



<http://www.diva-portal.org>

This is the published version of a paper published in *Measurement*.

Citation for the original published paper (version of record):

Kulkarni, R., Qazizadeh, A., Berg, M. (2023)

Unsupervised rail vehicle running instability detection algorithm for passenger trains
(iVRIDA)

Measurement, 216: 112894-112894

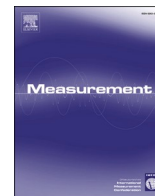
<https://doi.org/10.1016/j.measurement.2023.112894>

Access to the published version may require subscription.

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-326371>



Unsupervised rail vehicle running instability detection algorithm for passenger trains (*iVRIDA*)

Rohan Kulkarni^{*}, Alireza Qazizadeh, Mats Berg

Engineering Mechanics Department, KTH Royal Institute of Technology, 100 44 Stockholm, Sweden

ARTICLE INFO

Keywords:

Vehicle hunting
Unsupervised machine learning
Sparse autoencoder
LSTM encoder decoder
Worn wheel
Failed yaw damper

ABSTRACT

Intelligently identifying rail vehicle faults instigating running instability from carbody floor acceleration is essential to ensure operational safety and reduce maintenance costs. However, the vehicle-track interaction's nonlinearities and scarcity of running instability occurrences complicate the task. The running instability is an anomaly in the vehicle-track interaction. Thus, we propose unsupervised anomaly detection and clustering algorithms based *iVRIDA* framework to detect and identify running instability and corresponding root cause. We deploy and compare the performance of the *PCA-AD* (baseline), Sparse Autoencoder (*SAE-AD*), and LSTM-Encoder-Decoder (*LSTMEncDec-AD*) model to detect the running instability occurrences.

Furthermore, we deploy a *k-means* algorithm on latent space to identify clusters associated with root causes instigating instability. We deployed the *iVRIDA* framework on simulated and measured accelerations of European high-speed rail vehicles where *SAE-AD* and *LSTMEncDec-AD* models showed 97% accuracy. The proposed method contributes to smart maintenance by intelligently identifying anomalous vehicle-track interaction events.

1. Introduction

In Prognostics and Health Management (PHM), data-driven, physical model-based, and hybrid methods are the three main approaches for implementing diagnostic and prognostic solutions. The applicability of data-driven methods is primarily hindered by the unavailability of extensive training data which adequately covers the relevant state space. On the other hand, the applicability of physical model-based methods is not always suitable because the degradation process of the underlying technical system is highly complex and cannot be accurately described by first principles. The hybrid methods, which combine data-driven and model-based approaches, offer the possibility to mitigate the limitations of both previous methods partially. A hybrid model needs sufficiently detailed data-driven and physics-based models; however, development of both models is a cumbersome and expensive task. The

physics-informed machine learning is a transitional field between data-driven and hybrid approaches where the physical knowledge is integrated into a machine learning model. Physics-informed machine learning is a highly active research area and, in the literature, has been referred to as *Physics informed machine learning*, *Theory-guided data science*, *Physics-based learning*, and *Physics guided machine learning*. This paper proposes a physics-guided machine learning algorithm for intelligent vehicle running instability detection and responsible vehicle fault identification. In other words, we present rail vehicle dynamics guided unsupervised vehicle running instability detection algorithm (*iVRIDA*).

The self-excited oscillations of wheelsets, running gears, or the whole vehicle in lateral direction constitutes a phenomenon which is called running instability. Running instability associated with wheelset, bogie and whole vehicle are called wheelset running instability, bogie running instability and vehicle running instability, respectively. Since, very early

Abbreviations: AD, Anomaly Detection; ADR, Anomaly Detection Rate; AE, Autoencoder; CAF, Construcciones y Auxiliar de Ferrocarriles; CFA-I, Carbody Floor Accelerometer Above Leading Bogie; CFA-II, Carbody Floor Accelerometer Above Trailing Bogie; D1, Track Irregularity Wavelength Range; FRA, Federal Railroad Administration; **h, H**, Latent Space Representations; *iVRIDA*, Intelligent Vehicle Running Instability Detection Algorithm; KL, Kullback-Leibler; LSTM, Long Short Term Memory; *LSTMEncDec*, LSTM Encoder Decoder Network; *LSTMEncDec-AD*, LSTM Encoder Decoder Anomaly Detection; MAE, Mean Absolute Error; NN, Neural Network; PCA, Principal Component Analysis; PCA-AD, Principal Component Analysis - Anomaly Detection; PHM, Prognostics and Health Management; PIVOT II, Performance Improvement for Vehicles on Track II; PSD, Power Spectral Density Estimate; SAE, Sparse Autoencoder; *SAE-AD*, Sparse Autoencoder - Anomaly Detection; VFI, Vehicle Fault Identification; *VFIA*, Vehicle Fault Identification Accuracy; VRID, Vehicle Running Instability Detection; *W_z*, Sperling Index in lateral direction.

^{*} Corresponding author.

E-mail address: rohank@kth.se (R. Kulkarni).

<https://doi.org/10.1016/j.measurement.2023.112894>

Received 18 October 2022; Received in revised form 3 February 2023; Accepted 13 April 2023

Available online 17 April 2023

0263-2241/© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Table 1
Standards including vehicle running instability detection procedure [5,6].

| Standard | Acceleration signal | Filter | Alarm method |
|--|----------------------------------|--------------------|---|
| EN14363[7] / TSI-LOC & PAS/ UIC515 [8] | Lateral bogie frame acceleration | Band-pass | Hunting alarm if root-mean-square value > limit value on bogie mass |
| FRA[9] | Lateral bogie frame acceleration | 0.5–10 Hz bandpass | Hunting alarm if root-mean-square value (mean removed) for 2 s > 3.92 m/s ² |
| AAR | Lateral carbody acceleration | 15 Hz low-pass | Hunting alarm if peak-to-peak value > 14.71 m/s ² or standard deviation > 1.28 m/s ² |

days of railway engineering, a rail-vehicle/running gear oscillating at a limit cycle is termed as running instable. The reason being that the amplitude of limit cycle is usually larger than the amplitudes of track irregularities. Thus, the wheelset/running gear/ rail vehicle does not follow the track at all; and the vehicle starts oscillating over the track with constant amplitude i.e., limit cycle amplitude. Vehicle running instability is an important phenomenon in vehicle-track dynamic interaction and typically appears at a high vehicle speed and on a straight track or in large-radius curves. Running instability is an intrinsic behavior of a vehicle system that depends on the vehicle's health and track subsystems. The foremost reasons for running instability are poor vehicle yaw dampers, too soft primary suspension in the horizontal plane, or poor wheel-rail interface geometry. Vehicle instability is a safety concern and can cause passenger discomfort [1].

In Europe, high-speed trains (above 250 km/h) are mandatorily equipped with hunting alarm systems in accordance with TSI-LOC & PAS [2] and if an alarm is triggered, the train driver must reduce vehicle speed and reports the incident to the train operator and the infrastructure manager [3]. Various international standards, listed in Table 1, outline procedures for detecting hunting by analyzing the filtered lateral acceleration data and comparing it against established limit criteria. In these standards, the limit criteria are mainly aim to detect fully developed limit cycles and alert the driver before a dangerous situation arises [4]. Thus, the limit criteria are defined for certifying the vehicle's

stability during certification tests. Unfortunately, those criteria and corresponding methods are not ideal for condition monitoring objective. Hence, recent articles propose methodologies for detecting vehicle instability and identifying root causes in-service conditions.

The methods proposed by Bruni et al. [10], Forsberg et al. [11], Cavlo et al. [12], Bosso et al. [13], Kulkarni et al. [6], and Lebel et al. [14] are physical model-based methods. Bruni et al. [10] and Calvo et al. [12] employed Operational Modal Analysis (OMA) techniques to detect vehicle running instability from onboard accelerations. Forsberg et al. [11], Bosso et al. [13], Kulkarni et al. [6], and Lebel et al. [14] proposed spectral analysis-based methods for vehicle running instability detection. In recent years, Gasparetto et al. [15], Ning et al. [16], Zeng et al. [5], Ye et al. [17,18], Kulkarni et al. [19], Sun et al. [20,21] and Wang et al. [22] proposed purely data-driven methods for diagnosis of vehicle running instability. The methods proposed by Gasparetto et al. [15], Ning et al. [16], Ye et al. [17,23], Kulkarni et al. [19,24], Sun et al. [20,21] and Wang et al. [22] are typical data-driven methods where a certain statistical classifier is trained and tested based on features extracted from vehicle response. Thus, these data-driven methods are based on supervised classification learning, and in practice, many challenges arise when deploying, continuously training, and updating supervised classifiers on measurements. These challenges are:

- (1) Occurrences of vehicle running instability and root cause identification reports, which represent ground truth for supervised learning tasks, are usually not collected and reported systematically and consistently [3].
- (2) The vehicle-track interaction is a complex open system with many sources of variability; thus, there exists a significant difference between the training domain and the target domain. The pretrained supervised classification model does not perform well in these scenarios because of inherent bias [25].
- (3) The training and test datasets are sampled independently from static statistical distribution is the underlying assumption of most supervised classification models. This fundamental assumption ensures that model parameters identified in training fit the test datasets. However, the assumption usually does not hold for real-world datasets related to open complex systems such as vehicle-track interaction; since the operating conditions vary significantly over the service time [26].

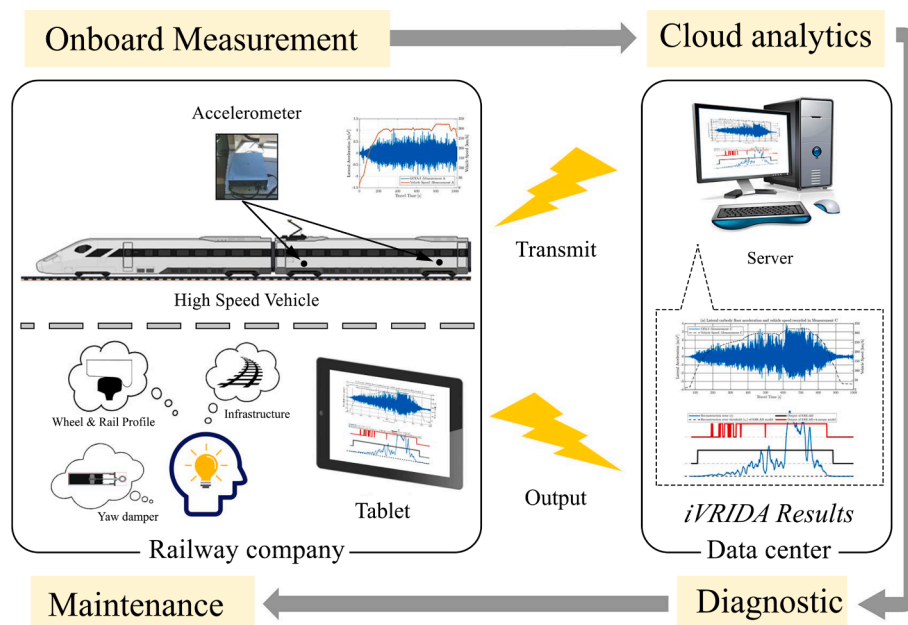


Fig. 1. Overview of deployment of the iVRIDA Framework for onboard monitoring [graphical illustration inspired from [45].

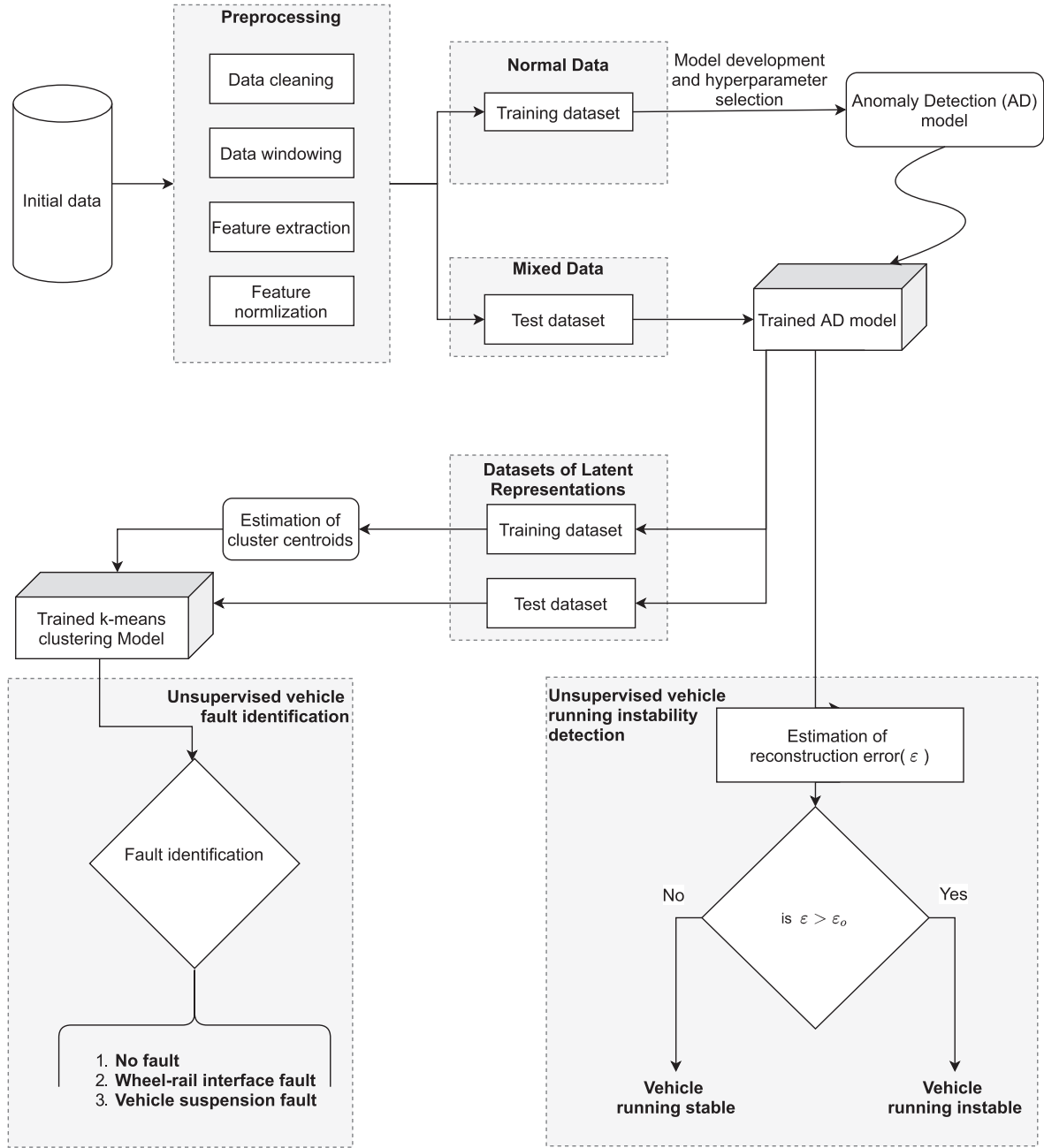


Fig. 2. The framework of the proposed iVRIDA algorithm.

- (4) In real-world sensor data from rail vehicles, there are scenarios when the behavior of a vehicle changes based on usage and external factors, which are difficult to capture. For example, a loaded vehicle behaves differently from an unloaded vehicle. Further, the relevant information on whether a vehicle is loaded or unloaded may not be available. The amount of payload on a vehicle at a time may be unknown or change frequently/abruptly as the vehicle travels on a route [27].
- (5) In recent years, the volume of collected data is rapidly grown; however, the data quality is not always satisfactory, which significantly affects the performance of pretrained supervise classification models [25].

All these challenges can be tackled if we learn partly or wholly unlabelled data, typically available in abundance. In the context of PHM of machines [28], unsupervised learning has successfully identified the

causes of system failures without supervision. One of PHM's most popular unsupervised learning approaches is signal reconstruction, also referred to as the residual-based approach. The general idea behind the signal reconstruction is to define a model that can learn the healthy system behavior and to distinguish it afterward from system states that are dissimilar to those observed under normal operating conditions. This technique is known as novelty or anomaly detection (AD) [29].

In this paper, we propose rail vehicle dynamics guided unsupervised vehicle running instability detection algorithm (iVRIDA, see Fig. 1). The proposed algorithm is unsupervised; thus, there is no need for labeled and balanced datasets. Here, we suggest unsupervised anomaly detection (AD) algorithms for detecting vehicle running instability. Then, unsupervised clustering of latent space to identify the vehicle fault instigating running instability. Moreover, we propose a generalistic AD-based vehicle instability detection framework, where any signal reconstruction-based AD algorithm and unsupervised clustering

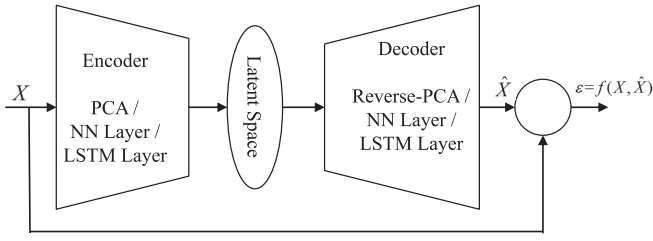


Fig. 3. A schematic of the reconstruction-based AD model [34].

algorithm can be integrated. In the detailed investigation, we compare the performance of three AD algorithms, namely: Principal Component Analysis (PCA-AD; Baseline model), Sparse Autoencoder (SAE-AD), and LSTM Encoder Decoder (LSTMEncDec-AD) within the *iVRIDA* framework to detect the occurrence of vehicle running instability.

