring system for anomaly detection using smart phones," in *Big Data Analytics in the Social and Ubiquitous Context*. Cham, Switzerland: Springer, 2014, pp. 128–146.

[26] Y. Tai, C. Chan, and J. Y. Hsu, "Automatic road anomaly detection using smart mobile device," in *Proc. Conf. Technol. Appl. Artif. Intell.*, Hsinchu, Taiwan, 2010, pp. 1–8.

[27] B. Bose, J. Dutta, S. Ghosh, P. Pramanick, and S. Roy, "D&RSense: Detection of driving patterns and road anomalies," in *Proc. 3rd Int. Conf. Internet Things, Smart Innov. Usages (IoT-SIU)*, Feb. 2018, pp. 1–7.

[28] M. R. Carlos, M. E. Aragón, L. C. González, H. J. Escalante, and F. Martínez, "Evaluation of detection approaches for road anomalies based on accelerometer readings—Addressing who's who," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 10, p. 3343, Oct. 2018.

[29] C. Kyriakou, S. E. Christodoulou, and L. Dimitriou, "Roadway pavement anomaly classification utilizing smartphones and artificial intelligence," in *Proc. 18th Medit. Electrotech. Conf. (MELECON)*, Apr. 2016, pp. 1–6.

[30] S.-R. G. Christopoulos, S. Kanarachos, and A. Chroneos, "Learning driver braking behavior using smartphones, neural networks and the sliding correlation coefficient: Road anomaly case study," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 65–74, Jan. 2019.
[31] F. Seraj, K. Zhang, O. Turkes, N. Meratnia, and P. J. M. Havinga, "A smartphone based method to enhance road pavement anomaly detection by analyzing the driver behavior," in *Proc. ACM Int. Joint Conf. Pervas. Ubiquitous Comput. Proc. ACM Int. Symp. Wearable Comput. (UbiComp)*. New York, NY, USA: ACM, 2015, pp. 1169–1177.
[32] F. Orhan and P. E. Eren, "On-road anomaly detection by multimodal sensor analysis and multimedia processing," *Proc. SPIE*, vol. 9026, Mar. 2014, Art. no. 902610.
[33] C. Yi, Y. Chuang, and C. Nian, "Toward crowdsourcing-based road pavement monitoring by mobile sensing technologies," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1905–1917, 2015.
[34] H. Oliveira and P. L. Correia, "Automatic road crack detection and characterization," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 155–168, Mar. 2013.
[35] R. Amhaz, S. Chambon, J. Idier, and V. Baltazart, "Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2718–2729, Oct. 2016.
[36] W. Sultani, S. Mokhtari, and H.-B. Yun, "Automatic pavement object detection using superpixel segmentation combined with conditional random field," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2076–2085, Jul. 2018.
[37] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling, "Feature pyramid and hierarchical boosting network for pavement crack detection," 2019, *arXiv:1901.06340*. [Online]. Available: http://arxiv.org/abs/1901.06340
[38] T. Kamitani, S. Fujimoto, H. Yoshimura, M. Nishiyama, and Y. Iwai, "Anomaly detection using local regions in road images acquired from a hand-held camera," in *Proc. IEEE 7th Global Conf. Consum. Electron. (GCCE)*, Oct. 2018, pp. 375–378.
[39] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, p. 533, 1986.
[40] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. OSDI*, vol. 16. 2016, pp. 265–283.
[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[42] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

DAWEI LUO received the B.S. degree from the College of Mechanical Engineering, Chongqing University, Chongqing, China, in 2013, and the M.S. degree from the Department of Automotive Engineering, Chongqing University, in 2016, where he is currently pursuing the Ph.D. degree.

Since 2016, he has been a Visiting Scholar and a Research Assistant with the Robotics and Intelligent Vehicles Group, Ford Research and Advanced Engineering, Research Innovation Center, Dearborn, MI, USA. He is also visiting the University of Michigan – Dearborn. His research interests include deep machine learning and its application to automotive and transportation systems, driver identification and modeling, and autonomous vehicles.

JIANBO LU received the Ph.D. degree in aerospace engineering from Purdue University, in 1997.

Since 2000, he has been with Ford Motor Company, where he is currently the Technical Expert and a Group Leader in robotics and intelligent vehicles with the Research and Innovation Center. He has published over 80 articles in dynamical system and controls and their applications. He holds 134 U.S. patents in vehicle control, active safety, driver assistance, and intelligent vehicles. His applications of work can be found in a number of current implementations in tens of millions of vehicles. He was elected to IEEE Fellow for contributions to the control of automotive systems with applications in vehicle safety and performance. He was received two times with the highest Ford Corporate Award—the Henry Ford Technology Award, in 2002 and 2013—for developing and implementing automotive technologies with significant business impact. He served as the Vice Chair for Industry and Application at the 2015's American Control Conference. He was the Founder and the Co-Chair of the Intelligent Vehicular Systems and Control Technical Committee under the IEEE Society of System, Man, and Cybernetics. He is on the Editorial Board of the *International Journal of Vehicle Autonomous Systems* and the *International Journal of Vehicle Performance*. He was an Associate Editor of the *IFAC Journal of Control Practice Engineering*, from 2008 to 2014, and the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, from 2010 to 2016.

GANG GUO received the B.S., M.S., and Ph.D. degrees in automotive engineering from Chongqing University, Chongqing, China, in 1982, 1984, and 1994, respectively.

He is currently the Chair and a Professor with the Department of Automotive Engineering, Chongqing University. He is also an Associate Director of the Chongqing Auto-motive Collaborative Innovation Center. He has authored or coauthored over 100 refereed journal and conference publications. His research interests include connected vehicles, multi-sense perception, human–machine interface, brain–computer interface, intelligent manufacturing, and user experience. He is a Senior Member of the China Mechanical Engineering Society. He is also a member of the China User Experience Alliance Committee. He is also the Director of the China Automotive Engineering Society.