Moreover, we show that an unsupervised *k-means* clustering algorithm can identify clusters and associate clusters with respective root causes of observed vehicle dynamic instability. The later step of the *iVRIDA* framework is called unsupervised vehicle fault detection. In this investigation, we show the superiority of the SAE-AD + *k-means* variant of *iVRIDA* in detecting the occurrence of vehicle dynamic instability and identification of corresponding vehicle fault on vehicle accelerations measured on a typical high-speed rail vehicle operating in Europe.

Following this introductory Section, the rest paper is structured as follows. In Section 2, the *iVRIDA* framework is explained, where the theoretical background of AD and clustering models is briefly described. In Section 3, the application of the *iVRIDA* on simulated and measured vehicle accelerations is described. The section documents the simulation, the acquisition and analysis of onboard measurements, followed by the articulation of the machine learning problem. Section 4 presents the results of unsupervised vehicle running instability detection and unsupervised vehicle fault identification instigating dynamic instability. In Section 5, an offline demonstration of the *iVRIDA* framework on measured accelerations is described. Finally, our conclusions are summarized in Section 6.

2. The *iVRIDA* framework

The emergence of running instability of rail vehicles during regular operation is a typical example of anomalous dynamic behavior. This abnormal behavior is reflected in accelerations (vibrations) in the car-body. The AD-based framework for vehicle running instability detection and the root cause identification is illustrated in Fig. 2. The vehicle accelerations measured with one or more accelerometers are stored in the database. This initial data is pre-processed according to a four-step scheme: data cleaning, data windowing, feature extraction, and feature normalization. The dataset of the regular operation of the vehicle forms a training dataset used for training the multiple AD models. The AD models are trained on the training dataset and are optimized for reconstructing the input time series by minimizing the reconstruction error (ϵ). Finally, the maximum reconstruction error (ϵ) observed on the training dataset is defined as the reconstruction error threshold (ϵ_0) of a particular AD model.

The AD model is employed on the test dataset to detect anomalies, if present. During this phase, the trained AD model has two outputs: latent space representation vector (h) of the input time series and reconstruction error (ϵ). The later output of the AD model is compared against the reconstruction error threshold (ϵ_0) to identify the anomaly in vehicle accelerations. This branch of the *iVRIDA* algorithm is called an Unsupervised Vehicle Running Instability Detection (VRID). The AD model's second output is the input sequence's latent space representation. The dataset composed of all observation's latent space representations is a matrix H ; which is later, grouped into clusters with the *k-means* algorithm, where each cluster is associated with a respective vehicle fault

that initiates vehicle running instability. This branch of *iVRIDA* is unsupervised vehicle fault identification which intelligently identifies vehicle fault (VFI) instigating running instability.

2.1. Anomaly detection (AD) models

The anomalies are classified into point anomaly, pattern anomaly, and contextual anomaly, and the vehicle running instability occurrence is a combination of pattern and contextual anomalies. In the past two decades, many AD models have been proposed, which are categorized into *distance-based*, *ensemble-based*, *statistical*, *domain-based*, and *reconstruction-based methods*. Reconstruction-based methods are often used in safety-critical applications [30] (e.g., vehicle running instability occurrence) for regression or classification purposes. These methods can autonomously model the underlying data, and when test data are presented to the AD model, the presence of anomaly is detected after comparing reconstruction error (ϵ) against the reconstruction error threshold (ϵ_0). Reconstruction-based methods are classified into subspace-based methods and neural network methods. In the present investigation, the subspace-based method: PCA [31] and the neural network-based method: Sparse Autoencoder (SAE) [32], and the Long Short Term Memory (LSTM) [33] method are compared against each other for their performance in VRID and VFI branches of *iVRIDA*. The fundamental schematic of reconstruction-based AD models is shown in Fig. 3.

Typically, reconstruction-based AD models are trained to reconstruct instances of healthy machine behavior, with the target being the input itself [34]. During training, the model parameters are optimized to minimize the mean absolute error (MAE) between the input and reconstructed sequence, which is termed as reconstruction error (ϵ) given by:

$$\epsilon = \frac{\sum_{i=1}^n |\hat{x}_i - x_i|}{n} \quad (1)$$

where, \hat{x} and x are reconstructed and original sequences, respectively. i is element index,

Then, the reconstruction error (ϵ) of any test observation is used for computing the likelihood of anomaly in the test sequence. Thus, reconstruction-based models are trained only on normal or healthy data, and when the pretrained model encounters an anomalous sequence, it will not be able to reconstruct it well and hence would result in higher reconstruction error (ϵ) compared to those of a normal sequence. This is particularly useful in scenarios when anomalous data is not available or is sparse, making it difficult to learn a classification model over normal and anomalous sequences. This is especially true for rail vehicles and railway tracks that undergo periodic maintenance and therefore get serviced before anomalies/faults show up in the sensor readings.

2.1.1. PCA-AD (Baseline Model)

The PCA-AD model is based on a simple linear method called Principal Component Analysis (PCA). PCA finds the linear projection that best preserves the variance measured in the input space [35]. The linear projections are created by identifying the dataset's eigenvectors and sorting them by their eigenvalues. The dataset is projected along the first k eigenvectors to reduce the number of dimensions. This k dimensional space is the latent space of the PCA-AD model. The data is reconstructed to the original space by performing reverse PCA to obtain reconstructed input. The reconstruction error (ϵ) between the original and reconstructed input is calculated with Equation (1). Although PCA is both easy to use and efficient, its effectiveness is limited when data is not linearly correlated. PCA-AD model is chosen as a baseline model to have a fair comparison with other AD models.

2.1.2. SAE-AD model

Autoencoder (AE) [36] is an unsupervised neural network that is

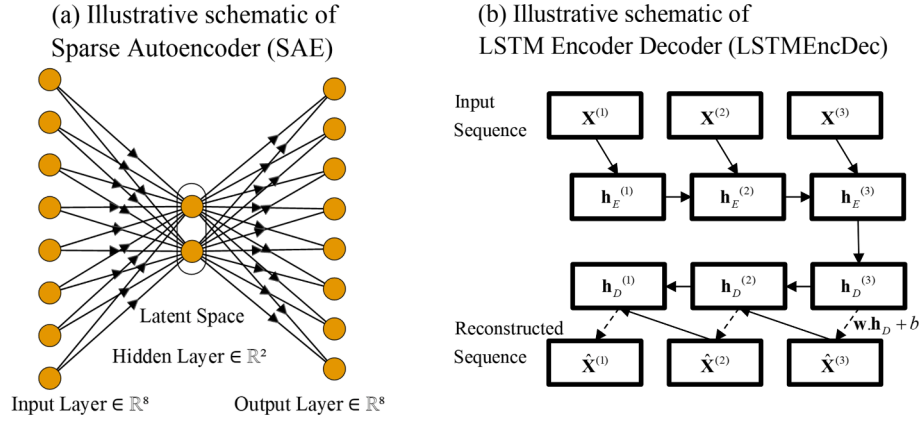


Fig. 4. Illustrative schematics of SAE and LSTMEncDec networks.

trained to reconstruct the input X on the output layer \hat{X} in a two-phase process: *encoding* learns the hidden/latent space representation (h) of the data X via a feature-extracting function and *decoding* maps h back into the input space to obtain a reconstruction of the data X . The most critical drawback of AE is the tendency to (learn) identity functions without extracting meaningful information about the input data. The Sparse Autoencoder (SAE) was proposed to overcome the drawbacks of AE; see Fig. 4 (a). SAE exploits the inner structure of the input data by including a sparsity constraint on the activations of the hidden units by adding the Kullback-Leibler (KL) divergence term to the cost function [32]. The sparsity of the hidden layer neuron is that neuron's average output activation value. The average output activation of a neuron i is defined as:

$$\hat{\rho}_i = \frac{1}{n} \sum_{j=1}^n z_i^{(1)}(x_j) = \frac{1}{n} \sum_{j=1}^n h(w_i^{(1)T} x_j + b_i^{(1)}) \quad (2)$$

where, n is the number of samples in the training dataset, x_j is the j^{th} training sample, $w_i^{(1)T}$ is the i^{th} row of the weight matrix $W^{(1)}$, and $b_i^{(1)}$ is the i^{th} entry of the bias vector $b^{(1)}$. A higher $\hat{\rho}_i$ means the neuron is responsive to most training samples, whereas a lower $\hat{\rho}_i$ means the neuron is responsive to only a few training samples. Thus, inserting a term into a cost function that constrains the values of $\hat{\rho}_i$ to lower values introduces sparsity into AE, where each hidden layer reacts to a small number of training samples. The Sparsity regularizer [37] added to the cost function through the Kullback-Leibler (KL) divergence term is defined as:

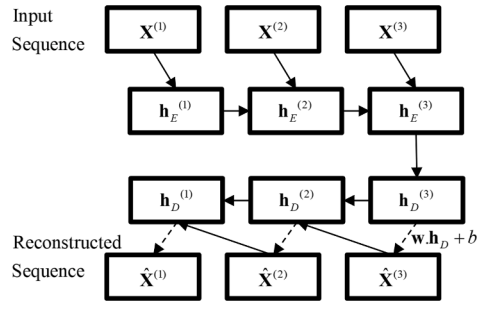
$$\Omega_{\text{sparsity}} = \sum_{i=1}^{D^{(1)}} KL(\rho || \hat{\rho}_i) = \sum_{i=1}^{D^{(1)}} \rho \log\left(\frac{\rho}{\hat{\rho}_i}\right) + (1 - \rho) \log\left(\frac{1 - \rho}{1 - \hat{\rho}_i}\right), \quad (3)$$

The KL divergence function measures similarity between ρ and $\hat{\rho}_i$ distributions; its value is zero when the distributions are the same and becomes larger as these diverge from each other. Adding this term to the cost function, which is minimized during the training procedure, forces distributions ρ and $\hat{\rho}_i$ to be close to each other. During the SAE training Ω_{sparsity} may become small by increasing values of the weights $w^{(1)}$ and reducing values of $z^{(1)}$. Thus, to prevent it from happening; an L_2 regularization term is added to the cost function and is defined as:

$$\Omega_{\text{weights}} = \frac{1}{2} \sum_{l=1}^L \sum_{j=1}^{n_l} \sum_{i=1}^{k_l} (w_{ji}^{(l)})^2, \quad (4)$$

where L is the number of hidden layers, whereas n_l and k_l are the output and input sizes of layer l . Finally, the cost function of SAE is an adjusted mean squared error function which is given as:

(b) Illustrative schematic of LSTM Encoder Decoder (LSTMEncDec)



$$E = \frac{1}{N} \sum_{n=1}^N \sum_{k=1}^K (x_{kn} - \hat{x}_{kn})^2 + \lambda^* \Omega_{\text{weights}} + \beta^* \Omega_{\text{sparsity}}, \quad (5)$$

where λ and β are coefficients of the L_2 regularization term and the sparsity regularization terms, respectively.

2.1.3. LSTMEncDec-AD model

LSTM networks [33] are recurrent neural network models that have been a very popular choice for various sequence learning tasks like handwriting recognition, speech recognition, and time-series analysis. Recently an LSTM-based Encoder-Decoder (LSTMEncDec-AD) model has been proposed for AD in multi-sensor time-series [27]. In LSTMEncDec architecture, both the encoder and decoder are LSTM networks. The encoder network compresses the input to latent space, which is then decompressed with the decoder network into the original space. Consider a sequential data $X = \{x^{(1)}, x^{(2)}, \dots, x^{(L)}\}$ of length L , where each point $x^{(i)} \in R^m$ is a m dimensional vector for m variables. Given X , $h_E^{(i)}$ is the hidden state of the encoder for each $i \in \{1, 2, \dots, L\}$, where $h_E^{(i)} \in R^c$, c is the number of LSTM units in the hidden layer of the encoder. The encoder and decoder are jointly trained to reconstruct the input X in reverse order [27], i.e., the output is $\{x^{(L)}, x^{(L-1)}, \dots, x^{(1)}\}$. The final hidden state $h_E^{(L)}$ of the encoder is input for the decoder. A linear layer on top of the decoder layer is used to predict the target. During training, the encoder uses $x^{(i)}$ as an input to obtain the hidden state $h_D^{(i-1)}$, then predicts the corresponding target $x^{(i-1)}$. During inference, the predicted value $\hat{x}^{(i)}$ is the input to the decoder to obtain $h_D^{(i-1)}$ and predicts $\hat{x}^{(i-1)}$; see Fig. 4 (b). The loss function of the LSTM-EncDec mode is as follows:

$$E = \sum_{X \in S_N} \sum_{i=1}^L \|x^{(i)} - \hat{x}^{(i)}\|^2, \quad (6)$$

where S_N is a set of normal training sequences.

2.2. Unsupervised clustering with k -means

Clustering is defined as identifying subgroups in the data such that observations in the same subgroup (cluster) are (very) similar while the observations in other clusters are very different. In clustering, the primary objective is to investigate the dataset's structure by forming subgroups. Thus, there is no need to incorporate ground truth in the training phase of the clustering algorithms, and these are unsupervised learning methods. The most common clustering method is k -means [38] because of its simplicity.

Table 2
Some parameters of X2000 vehicle model [39].

| Vehicle | Component | Mass (kg) | Mass moment of inertia | | |
|---------|-----------|-----------|------------------------------|------------------------------|------------------------------|
| | | | J_{xx} (kgm ²) | J_{yy} (kgm ²) | J_{zz} (kgm ²) |
| X2000 | Carbody | 32,000 | 64,446 | 1,519,965 | 1,519,965 |
| | Bogie | 5500 | 3000 | 6000 | 8000 |
| | Wheelset | 1300 | 750 | 250 | 750 |

2.2.1. The *k*-means clustering algorithm

The *k*-means algorithm is an iterative algorithm that partitions the input dataset into *k* predefined distinct non-overlapping subgroups where each observation is associated with only one cluster. The algorithm tries to make the intra-cluster data points as similar as possible while keeping the other clusters as different and far away as possible. The algorithm assigns datapoints to a cluster such that the sum of the squared distance between the observations and the cluster's centroid

(arithmetic mean of all the data points belonging to the cluster) is minimized but maximizes the distance between two centroids. Thus, the *k*-means algorithm solves the Expectation-Maximization problem [38]. The Expectation step assigns the data points to the closest cluster, and the Maximization step computes the centroid of each cluster. The objective function is.

$$J = \sum_{i=1}^m \sum_{k=1}^K w_{ik} \|x^i - \mu_k\|^2 \quad (7)$$

where $w_{ik} = 1$ for datapoint x_i if it belongs to the cluster k ; otherwise $w_{ik} = 0$. μ_k is the centroid of s cluster.

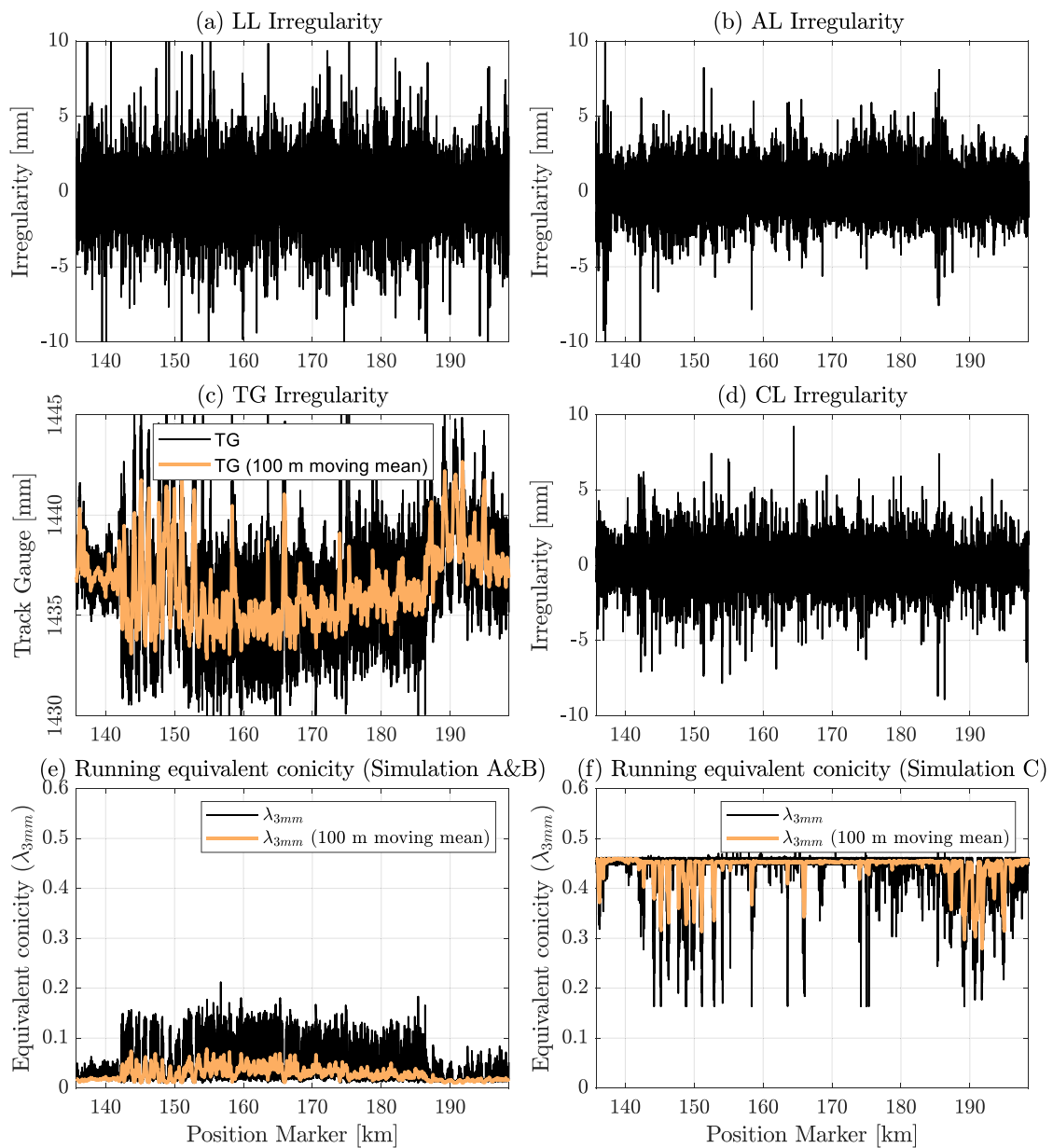


Fig. 5. Spatial distributions of the four types of track irregularities (subfigures a-d). The running equivalent conicity (λ_{3mm}) of nominal wheel and worn wheel profile with nominal rail, respectively, in subfigures e-f.

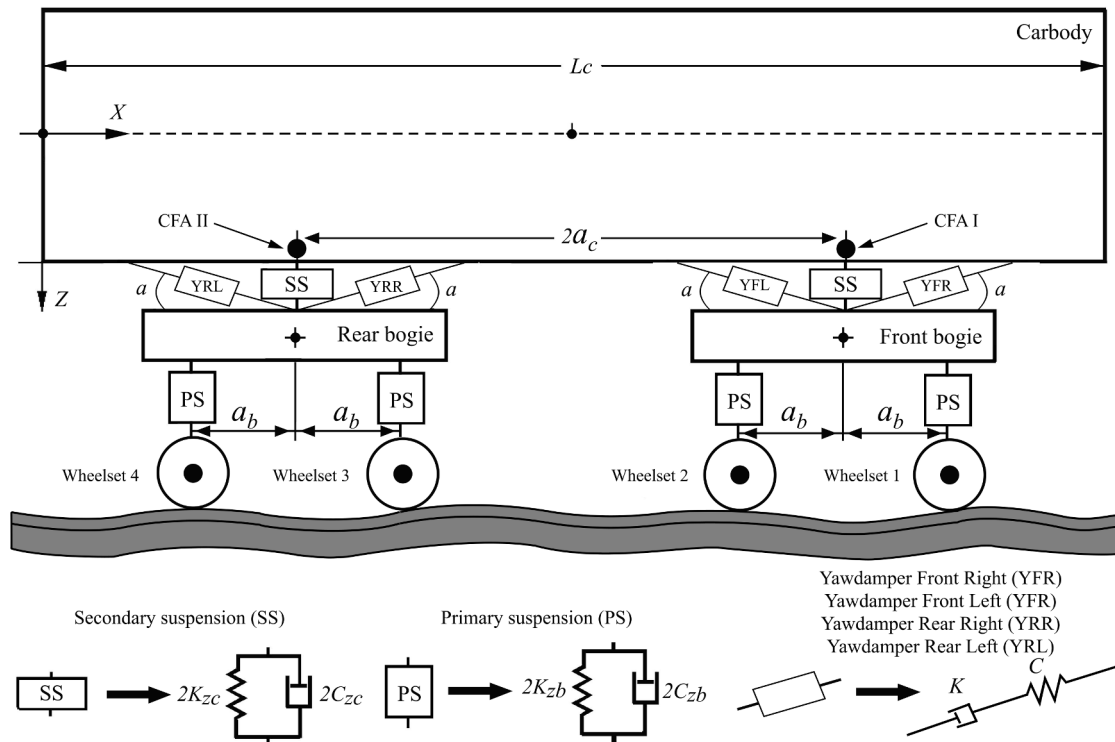


Fig. 6. Schematic of vehicle model and sensor architecture of the iVRIDA Scheme.

3. Application of iVRIDA

3.1. Case studies

3.1.1. Simulated acceleration-based case study (CS-I)

The Swedish train operator SJ operates the high-speed train X2000 on the Swedish rail network. Most SJ X2000 trains consist of a power car, five intermediate vehicles, and a driving trailer and operate at a top speed of 200 km/h. An intermediate vehicle is studied here [39] through vehicle dynamic simulations.

3.1.1.1. MBS model of vehicle. The vehicle model consists of a carbody, two bogie frames, and four wheelsets modeled as 6 DOF rigid bodies connected by primary and secondary suspension elements. The primary and secondary suspension consists of spring and viscous damper elements in the x, y, and z-directions. Since the X2000 vehicle is specifically designed to run in curves with high cant deficiency, the primary suspension is relatively soft to give the wheelsets improved radial self-steering capabilities. The X2000 vehicle model is also equipped with four yaw dampers, i.e., two per bogie, which works in a longitudinal direction. The vehicle data is mainly confidential, but some information is shown in Table 2, and the static axle loads are 118.1 kN for all wheelsets.

3.1.1.2. Track layout and track irregularity. In the present investigation, the measured irregularities on the Swedish track section 416 (BDL416) are used for exciting the vehicle stochastically. The track section is selected because of the availability of measured rail profiles and track irregularity data. The Longitudinal Level (LL), Alignment Level (AL), Track Gauge (TG), and Cross Level (CL) irregularities are used in the simulation, and the spatial distributions of the track irregularities are shown in Fig. 5 (subfigures a-d) with black color. The quality of the track irregularities of the shortlisted section is evaluated according to the methodology and limits defined in EN 13848-5 [40]. The track section does not have any isolated track irregularity defects. LL irregularities are mainly in Class B and Class C. AL irregularities are equally distributed in

Classes A, B, and C, respectively. CL irregularities of the selected track section are slightly poor where most of the track is in Class D, with less than 5% of the track section in Class E. In subfigure (c), black and yellow lines show the TG and 100 m moving mean of TG. The BDL416 has a tight gauge for several kilometers between the 144 km and 185 km position markers (subfigure c).

The running equivalent conicity is calculated using the two equivalent conicity functions. These two functions are obtained by combining nominal wheel profile S1002t31.5 [48] mm and worn wheel profile (WP-C) [6] with 60E1i30 [47] rail profile, respectively. In the calculation, it is assumed that the rail profiles are constant along the track, and only the variation in the track gauge is considered. The running equivalent conicity at 3 mm (λ_{3mm}) of these two combinations of wheel and rail profiles along the track is shown in Fig. 5 (subfigure e-f). In both subfigures, λ_{3mm} and 100 m moving average of λ_{3mm} are shown in black and yellow lines, respectively. The λ_{3mm} of the first combination is very low in the beginning and end of the BDL416; even though there is a slight increase in λ_{3mm} in between 144 km and 185 km markers, the value of 100 m moving average λ_{3mm} is always below 0.1. On the contrary, the λ_{3mm} value of the second combination is significantly high along the track section; moreover, the λ_{3mm} value is consistently 0.45 between the 144 km and 185 km markers.

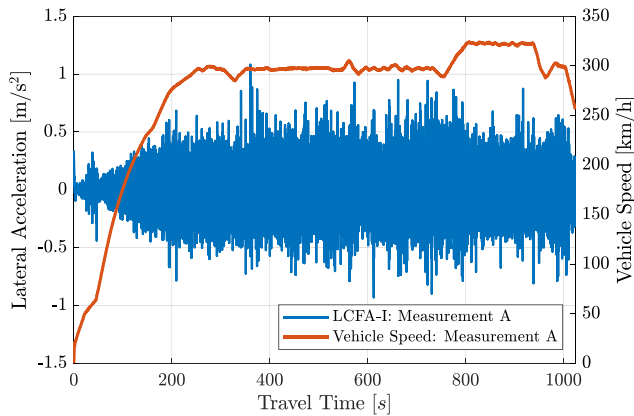
3.1.2. Measured acceleration-based case study (CS-II)

This case study is based on measured onboard accelerations acquired on a vehicle of a high-speed train family, which typically operates at 300 km/h on track with 1435 mm nominal track gauge are studied. The vehicles (see Fig. 6 (a)) of this family are equipped with many sensors which measure and send data to cloud storage (Fig. 1). The stored data is usually used for the deployment of the latest and most innovative maintenance and optimization technologies to maximize the trains' quality of service and reduce life cycle costs by implementing predictive and condition-based maintenance strategies. In the present investigation, vehicle accelerations measured on an intermediate vehicle is studied. The vehicle consists of a carbody, two bogie frames, and four wheelsets connected by primary and secondary suspension elements.

Table 3

Parameters of simulations Sim A, Sim B, and Sim C.

| Simulation ID | Wheel Profile | 100 m Running Equivalent Conicity | Yaw Damper condition | Wheel condition |
|---------------|-------------------|-----------------------------------|----------------------|-----------------|
| Sim A | S1002t31.5 | less than 0.1 (Fig. 5 (e)) | No fault | No fault |
| Sim B | S1002t31.5 | less than 0.1 (Fig. 5 (e)) | Front left failed | No fault |
| Sim C | Worn wheel (WP-C) | ~ 0.5 (Fig. 5 (f)) | No fault | Worn wheel |

**Fig. 7.** Vehicle speed and lateral acceleration recorded at CFA-I during Measurement A.

The primary and secondary suspension has springs and viscous dampers in the vertical (Z) and lateral (Y) directions. The vehicle is also equipped with four yaw dampers, i.e., two per bogie, which acts in the longitudinal (X) direction. These hydraulic dampers are designed to prevent running instability of the vehicle even at extreme service conditions. However, the vehicle may occasionally run unstably if one or more yaw dampers are malfunctional.

3.1.3. Vehicle acceleration database

The vehicle accelerations used in this research are measured with two bi-directional accelerometers mounted on the carbody floor above the bogie pivot points of the leading and trailing bogie, respectively. These two accelerometers are designated as CFA-I and CFA-II, respectively. CFA-I and CFA-II measure lateral and vertical accelerations and store them locally for post-processing. A speed meter accompanying accelerometers on the vehicle and the sensor architecture, along with travel direction, is shown in Fig. 6.

3.1.3.1. Simulated acceleration database. This database consists of the results of three simulation cases, namely Sim A, Sim B, and Sim C. These three simulations based on the vehicle model and track irregularities described in the earlier section, whereas, the wheel-rail interface friction coefficient is selected as 0.6 to simulate extremely dry friction conditions. The additional simulation parameters are summarised in Table 3, where Sim A is a healthy vehicle running on BDL416. Sim B and Sim C are faulty vehicles where the faults are failed yaw damper and worn wheel profile, respectively.

3.1.3.2. Measured acceleration database. The onboard accelerations are measured on a vehicle during research trials on the same European high-speed network. The data is obtained in three measurement campaigns, and three campaigns are designated as Measurement A, Measurement B, and Measurement C, respectively. Moreover, the vehicle during

Measurement A was completely healthy whereas during Measurement B and Measurement C vehicle's one yaw damper was failed. The vehicle speed and lateral acceleration at CFA-I recorded during measurement A are shown in Fig. 7. There, the time is on the horizontal axis; lateral carbody floor acceleration is on the left vertical axis, and vehicle speed is on the right vertical axis. The lateral carbody floor acceleration is shown in blue, whereas vehicle speed is shown in a dashed black line. It can be observed from the figure that the vehicle is accelerating from 0 to 330 km/h. The lateral carbody floor acceleration amplitude gradually increases along with the vehicle speed and rises upto 1 m/s^2 . The total measurement duration is 20 min.

3.2. Analysis of vehicle accelerations

The vehicle accelerations acquired in Sim A-C and Measurements A-C are compared against each other using methods typically used in the railway industry. The objective of analyzing vehicle accelerations using domain-specific methods is to discover the dynamic status of the vehicles. This investigation uses fundamental spectral techniques such as Power Spectral Density (PSD) estimation and ride quality index [41]. The lateral accelerations acquired at CFA-I in Sim A-C and Measurement A-C are compared against each other in the frequency domain in Fig. 8 (a) and (c), respectively, where horizontal and vertical axes are Frequency and PSD, respectively. The lateral ride quality [41] of Sim A-C and Measurement A-C is compared against each other in Fig. 8 (b) and (d), respectively, where horizontal and vertical axes are Time and Sperling index, respectively. In subfigures (a, b) the black, green, and dashed-violet lines correspond to Sim A, B, and C, whereas in subfigures (c & d) those three lines correspond to Measurement A, B, and C, respectively. The PSDs of Sim A and Measurement A don't show any dominant peak, whereas Sim B, Sim C, Measurement B, and Measurement C show extreme peaks at 4.8 Hz, 6.1 Hz, 2.2 Hz, and 2.2 Hz, respectively. Thus, the frequency representation of Sim B & C is very different from that of Sim A; similarly, Measurement B & C is very different from that of Measurement A. In Fig. 8 (b) and (d), the sperling index of Sim A and Measurement A oscillates around 2, which means that the vehicles have very comfortable rides. On the contrary, the ride quality during Sim B, Sim C, and Measurements B and C is very poor and sometimes touch the sperling index 4.

Table 4 summarizes the dynamic status of the vehicles in Sim A-C (Table 4 (a)) and Measurements A-C (Table 4 (b)). The respective vehicles are running dynamically stable in Sim A and Measurement A, whereas the same respective vehicles are running dynamically unstable in Sim B & C and Measurement B & C. The vehicle instability in Sim B & C is bogie + carbody instability with 4.8 Hz and 6.1 Hz, respectively, whereas in Measurements B and C, it is carbody instability, and the instability frequency is 2.4 Hz.

3.3. Formulation of machine learning problem

3.3.1. Pre-processing and dataset formation

First, the lateral accelerations measured at CFA-I and CFA-II are down-sampled from a 1000 Hz sampling rate to 200 Hz because the most interesting vehicle dynamic behavior is observed below 100 Hz. The vehicle running instability is typically observed at high speed; thus, the time sequences when the vehicle speed is less than 200 km/h are removed. The remaining time series of down-sampled lateral accelerations is divided into sequences of 16 s with 15 s of overlap between the sequences. Each sequence is scaled by the mean and standard deviation of the healthy data of the respective case studies i.e., Sim A and Measurement A. This results in three datasets for each case-study (CS-I & CS-II) containing two one-dimensional time series of length 3200, each labelled by respective labels.

Moreover, these six datasets are transformed with a feature extraction method. Here, the often applied power spectral density (PSD) method is applied separately on lateral acceleration measured at CFA-I

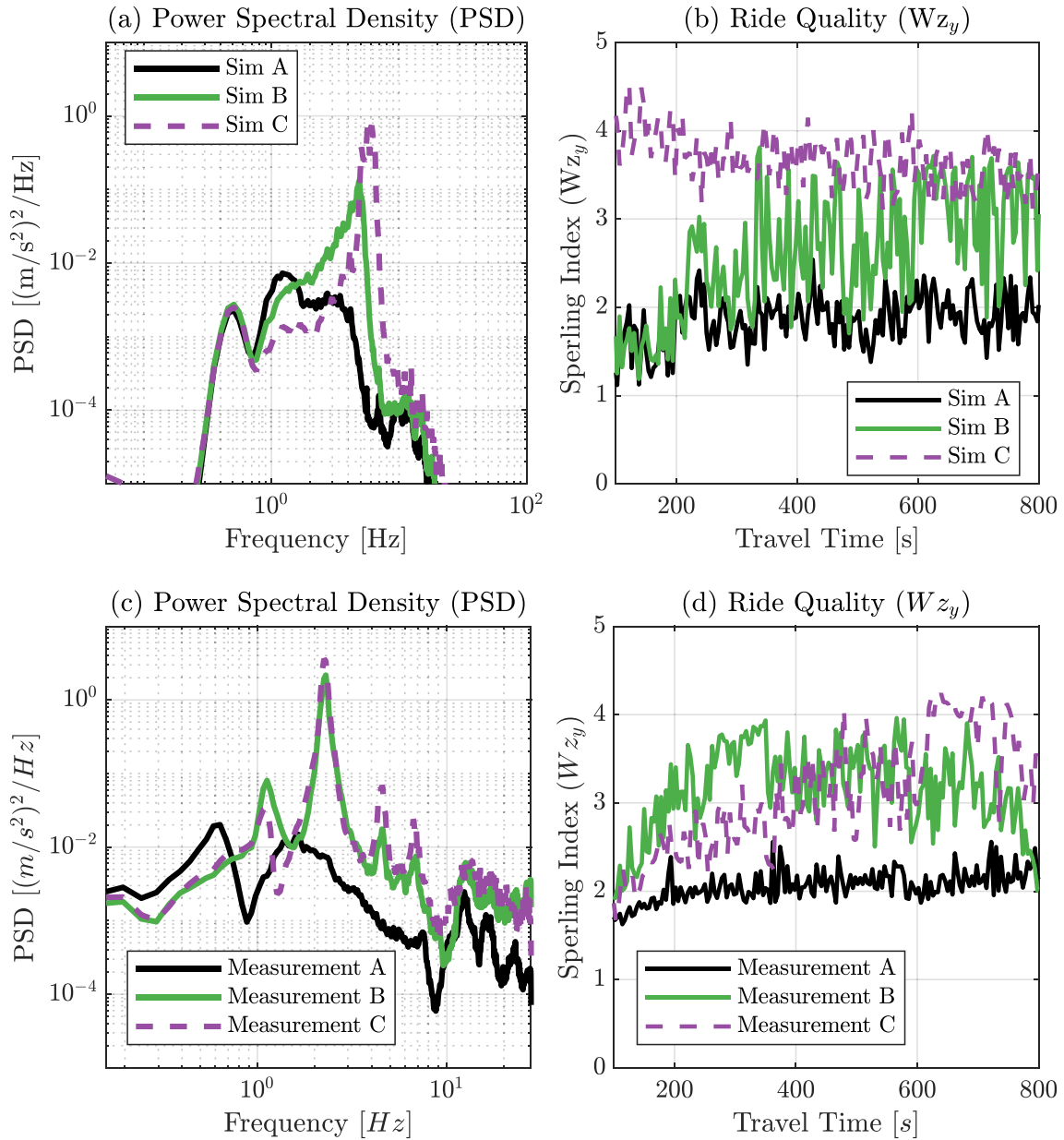


Fig. 8. PSD and the ride quality assessment of lateral acceleration evaluated at CFA-I for Sim A - C and Measurement A - C, respectively.

Table 4

Summary of vehicle acceleration database.

| Table 4 (a) | | Table 4 (b) | |
|---------------|-------------------------------|-------------------------|-------------------------------|
| Simulation ID | Dynamic Status of the vehicle | Measurement campaign ID | Dynamic status of the vehicle |
| Sim A | Stable | Measurement A | Stable |
| Sim B | Unstable | Measurement B | Instable |
| Sim C | Unstable | Measurement C | Instable |

and CFA-II to extract frequency domain features. The PSD features are calculated based on the previously extracted time series dataset, whereby the vector of PSD amplitudes between 1 and 20 Hz is considered as the input feature vector for all AD models. These datasets are designated as Sim-A, Sim-B, Sim-C, Measurement-A, Measurement-B, and Measurement-C, with datasets Sim-A and Measurement-A corresponding to the stable dynamic behavior of the respective vehicles and the other four containing anomalies, i.e., occurrences of running

instability.

3.3.2. Unsupervised vehicle running instability detection with AD models

In CS-I, AD models are trained with dataset Sim-A and tested with mixed datasets of Sim-A, Sim-B, and Sim-C. Similarly, in CS-II, AD models are trained with dataset Measurement-A and tested with a mixed dataset of Measurement-A and Measurement-B. All AD models are independently trained till 2000 epochs. After completion of the training procedure, the reconstruction error (ϵ) on the respective training dataset is calculated between the original and reconstructed samples. The 110% of the maximum reconstruction error (ϵ) of the respective training dataset is designated as the reconstruction error threshold (ϵ_0) for the respective AD model. This threshold is used for detecting anomalies in the respective test datasets, i.e., vehicle running instability. The respective hyperparameters of each AD model are optimized for the minimum ϵ_0 , highest Anomaly Detection Rate (ADR) in the test dataset and the smallest size of the latent space.

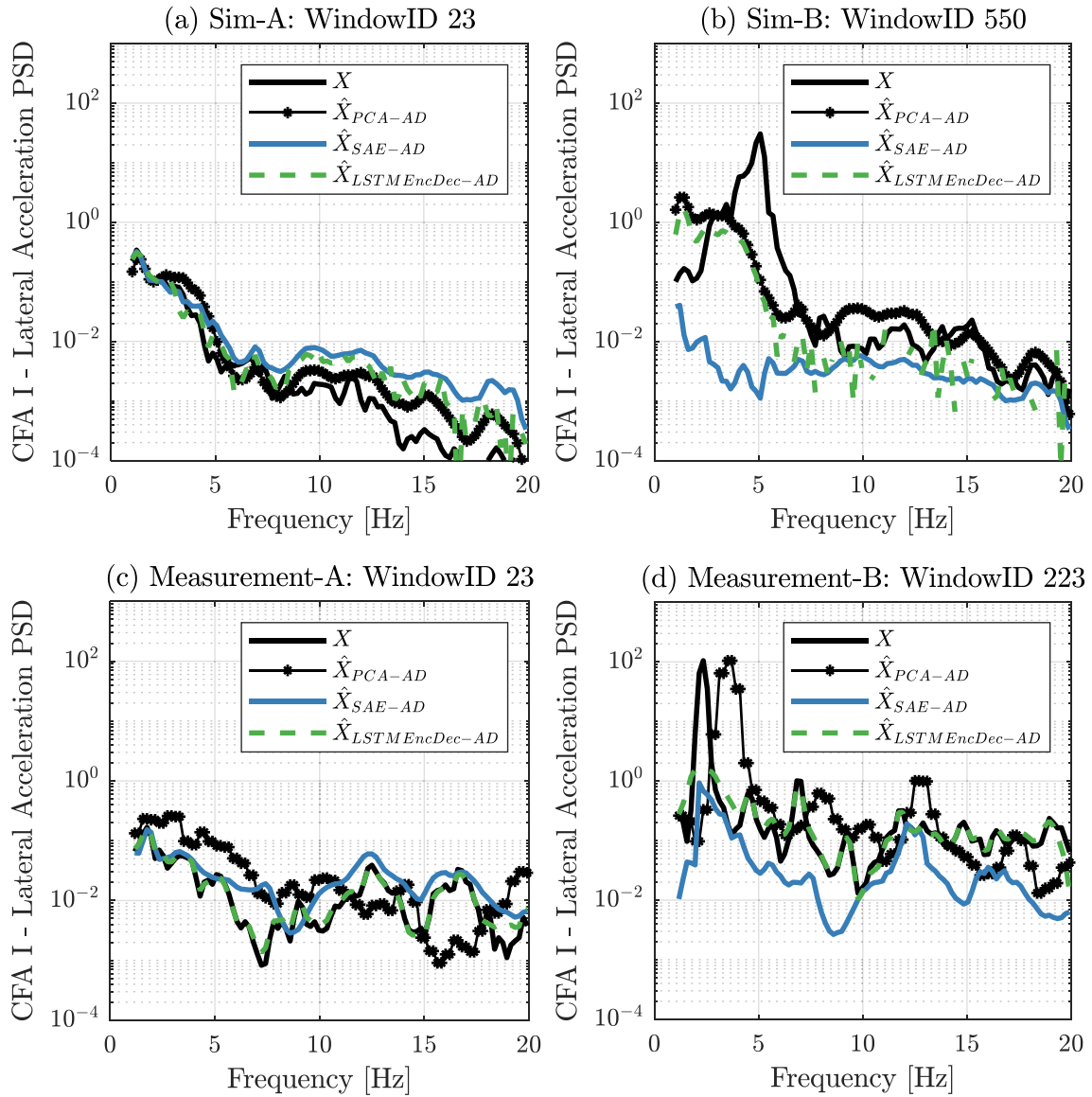


Fig. 9. Prevalidation of the three AD models. (a-d) comparison of the original input sequence (X) with reconstructed sequences (\hat{X}_{PCA-AD} , \hat{X}_{SAE-AD} , $\hat{X}_{LSTMEncDec-AD}$) on the randomly chosen window of Sim-A, Sim-B, Measurement-A, and Measurement B, respectively.

3.3.3. Unsupervised vehicle fault identification instigating vehicle instability

Since discovering novel fault types requires unsupervised evaluation of the latent feature space, a clustering approach was applied to the learned or extracted features. The latent features database (\mathbf{H}) of the AD models is transformed into the principal component space using PCA with an explained variance of 95%. Then, the k -means clustering algorithm is used to identify clusters in the database (\mathbf{H}) as the clustering algorithm assigns each sample to the cluster with the nearest mean [42].

4. Results of the case studies

4.1. Unsupervised vehicle running instability detection

4.1.1. Prevalidation

Firstly, we verify the three trained AD models of respective case studies by looking at their reconstruction performance on two sequences. We randomly chose one sequence measured at CFA-I from Sim-A, Sim-B, Measurement-A, and Measurement-B each. The reconstructed sequences (\hat{X}_{PCA-AD} , \hat{X}_{SAE-AD} , $\hat{X}_{LSTMEncDec-AD}$) are compared with the respective original sequences X in Fig. 9, where subfigures (a), (b), (c),

and (d) correspond to randomly chosen observations from datasets Sim-A, Sim-B, Measurement-A, and Measurement-B, respectively. In subfigures, horizontal and vertical axes are frequency and PSD estimates, respectively; also, the sequences X , \hat{X}_{PCA-AD} , \hat{X}_{SAE-AD} , and $\hat{X}_{LSTMEncDec-AD}$ are shown by black, star-black, blue, and dashed-green lines, respectively.

In subfigures (a) and (c), we can see that the reconstructed sequences of three AD models have good agreement with the original sequences of respective healthy datasets. Moreover, it is fascinating to point out that $\hat{X}_{LSTMEncDec-AD}$ in subfigure (c) almost coincides with the original sequence X except for divergence at a few frequencies. In subfigures (a) and (c), the simple baseline model PCA-AD shows maximum reconstruction error (as expected), and the neural network-based SAE-AD and LSTMEncDec-AD models show much smaller reconstruction errors. The ϵ of PCA-AD, SAE-AD and LSTMEncDec-AD for Sim-A (WindowID 23) is 0.09, 0.05 and 0.04 whereas for Measurement-A (WindowID 23) is 0.04, 0.01, and 0.002 respectively.

In subfigures (b) and (d), we see that the sequences reconstructed by three AD models have, as expected, inferior agreement with the original sequence of anomalous (faulty) respective datasets. In subfigure (d), it is

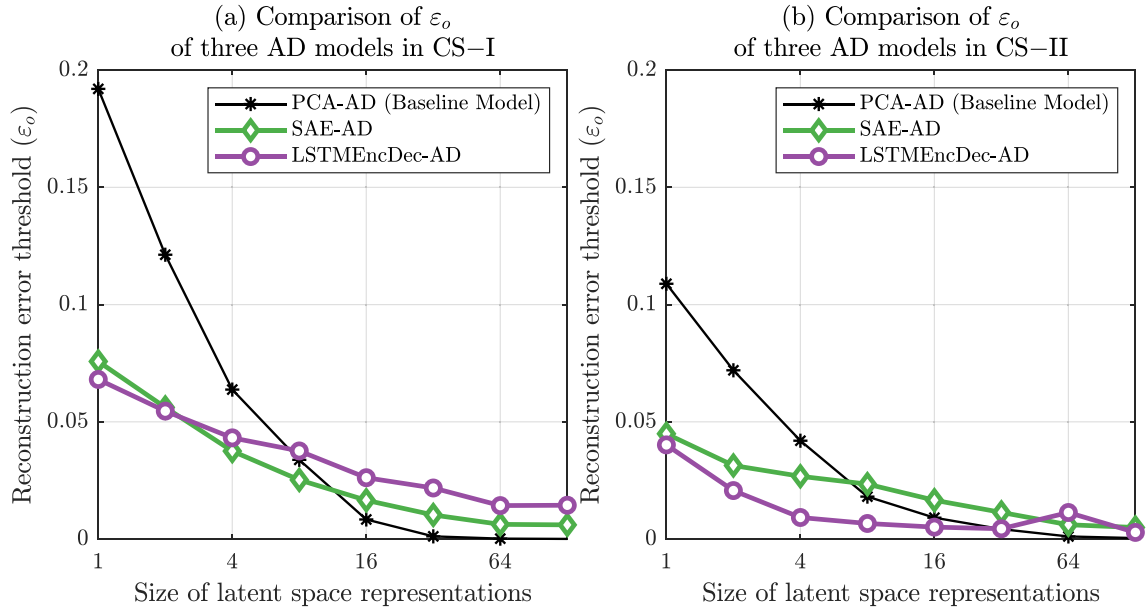


Fig. 10. Comparison of reconstruction error threshold (ε_0) of AD models in CS-I and CS-II.

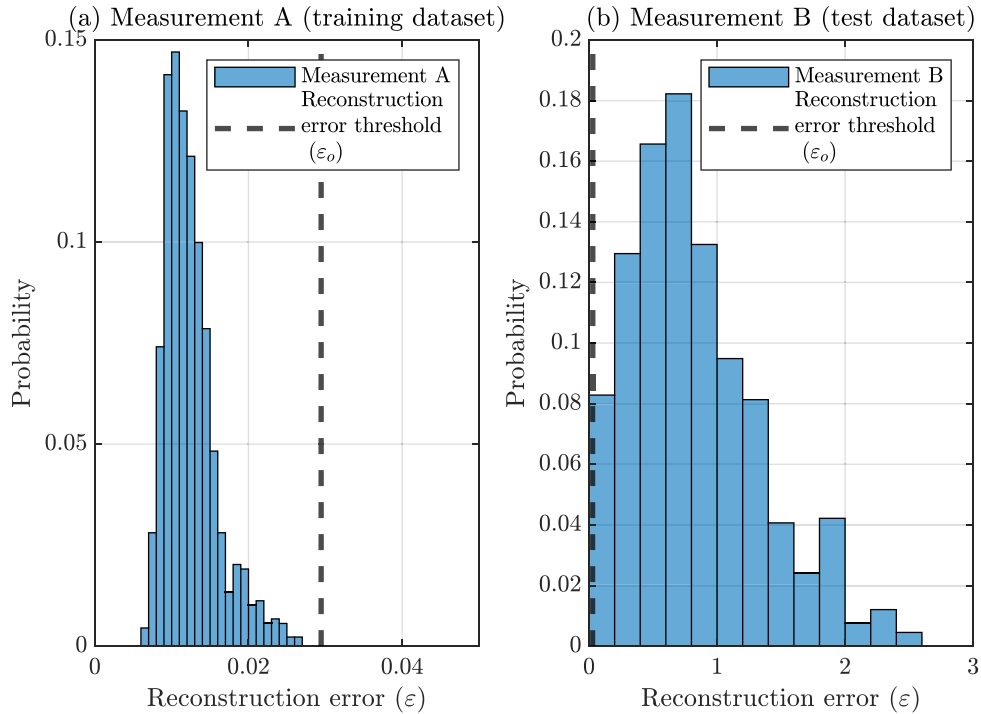


Fig. 11. A representative histograms of reconstruction error ε on Measurement-A and Measurement-B datasets of CS-II for SAE-AD with $\mathbf{h} \in \mathbb{R}^8$.

interesting to point out that \hat{X}_{PCA-AD} shows a reconstructed peak but is offset by a few Hz, and the \hat{X}_{SAE-AD} , and $\hat{X}_{LSTMEncDec-AD}$ also cannot reconstruct the peak. The ε of PCA-AD, SAE-AD, and LSTMEncDec-AD for Sim-B (WindowID 550) is 1.2, 1.2, and 1.1, whereas for Measurement-B (WindowID 223) is 6.64, 2.28, and 2.17 respectively. The ε values of Sim-B (WindowID 550) and Measurement-B (WindowID 223) are much higher than that of Sim-A (WindowID 23) and Measurement-A (WindowID 23) for all AD models. Thus, we confirm that all three AD models are trained to reconstruct only normal healthy sequences, thus not anomalous ones.

4.1.2. Comparison of reconstruction error threshold (ε_0) for AD models

The results of the three AD models while training on healthy datasets of respective case studies is compared against each other in Fig. 10, where subfigure (a) and (b) correspond to CS-I and CS-II, respectively. In both subfigures, vertical and horizontal axes are the ε_0 size of the latent space, respectively. In both subfigures, the star-black, diamond-green, and circle-violet lines correspond to PCA-AD, SAE-AD, and LSTMEncDec-AD models, respectively. In both case studies, the baseline model PCA-AD has much higher ε_0 for smaller latent space size and, as expected, ε_0 reduces quickly with an increase in latent space size because of retention of a more significant number of principal components. The ε_0 of the other two deep learning-based models are much lower than the

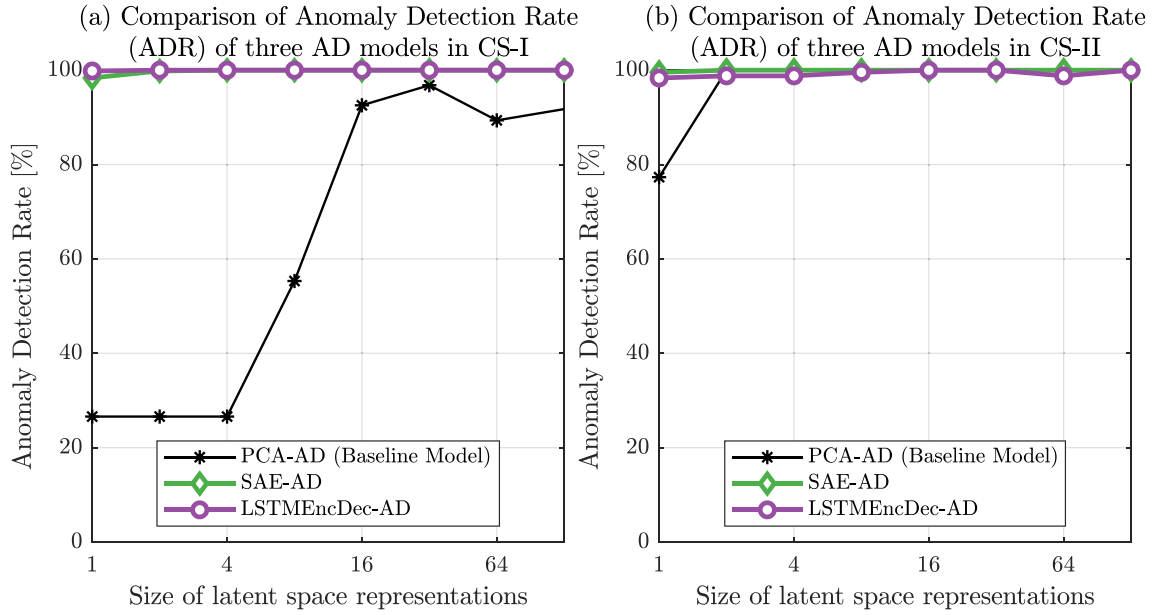


Fig. 12. Comparison of ADR among three AD models on test data.

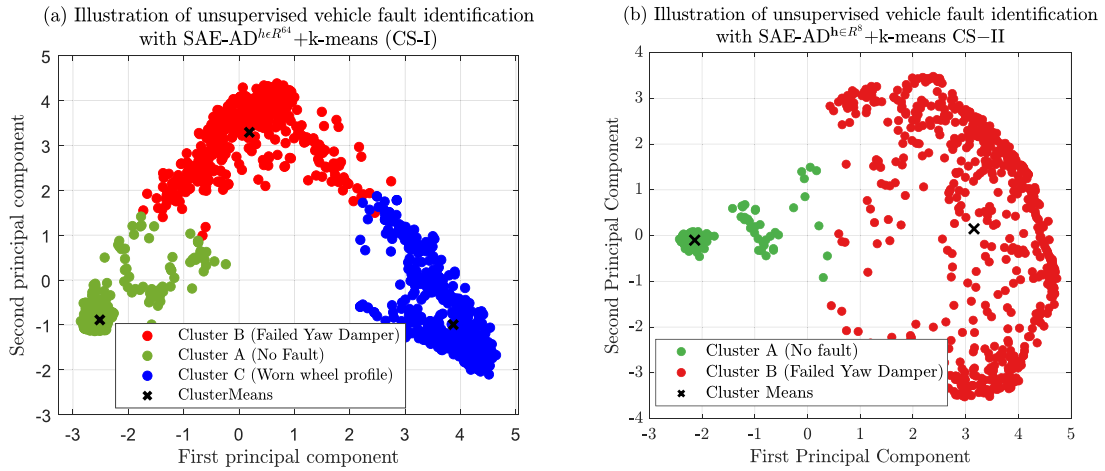


Fig. 13. Illustration of unsupervised fault identification with k -means.

baseline model for the smaller size of latent space. This shows a higher capability of compressing information into smaller dimensions for these deep learning-based models.

4.1.3. Detecting the vehicle running instability occurrences

The histogram of the ε for the training and testing dataset of CS-II is shown in Fig. 11 with objective of demonstrating the procedure of vehicle running instability incidences detection. In both subfigures, the ε is on the horizontal axis, whereas probability on y axis. In subfigures (a) and (b) ε , histograms for Measurement A and B, i.e., training and test dataset, are shown, and, the dashed-black vertical line indicates the reconstruction error threshold (ε_0) of $SAE-AD$ with $\mathbf{h} \in \mathbb{R}^8$, which is 0.027. The ε_0 is the threshold for detecting the anomalies in Measurement B, and the corresponding ADR is 100% (see Fig. 12 (b)).

The testing phase results of the three AD models are compared against each other in Fig. 12, where subfigures (a) and (b) correspond to CS-I and CS-II, respectively. In the subfigures, vertical and horizontal axes are the Anomaly Detection Rate (ADR) and size of the latent space (\mathbf{h}), respectively; and star-black, diamond-green, and circle-violet lines correspond to $PCA-AD$, $SAE-AD$, and $LSTMEncDec-AD$ models,

respectively. In CS-I, $SAE-AD$ and $LSTMEncDec-AD$ models outperform the $PCA-AD$ model; however, $PCA-AD$ shows improved ADR for larger horizontal axis values. Nevertheless, $SAE-AD$ and $LSTMEncDec-AD$ models show almost 100% ADR in CS-I. Similarly, in CS-II, $SAE-AD$ and $LSTMEncDec-AD$ models outperform the $PCA-AD$ model for dimension size 1. Moreover, $SAE-AD$ shows slightly better performance than $LSTMEncDec-AD$ for the smaller size of latent space. The baseline model $PCA-AD$, as expected, performs very poorly for only one latent space dimension. However, the baseline model has 100% ADR for the other latent space dimensions. All models detect vehicle running instability occurrences in both case studies. Moreover, deep learning-based models outperform the baseline model in respective case studies.

4.2. Unsupervised vehicle fault identification instigating vehicle instability

4.2.1. Prevalidation

In this phase, the latent space representation dataset (\mathbf{H}) is used for Vehicle Fault Identification (VFI), and the k -means clustering algorithm is used to find the clusters in \mathbf{H} . The \mathbf{H} datasets of CS-I and CS-II are designated as \mathbf{H}_{CS-I} and \mathbf{H}_{CS-II} respectively, and Fig. 13 (a) and (b)

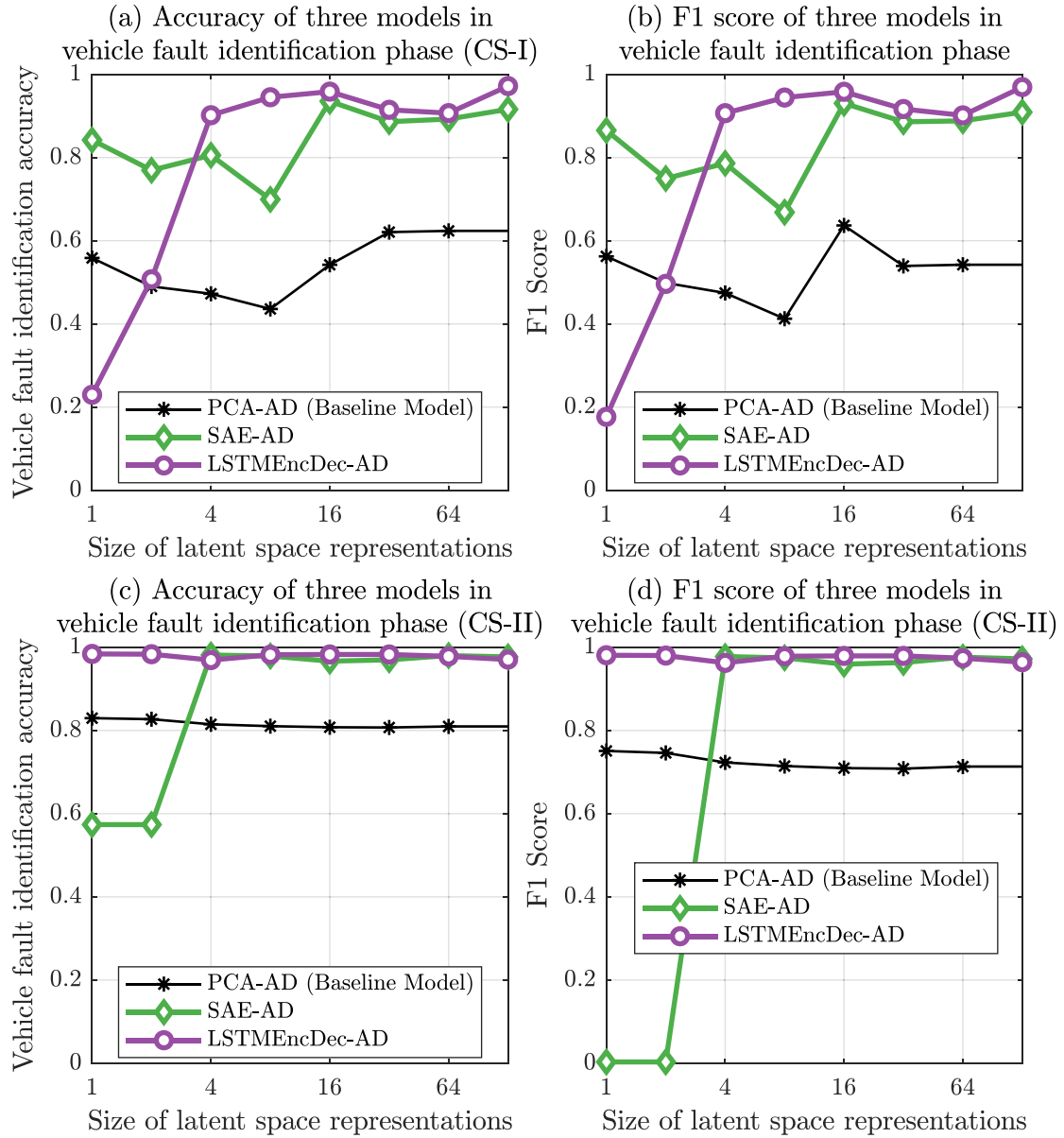


Fig. 14. Comparison of the three AD models in the vehicle fault identification phase. (a) and (c) Vehicle Fault Identification Accuracy (VFIA) plots of CS-I and CS-II; (b) and (d) F1 score in Vehicle Fault Identification phase for CS-I and CS-II.

show respective scatter plots. In both subfigures, the horizontal and vertical axes are 1st and 2nd principal component of the respective datasets \mathbf{H} . In subfigure (a), datapoints of the dataset \mathbf{H}_{CS-I} form three distinct clusters. *K-means* identifies the optimal centroids of the clusters (black cross mark) and designates ClusterID A, ClusterID B, and ClusterID C to these three different clusters. The respective clusters are shown in green, red, and blue colors; moreover, their centroids are $(-2.5, -0.9)$, $(0.2, 3.3)$, and $(3.9, -1)$. Furthermore, we discovered that datapoints of ClusterID A, ClusterID B, and ClusterID C belongs to Sim-A, Sim-B, and Sim-C of CS-I, respectively, with an overall accuracy of 0.9 (out of 1). Moreover, referring to the simulated vehicle's health as mentioned in Section 3.1.3.1, Sim-A, Sim-B, and Sim-C represent "No-fault", "Failed yaw damper", and "Worn wheel" respectively. Thus, these labels are transferred to respective clusters, and these three clusters are called as "no fault", "failed yaw damper" and "worn wheels" clusters.

Interestingly, similar clusters (subfigure (b)) are found in the datapoints of the dataset \mathbf{H}_{CS-II} , which corresponds to CS-II. Here, *k-means* identifies two clusters along with the optimal cluster centroids (black cross mark) and designates ClusterID A and ClusterID B to these two

clusters. The observations of ClusterID A are concentrated around the cluster centroid $(-2.14, -0.1)$ and shown in green color. ClusterID B's observations are in red, and their cluster centroid is at $(3.16, 0.15)$. The variance of ClusterID A datapoints is very low compared to the variance of ClusterID B datapoints. Also, in CS-II we discover that the ClusterID A and ClusterID B belong to Measurement-A and Measurement-B respectively with overall accuracy of 0.98 (out of 1). Moreover, referring to the actual vehicle's health as mentioned in Section 3.1.3.2, Measurement-A and Measurement-B represent "No-fault" and "Failed yaw damper" respectively. Thus, ClusterID A and ClusterID B are designated as "no-fault" and "failed yaw damper" as shown in the subfigure.

Thus, here we show that the datapoints of \mathbf{H} form distinct clusters associated with each fault instigating vehicle running instability. Moreover, we also show that the *k-means* clustering algorithm can be used for identifying the vehicle's fault instigating vehicle instability.

4.2.2. Comparison of AD model's VFI performance

In this section, we compare the performance of three variants of the iVRIDA algorithm for vehicle fault identification during CS-I and CS-II

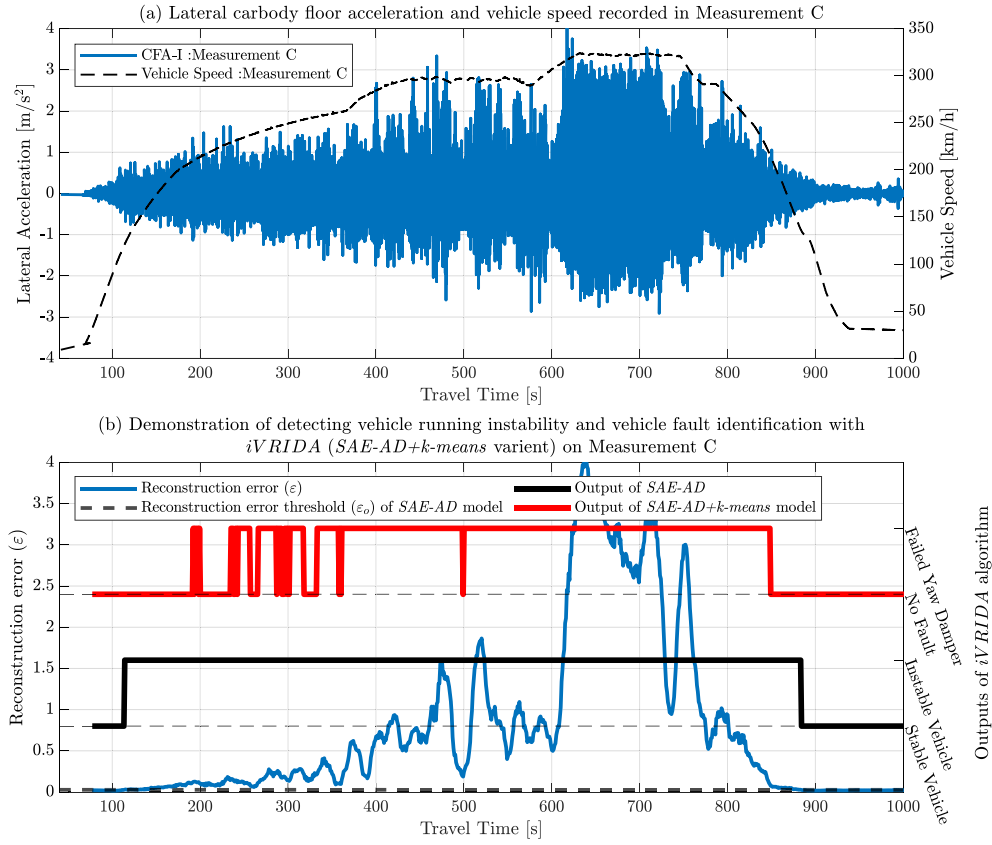


Fig. 15. Demonstration of *iVRIDA* on Measurement C. (a) Lateral carbody floor acceleration and vehicle speed; (b) Output of *iVRIDA* (SAE-AD ^{$h \in \mathbb{R}^8$} + k-means) algorithm.

with the help of a confusion matrices and associated metrics such as accuracy and F1 score [43,49]. The F1 score is the geometric mean of the recall and precision rates of the algorithm [44]. Fig. 14 (a) and (c) show the Vehicle Fault Identification Accuracy (VFIA) comparison of the three AD models for an increasing number of latent space dimensions during CS-I and CS-II, respectively. In both subfigures, the vertical and horizontal axes of the plot are the VFIA and size h , and star-black, diamond-green, and circle-blue lines correspond to *PCA-AD + k-means*, *SAE-AD + k-means* and *LSTMEncDec-AD + k-means* models respectively. In CS-I, the VFIA of the *PCA-AD + k-means* model is poor and hardly crosses the 0.6 value. The *SAE-AD + k-means* VFIA varies between 0.7 and 0.9 according to the size of latent space, which is excellent. The VFIA of *LSTMEncDec-AD + k-means* is very low for $h \in \mathbb{R}^1$ and $h \in \mathbb{R}^2$; however, afterward the *LSTMEncDec-AD + k-means* also outperforms the *SAE-AD + k-means* model. The highest VFIA is 0.97, which is achieved by *LSTMEncDec-AD + k-means*.

Similarly, in subfigure (c), VFIA results of CS-II show that the VFIA of the baseline model *PCA-AD + k-means* is around 0.8 irrespective of the size h . The VFIA for the *SAE-AD + k-means* model is 0.58 for $h \in \mathbb{R}^1$ and $h \in \mathbb{R}^2$; however, afterward, the model exhibits excellent VFIA for a larger size h . The VFIA for the *LSTMEncDec-AD + k-means* model is again excellent, and the model has the capability for high VFIA even if the latent space size is small. It must be noted here that the *SAE-AD + k-means* and *LSTMEncDec-AD + k-means* models outperform the baseline model and exhibit excellent accuracy for vehicle fault identification.

Fig. 14 (b) and (d) show the F1 score in the vehicle fault identification phase for case studies CS-I and CS-II. In both subfigures, the vertical and horizontal axes of the plot are the F1 score and size h . The star-black, diamond-green, and circle-blue lines correspond to *PCA-AD + k-means*, *SAE-AD + k-means* and *LSTMEncDec-AD + k-means* models respectively.

In CS-I, the F1 score of the baseline model *PCA-AD + k-means* is also abysmal and hardly crosses the 0.6 value. The *SAE-AD + k-means*'s F1 score varies between 0.7 and 0.9 according to the size of the latent space, which is excellent. The F1 score of *LSTMEncDec-AD + k-means* is also low for $h \in \mathbb{R}^1$ and $h \in \mathbb{R}^2$; however, afterward, the *LSTMEncDec-AD + k-means* also outperforms the *SAE-AD + k-means* model. The highest F1 is 0.97 which is achieved by *LSTMEncDec-AD + k-means*.

Similarly, in subfigure (d), the F1 score of CS-II shows that the F1 score of the baseline model *PCA-AD + k-means* is around 0.75, irrespective of the size h . The F1 score for the *SAE-AD + k-means* model is 0 for the $h \in \mathbb{R}^1$ and $h \in \mathbb{R}^2$, however afterward, the model exhibits excellent F1 score for larger sizes h . The F1 for the *LSTMEncDec-AD + k-means* model is always excellent, and the model has the capability for high F1 even if the latent space size is very small. It must be noted here that the *SAE-AD + k-means* and *LSTMEncDec-AD + k-means* models outperform the baseline model and exhibit excellent accuracy for vehicle fault identification. The F1 score above 0.95 emphasize both models' superb precision and recall rates. This means that the *SAE-AD + k-means* and *LSTMEncDec-AD + k-means* algorithms are accurate, precise, and reliable for identifying vehicle faults instigating vehicle running instability. The following section shows a virtual demonstration of the *iVRIDA* algorithm.

5. Demonstration of the *iVRIDA* algorithm and discussion

5.1. Demonstration of the *iVRIDA* algorithm on Measurement-C

The *iVRIDA* algorithm is here deployed on the accelerations acquired during Measurement C to demonstrate and validate the algorithm. The vehicle accelerations recorded at CFA-I and CFA-II during Measurement C are pre-processed according to the procedure described in Section

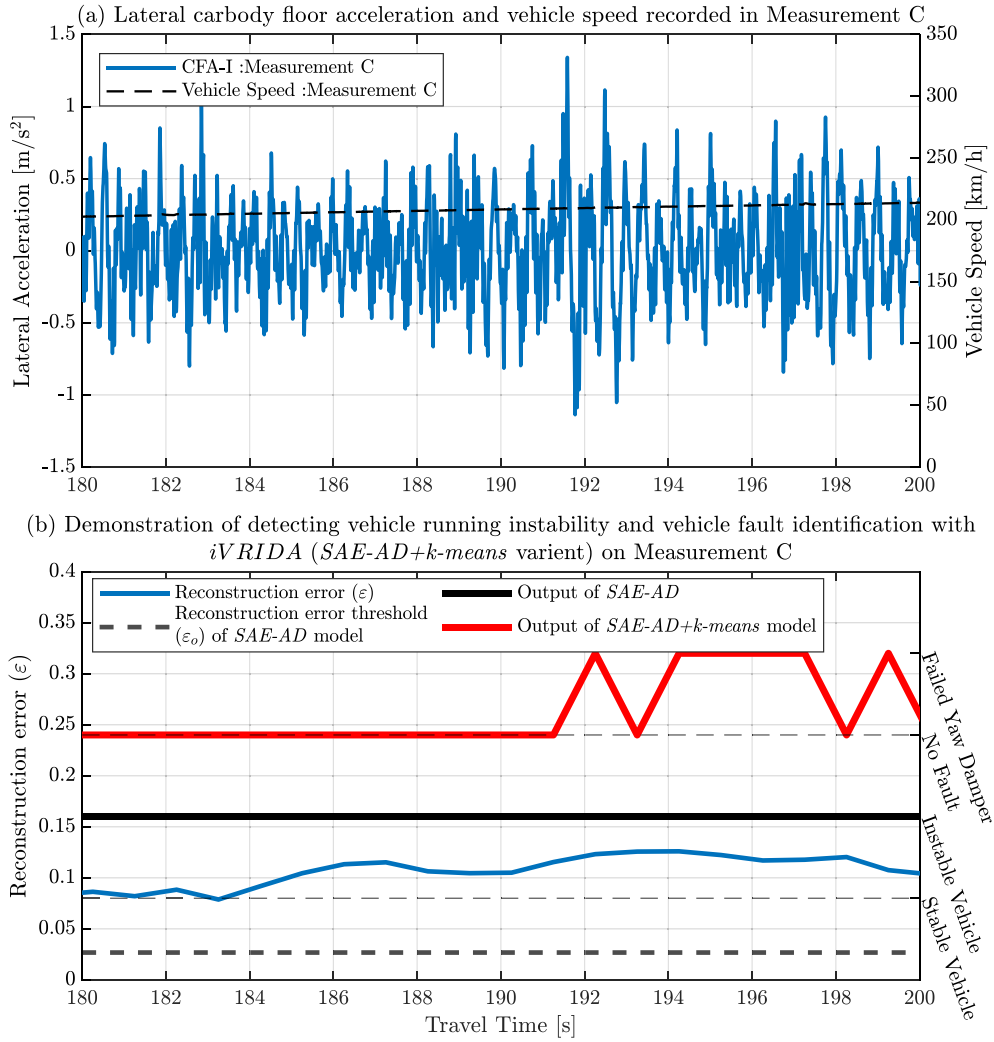


Fig. 16. Closeup of Fig. 15 i.e. Demonstration of *iVRIDA* on Measurement C. (a) Lateral carbody floor acceleration and vehicle speed profile; (b) Output of *iVRIDA* ($\text{SAE-AD}^{h \in \mathbb{R}^8} + k\text{-means}$) algorithm.

3.3.1 where the acceleration signals are windowed, normalized, and features are extracted. In the demonstration phase, the $\text{SAE-AD} + k\text{-means}$ model of the *iVRIDA* scheme is used, and more specifically, we deploy $\text{SAE-AD} + k\text{-means}$ with $h \in \mathbb{R}^8$. The results of the Vehicle Running Instability Detection (VRID) and Vehicle Fault Identification (VFI) are explained with the help of Fig. 15. Fig. 15 (a) shows the vehicle speed and lateral acceleration recorded at CFA-I during Measurement C. In Fig. 15 (a), the time is on the horizontal axis, acceleration is on the left vertical axis, and vehicle speed is on the right vertical axis. The acceleration and vehicle speed are shown in blue and dashed-black line, respectively. The vehicle accelerates from 0 km/h to 330 km/h, and then the vehicle speed is gradually reduced to 30 km/h. The acceleration amplitude gradually increases along with increasing vehicle speed and rises to 4 m/s², making the ride very violent. The results obtained after analyzing the accelerations with the *iVRIDA* scheme are shown in Fig. 15 (b). In this subfigure, the time is on the horizontal axis, whereas the left and right vertical axis are ϵ and outputs of VRID and VFI branches of *iVRIDA*.

In Fig. 15 (b), the ϵ , $\epsilon_0^{\text{SAE-AD}^{h \in \mathbb{R}^8}}$, VRID output and VFI output are shown by blue, dashed black, black, and red lines, respectively. The blue curve, i.e., reconstruction error (ϵ), the curve starts with zero and gradually increases as the acceleration increases along with increasing vehicle speed. The ϵ curve crosses the threshold $\epsilon_0^{\text{SAE-AD}^{h \in \mathbb{R}^8}}$ at 114 s and stays above the same until 883 s. Even though the ϵ curve is above the

$\epsilon_0^{\text{SAE-AD}^{h \in \mathbb{R}^8}}$ in-between these two timestamps, the ϵ curve varies significantly and is directly correlated with acceleration. The ϵ reaches its maximum when the acceleration reaches its maximum. The VRID output curve is obtained from the ϵ curve. The VRID output shows that the vehicle is running unstable in-between 114 s till 883 s; otherwise, the vehicle is stable. The VFI output curve shows the identified vehicle fault, and it is observed that the curve fluctuates between “No Fault” and “Failed Yaw Damper” at the beginning of the measurement but eventually settles at the “Failed Yaw Damper” when the vehicle speed is above 250 km/h. This VFI output curve ultimately comes down to “No Fault” after the vehicle speed reduces below 250 km/h because the acceleration also reduces along with speed. We make the following observations from Fig. 15:

- (1) An enlarged section of Fig. 15 is shown in Fig. 16, where we take a close look at the vehicle behavior and outputs of *iVRIDA* from 180 s to 200 s. The vehicle speed and acceleration amplitudes are around 200 km/h and 0.5 – 1 m/s², respectively. The spurling index during these 20 s is also approximately 2.5 (see Fig. 8), which is quite well within acceptable limits. Moreover, the bandpass-filtered lateral acceleration doesn't show a strong sinusoidal pattern, typically visible during vehicle running instability. However, these accelerations are eventually amplified when vehicles exhibit violent running instability starting after

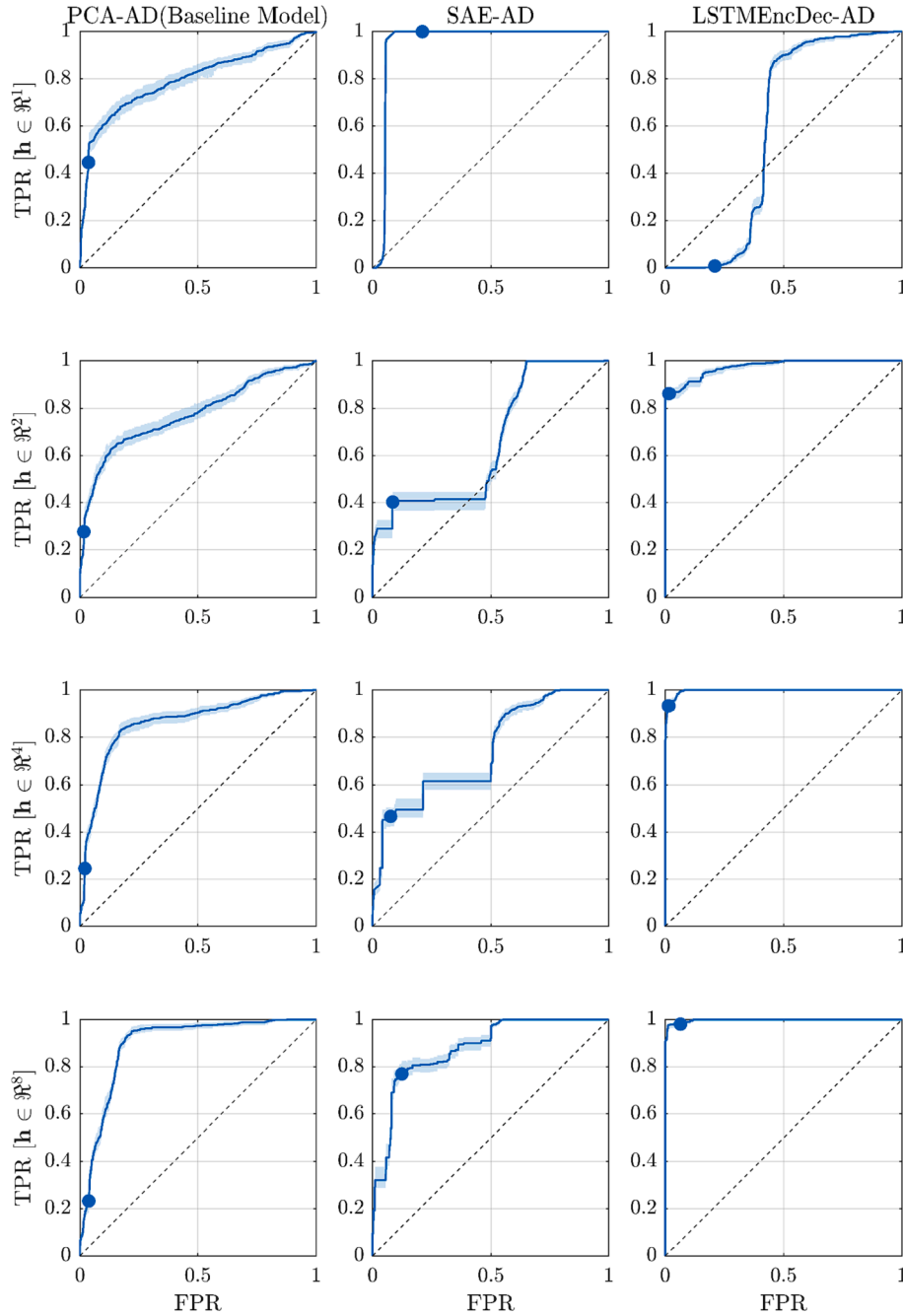


Fig. 17. ROC plot of yaw damper fault in CS-I for PCA-AD, SAE-AD and LSTMEncDec-AD for latent space size 1 to 8.

300 s. Usually, the vehicle gradually becomes unstable. This phase of initiating vehicle running instability is also typically observed in high-speed rail vehicles in China [38] and referred to as small amplitude hunting/instability.

The *iVRIDA* output for these 20 s is also shown in subfigure (b). The subfigure shows that the *VRID* branch of *iVRIDA* successfully detects the presence of small amplitude running instability. The *VFI* branch also identifies vehicle fault, i.e., “Failed Yaw Damper” which instigates small amplitude running instability; however, only after the 190-second mark. Nevertheless, the proposed *iVRIDA* detects the presence of small amplitude running instability which is present in the timestamp from 150 s to 300 s.

- (2) The *iVRIDA* consistently detects running instability and corresponding vehicle fault when the vehicle speed is above 200 km/h.
- (3) The vehicles are typically dynamically stable at lower operational speeds, which is also identified by *iVRIDA*, specifically at the beginning and end of Measurement C. The *VRID* and *VFI* outputs are directly correlated with reconstruction error (ϵ), directly correlated with the amplitude of lateral carbody floor acceleration.

5.2. Discussion on limitations of the current research

The section above, demonstrates the potential of deploying *iVRIDA* framework; as with every research, the current work has its limitations which must be acknowledged. However, it is important to note that these limitations should not be viewed as negative aspects of the *iVRIDA*

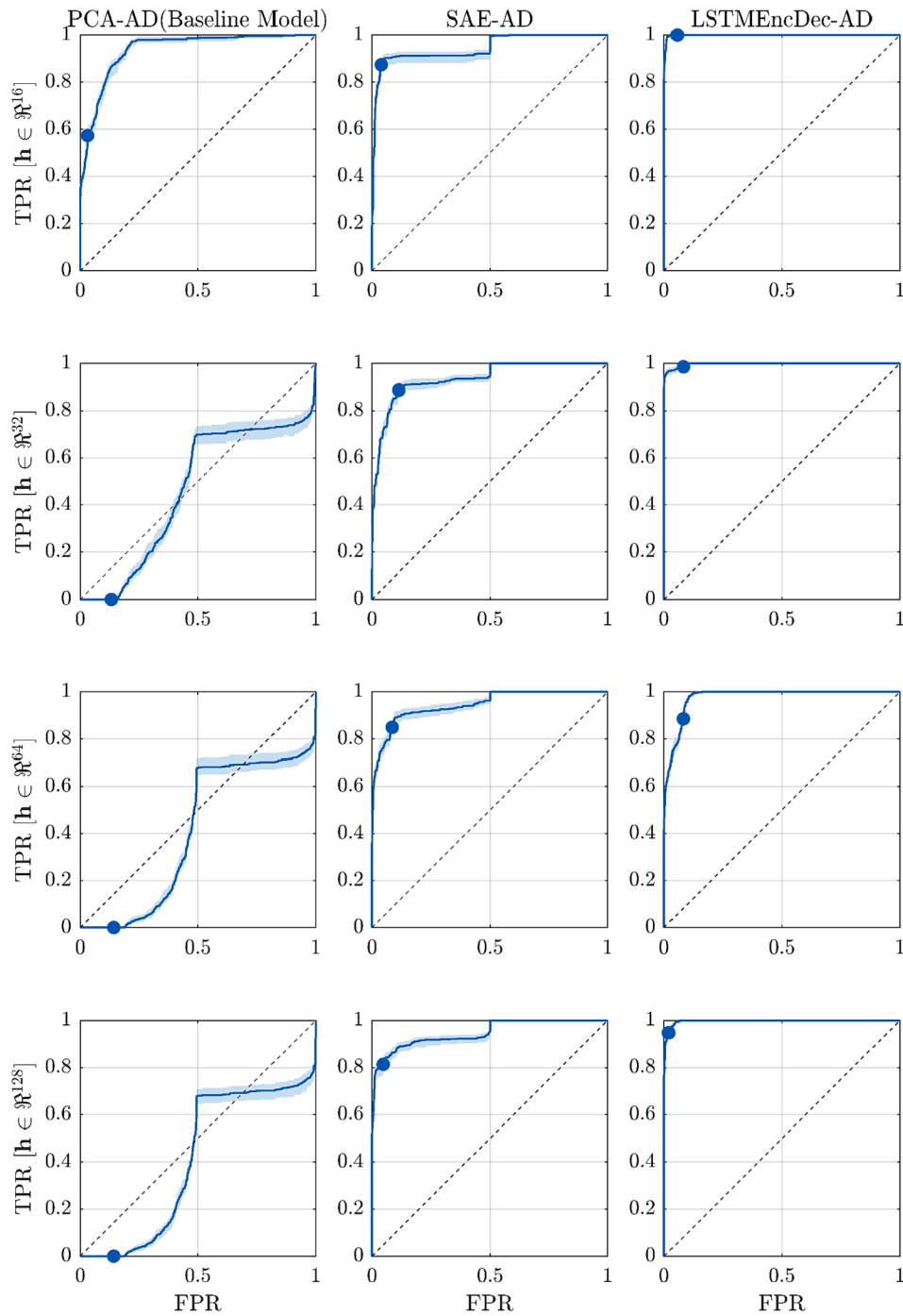


Fig. 18. ROC plot of yaw damper fault in CS-I for PCA-AD, SAE-AD and LSTMencDec-AD for latent space size 16 to 128.

framework, but rather as opportunities for further research and development.

- (1) The current research focuses on identifying and analyzing rigid body dynamic instability in rail vehicles. However, it is important to note that other forms of instability, such as those associated with flexbody modes of wheelset, bogie frame, and carbody, can also occur in real-world. While these instabilities are typically mitigated through proper vehicle design, they may still arise due to malfunctions of certain components. Additionally, these instabilities may manifest at different frequency ranges [4] and require different sensor types and architectures than those studied in this research. Further research investigation is need to

ensure the performance of the *iVRIDA* framework is not deteriorated due to appearance of instabilities associated with flexbody modes.

- (2) The *iVRIDA* framework's training dataset, Measurement A, represents an ideal healthy vehicle. Test datasets, Measurements B and C, represent ideal faulty datasets. The datasets originate from the same vehicle and were acquired during certification tests, which may introduce bias. However, it's important to note that in real-world scenarios, the distinction between healthy and faulty vehicles is not always clear cut as vehicle degradation is a continuous process. Additionally, the current dataset may not fully reflect real-world scenarios as Measurement A represents an ideal healthy vehicle and Measurements B and C represent ideal

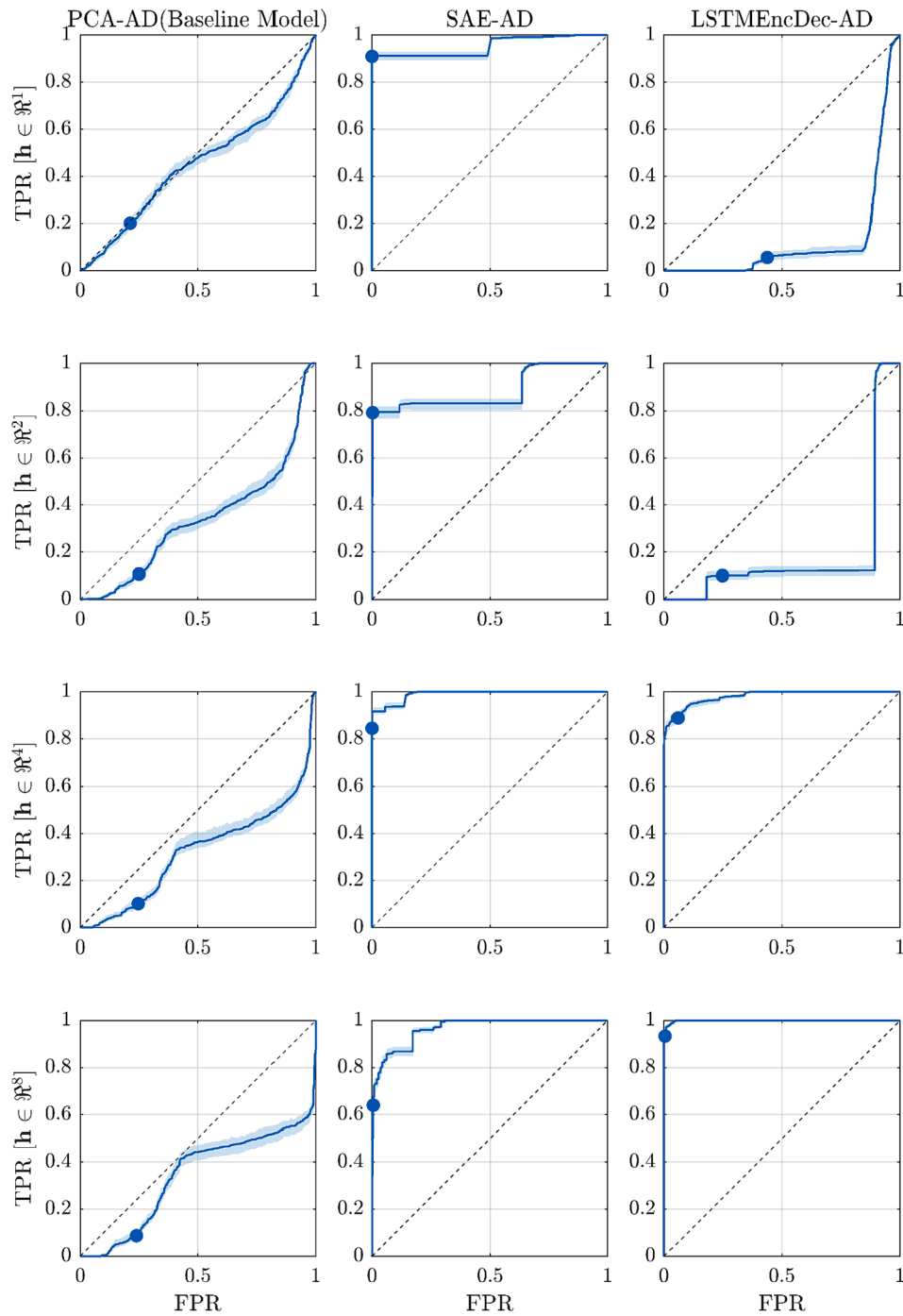


Fig. 19. ROC plot of wheel profile fault in CS-I for PCA-AD, SAE-AD and LSTMEncDec-AD for latent space size 1 to 8.

faulty vehicles. In reality, the boundary between healthy and faulty is not always clear cut.

- (3) This study utilizes a machine learning model that is trained on a limited amount of data, which may affect the generalization capability of the model to new, unseen data. To improve the model's performance and increase the confidence in its predictions, future research should consider using larger and more diverse datasets and sensitivity studies.

In conclusion, the *iVRIDA* framework demonstrated encouraging results until now. However, to further strengthen its performance, it is important to expand the dataset and rigorously verify its algorithm for increased accuracy.

6. Conclusions and future work

6.1. Conclusions

The appearance of running instability of high-speed rail vehicles negatively affects operational safety and passenger ride comfort. Moreover, the emergence of running instability is sporadic; i.e. it is an anomaly in the vehicle-track dynamic interaction. This paper proposes an unsupervised Anomaly Detection (AD) framework, *iVRIDA*, is developed to detect the emergence of the vehicle running instability of high-speed rail vehicles and to identify corresponding vehicle fault. More importantly, the *iVRIDA* framework utilizes only carbody floor accelerations which are easy to measure, store, and transfer to the cloud

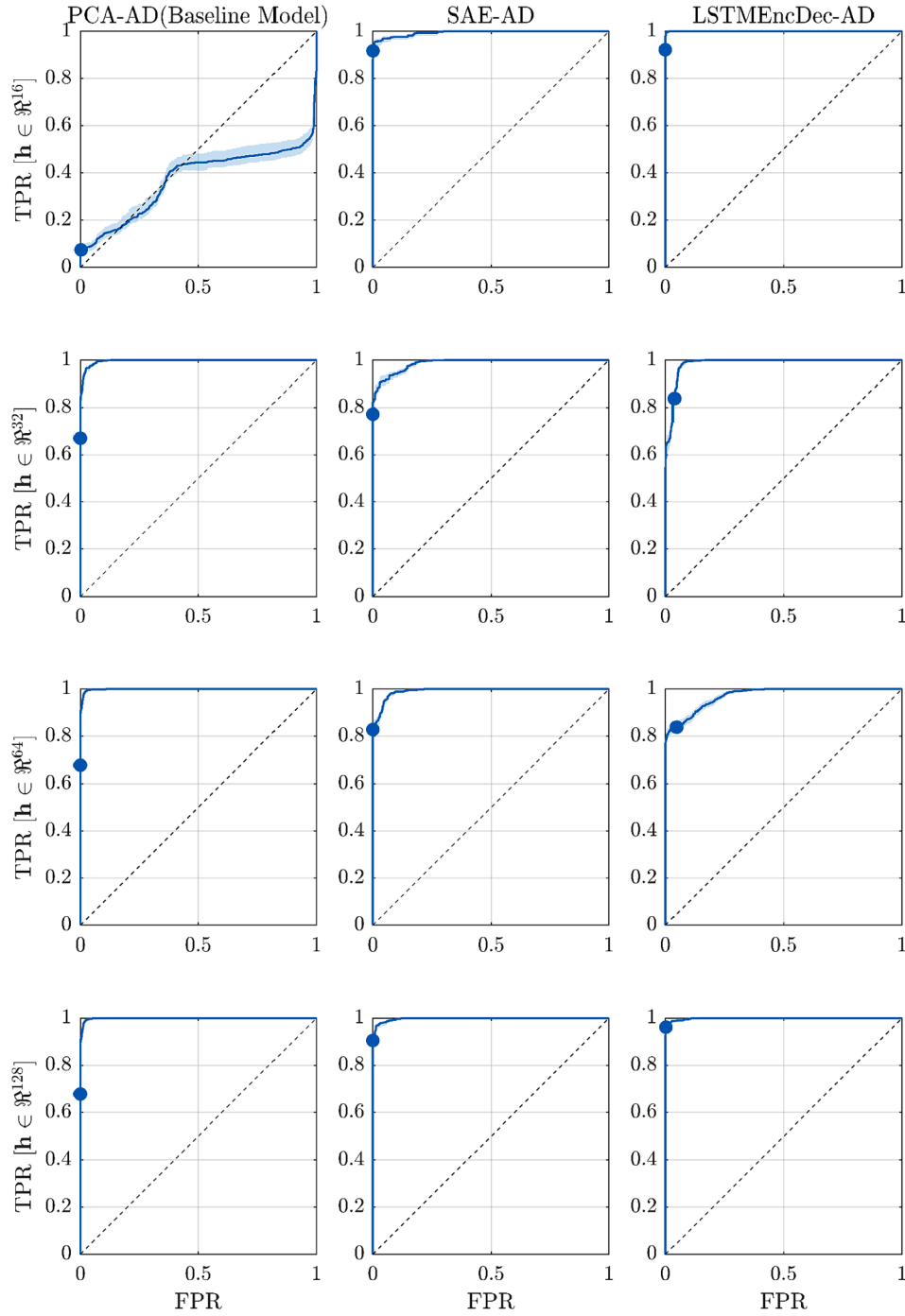


Fig. 20. ROC plot of wheel profile fault in CS-I for PCA-AD, SAE-AD and LSTMEncDec-AD for latent space size 16 to 128.

for further processing; additionally, the placement of accelerometers in the carbody is simple and cost-effective. The present work is concluded as follows:

- (1) This paper firstly expresses the vehicle running instability detection problem as an Anomaly Detection (AD) problem, which is a unique perspective towards vehicle running instability detection problem and has not yet been explored by other researchers. The problem of vehicle running instability detection and corresponding vehicle fault identification should be treated as an AD problem because vehicles seldomly run unstable during service, and accelerations recorded on the healthy (stable) vehicles are abundantly available. Moreover, ground truth about

vehicle running instability and respective vehicle fault is usually not collected and reported systematically and consistently. Thus, preparing labeled datasets mandatory for training supervised classification algorithms is impossible.

- (2) Thus, this paper proposes an unsupervised vehicle running instability detection algorithm, called *iVRIDA*, for detecting occurrences of vehicle running instability of high-speed rail vehicles. The *iVRIDA* framework incorporates three different AD models, namely PCA-AD (baseline model), SAE-AD, and LSTMEncDec-AD, to detect vehicle running instability from carbody floor acceleration. A popularly used unsupervised clustering algorithm, *k-means* is deployed to identify clusters in the latent space of AD models. We show that faults instigating the running

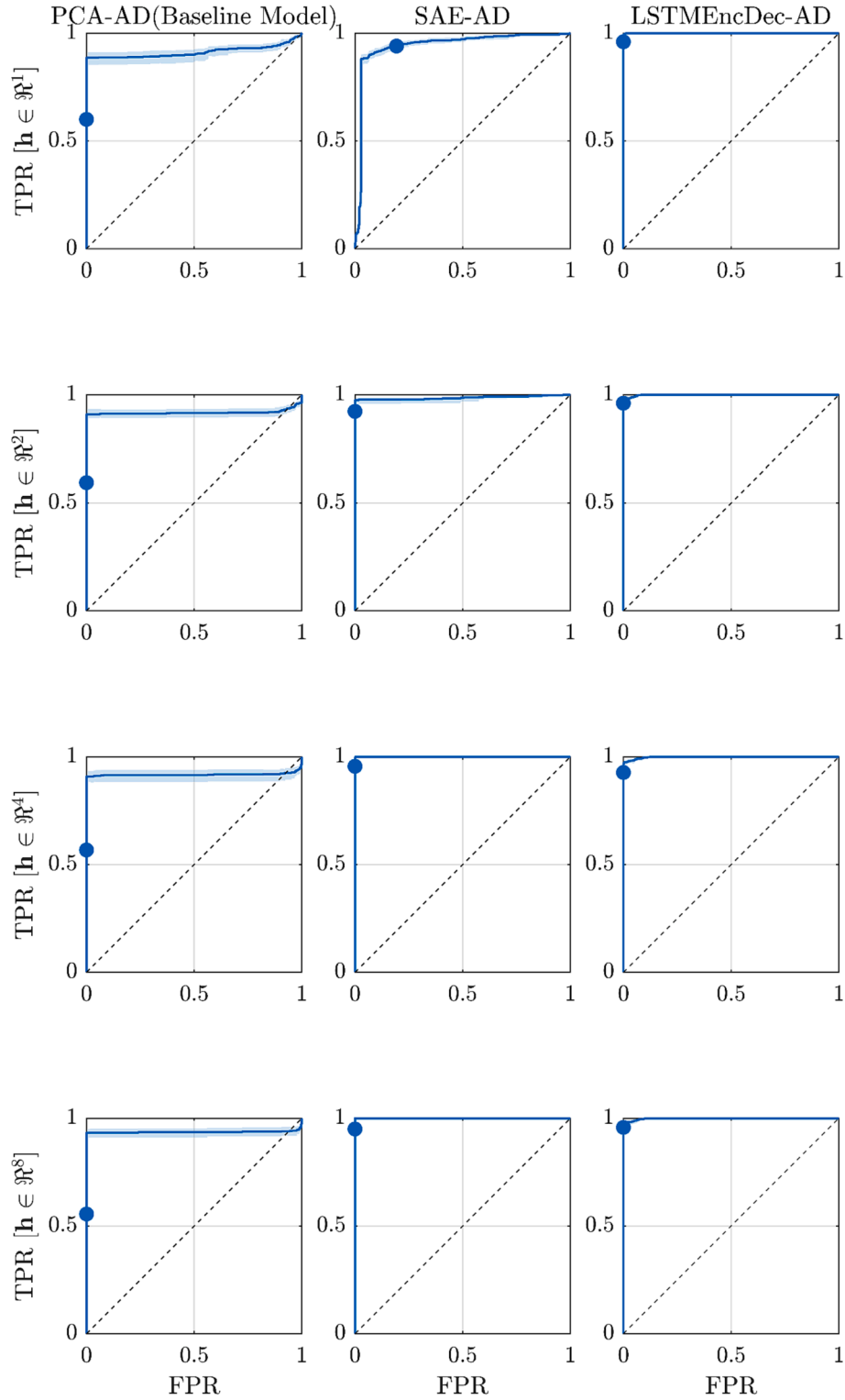


Fig. 21. ROC plot of yaw damper fault in CS-II for PCA-AD, SAE-AD and LSTMEncDec-AD for latent space size 1 to 8.

instability develop distinct clusters in the latent space. Thus, an unsupervised machine-learning algorithm can identify the faults instigating the vehicle running instability.

- (3) We demonstrate the potency of the *iVRIDA* framework with help of two case studies. These case studies are based on simulated and measured vehicle acceleration of European high-speed rail vehicles respectively. In both case studies, we found that the SAE-AD and LSTMEncDec-AD models outperform the baseline model in

unsupervised vehicle running instability detection and unsupervised fault identification.

- (4) The proposed *iVRIDA* framework is a generic framework where any signal reconstruction-based anomaly detection algorithm can be integrated in the VRID branch of *iVRIDA*. Moreover, any unsupervised / supervised clustering / classification algorithm can be integrated into the VFI branch to identify the underlying vehicle faults that instigate vehicle running instability.

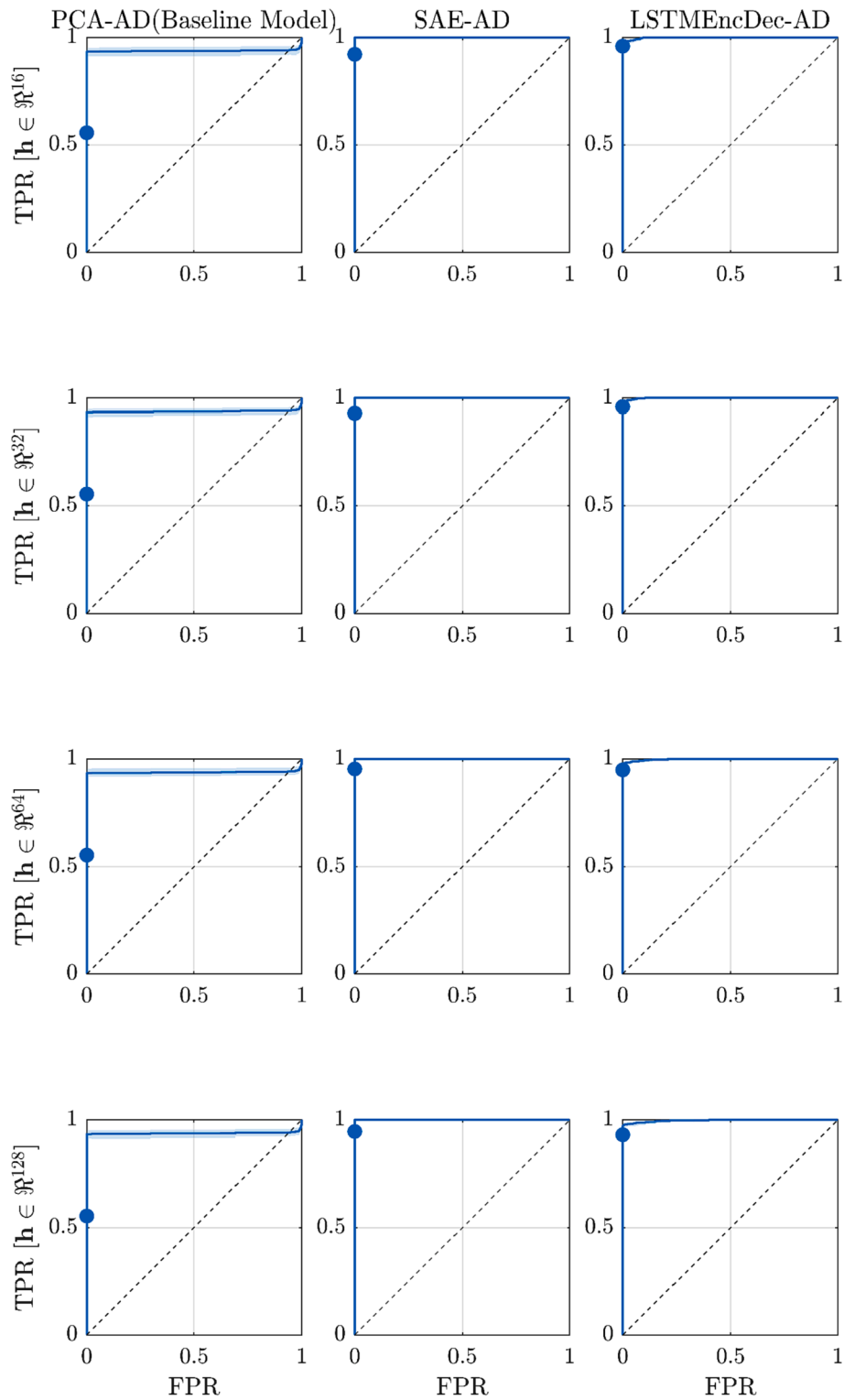


Fig. 22. ROC plot of yaw damper fault in CS-II for PCA-AD, SAE-AD and LSTMEncDec-AD for latent space size 16 to 128.

- (5) The proposed *iVRIDA* framework is under continuous development and validation process. The next step is to implement the *iVRIDA* framework on vehicle accelerations recorded on a fleet of rail vehicles.

6.2. Future work

6.2.1. Implementation of *iVRIDA* framework on fleet of inservice vehicles

- (1) The very interesting offshoot of current research work is to implement *iVRIDA* framework on fleet of passenger trains with federated learning approach.

- (2) The fleetwise implementation of *iVRIDA* framework enables investigating the performance of *iVRIDA* in very common everyday situations such as
 - (a) A particular vehicle exhibiting running instability at one or more than one track sections while other vehicles operating smoothly on same track sections.
 - (b) More than one vehicle exhibiting running instability on a particular track sections on particular day but running smoothly on rest days.
 - (c) A particular vehicle exhibiting running instability at a particular track section on a specific time whereas other vehicle smoothly passing through same track section on a same time.
- (3) Every individual vehicle may introduce its own bias in accelerations; this must be addressed before fleetwide implementation of *iVRIDA*.

6.2.2. Further developments of *iVRIDA* framework

- (1) The core of *iVRIDA* framework can be strengthened with diversification of data sources i.e. incorporating realtime information on track irregularities, rail profiles, wheel profiles and so on. This will open up very interesting avenue of research work.
- (2) The framework can also be expanded to incorporate various other types of features [46] extracted from carbody floor accelerations.
- (3) The *iVRIDA* framework can be extended to incorporate Variational Autoencoder, Convolutional Autoencoder, Generative adversarial network models. Furthermore, hyperparameters of each model can also be finetuned for further improvement in the performance.

CRediT authorship contribution statement. **Rohan Kulkarni:** Project administration, Conceptualization, Methodology, Investigation, Formal analysis, Visualization, Software, Validation, Writing – original draft, Writing – review & editing. **Alireza Qazizadeh:** Supervision, Writing – review & editing. **Mats Berg:** Project administration, Supervision, Writing – review & editing, Funding acquisition.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Rohan Kulkarni reports financial support was provided by European Union. Rohan Kulkarni reports financial support was provided by Swedish Transport Administration.

Data availability

The authors do not have permission to share data.

Acknowledgements

The authors gratefully acknowledge the vehicle accelerations and technical information shared by CAF. The authors are also grateful for technical support provided by the Swedish Transport Administration (Trafikverket) during this research project.

This document reflects the views of the authors and does not necessarily reflect the views or policy of the European Commission. While efforts have been made to ensure the accuracy and completeness of this document, the PIVOT II consortium of Shift2Rail shall not be liable for any errors or omissions, however caused.

Funding

The authors gratefully acknowledge the financial support from the

European Union's Horizon 2020 research and innovation programme under Grant Agreement No 881807 (PIVOT II) and the Swedish Transport Administration under research Excellence Area 1 via the KTH Railway Group.

Appendix A.: ROC plots of all models in CS-I and CS-II

ROC plots of CS-I

Yaw damper fault

The ROC curves along with optimal operating point for yaw damper fault class in CS-I for *PCA-AD*, *SAE-AD* and *LSTMEncDec-AD* for various latent space sizes are shown in Fig. 17 and Fig. 18. In all plots, x and y axis are False Positive Rate (FPR) and True Positive Rate (TPR). Moreover, in each figure, the first, second and third column corresponds to *PCA-AD + kmeans*, *SAE-AD + kmeans* and *LSTMEncDec-AD + kmeans* models. Further more each row of plot corresponds to different latent space dimensions and these dimension are mentioned in Y axis label. The ROC plots also confirm that the performance of *SAE-AD + kmeans* and *LSTMEncDec + kmeans* models is significantly better than baseline model.

Wheel profile fault

The ROC curves for wheel profile fault class in CS-I for *PCA-AD*, *SAE-AD* and *LSTMEncDec-AD* for various latent space sizes are shown in Fig. 19 and Fig. 20.

ROC plots of CS-II

Yaw damper fault

The ROC curves for yaw damper fault class in CS-II for *PCA-AD*, *SAE-AD* and *LSTMEncDec-AD* for various latent space sizes are shown in Fig. 21 and Fig. 22.

References

- [1] R. Kulkarni, A. Qazizadeh, M. Berg, et al., Investigating the effect of the equivalent conicity function's nonlinearity on the dynamic behaviour of a rail vehicle under typical service conditions, *Veh. Syst. Dyn.* 60 (2022) 3484–3503.
- [2] Technical Specification for Interoperability (TSI) : 'Rolling Stock — Locomotives and passenger rolling stock' subsystem of the rail system in the European Union issued on 18 November 2014.
- [3] M. Asplund, P. Söderström, Dälig gång fas 1: Förstudie för att förbättra gången för passagerartåg (Samarbetsprojekt mellan Trafikverket och SJ), Stockholm, Sweden, 2018.
- [4] S. Iwnicki, *Handbook of railway vehicle dynamics*. First Edition, CRC Press, 2006.
- [5] Y. Zeng, W. Zhang, D. Song, A new strategy for hunting alarm and stability evaluation for railway vehicles based on nonlinear dynamics analysis, *Proc. Inst. Mech. Eng. Part F J. Rail Rapid Transit.* 234 (2020) 54–64.
- [6] R. Kulkarni, A. Qazizadeh, M. Berg, et al., Vehicle running instability detection algorithm (*VRIDA*): a signal based onboard diagnostic method for detecting hunting instability of rail vehicles, *Proc. Inst. Mech. Eng. Part F J. Rail Rapid Transit.* 236 (2022) 262–274.
- [7] EN 14363:2016+A1 - Railway applications – Testing and simulation for the acceptance of running characteristics of railway vehicles – Running behaviour and stationary tests. 2019.
- [8] UIC 515-1: Passenger rolling stock - Trailer bogies - Running gear - General provisions applicable to the components of trailers bogies. International Union of Railways; 2003.
- [9] FRA Regulations. Title 49. Transportation, Part 238 – Passenger Equipment Safety Standards (49CFR238), Department of Transportation, Federal Railroad Administration, October 1, 2003.
- [10] S. Bruni, F. Cheli, A. Collina, et al., Road test data procedures for evaluating the hunting instability threshold of a railway vehicle from on board measurements, *Veh. Syst. Dyn.* 33 (2000) 168–179.
- [11] T. Forsberg, M. Berg, S. Stichel, et al., Condition monitoring of train ride stability, *Int. Conf. Railw. Bogies Run. Gears.* (2007) 271–280.
- [12] Calvo LE, Castellanos GA, Gimenez G. Analysis of instabilities in railway vehicles employing operational modal analysis. 2015 Joint Rail Conference 2015. p. 1–7.
- [13] N. Bosso, A. Gugliotta, N. Zampieri, Design and testing of an innovative monitoring system for railway vehicles, *Proc. Inst. Mech. Eng. Part F J. Rail Rapid Transit.* 232 (2018) 445–460.

- [14] D. Lebel, C. Soize, C. Funfschilling, et al., High-speed train suspension health monitoring using computational dynamics and acceleration measurements, *Veh. Syst. Dyn.* 58 (2020) 911–932.
- [15] L. Gasparetto, S. Alfi, S. Bruni, Data-driven condition-based monitoring of high-speed railway bogies, *Int. J. Rail Transp.* 1 (2013) 42–56.
- [16] J. Ning, Q. Liu, H. Ouyang, et al., A multi-sensor fusion framework for detecting small amplitude hunting of high-speed trains, *J. Vib. Control.* 24 (2018) 3797–3808.
- [17] Y. Ye, J. Ning, Small-amplitude hunting diagnosis method for high-speed trains based on the bogie frame's lateral-longitudinal-vertical data fusion, independent mode function reconstruction and linear local tangent space alignment, *Proc. Inst. Mech. Eng. Part F J. Rail Rapid Transit.* 233 (2019) 1050–1067.
- [18] Y. Ye, P. Huang, Y. Zhang, Deep learning-based fault diagnostic network of high-speed train secondary suspension systems for immunity to track irregularities and wheel wear, *Railw. Eng. Sci.* 30 (2022) 96–116.
- [19] R.R. Kulkarni, R.L. Giossi, P. Damsongsang, et al., *iVRIDA*: intelligent vehicle running instability detection algorithm for high-speed rail vehicles using temporal convolution network – a pilot study, *PHM Soc. Eur. Conf.* 7 (2022) 269–277.
- [20] J. Sun, E. Meli, W. Cai, et al., A signal analysis based hunting instability detection methodology for high-speed railway vehicles, *Veh. Syst. Dyn.* (2020) 1–23.
- [21] J. Sun, E. Meli, X. Song, et al., A novel measuring system for high-speed railway vehicles hunting monitoring able to predict wheelset motion and wheel/rail contact characteristics, *Veh. Syst. Dyn.* 1–23 (2022).
- [22] D. Wang, J. Ning, F. Zhao, et al., A small-amplitude hunting motion recognition method based on transfer learning, *J. Vib. Control.* 107754632211170 (2022).
- [23] Ye Y, Huang P. Deep learning-based fault diagnostic network of high-speed train secondary suspension systems for immunity to track irregularities and wheel wear. 2021.
- [24] R Kulkarni, A Qazizadeh, M Berg, et al., Fault detection and isolation method for vehicle running instability from vehicle dynamics response using machine learning. 11th International Conference on Railway Bogies and Running Gears, Budapest (2019).
- [25] Y. Lei, B. Yang, X. Jiang, et al., Applications of machine learning to machine fault diagnosis: a review and roadmap, *Mech. Syst. Signal Process.* 138 (2020), 106587.
- [26] Rahat M, Mashhadi PS, Nowaczyk S, et al. Domain Adaptation in Predicting Turbocharger Failures Using Vehicle's Sensor Measurements. 2022.
- [27] Malhotra P, Ramakrishnan A, Anand G, et al. LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection. 2016.
- [28] B. Rezaeianjouybari, Y. Shang, Deep learning for prognostics and health management: State of the art, challenges, and opportunities, *Meas. J. Int. Meas. Confed.* (2020) 163.
- [29] O. Fink, Q. Wang, M. Svensén, et al., Potential, challenges and future directions for deep learning in prognostics and health management applications, *Eng. Appl. Artif. Intell.* 92 (2020), 103678.
- [30] M.A.F. Pimentel, D.A. Clifton, L. Clifton, et al., A review of novelty detection, *Signal Process.* 99 (2014) 215–249.
- [31] I.T. Jolliffe, J. Cadima, Principal component analysis: a review and recent developments, *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* 374 (2016) 20150202.
- [32] Ng A. Sparse Autoencoder. CS294A Lect. notes. Stanford University; p. 1–19.
- [33] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.* 9 (1997) 1735–1780.
- [34] L. Basora, X. Olive, T. Dubot, Recent advances in anomaly detection methods applied to aviation, *Aerospace.* 6 (2019) 117.
- [35] J. Shlens, A tutorial on principal component analysis, 2014.
- [36] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, *Science* (80-) 313 (2006) 504–507.
- [37] B.A. Olshausen, D.J. Field, Sparse coding with an overcomplete basis set: a strategy employed by V1? *Vision Res.* 37 (1997) 3311–3325.
- [38] David Arthur, S. Vassilvitskii, K-Means++: the advantages of careful seeding, in: *Proc. Eighteenth Annu. ACM-SIAM Symp. Discret. Algorithms, SODA 2007*, New Orleans, Louisiana, USA, 2007.
- [39] B. Dirks, Vehicle Dynamics Simulation of Wheel Wear for Swedish High-Speed Train X2000. KTH Royal Institute of Technology, Stockholm, Sweden, 2003.
- [40] EN 13848-5:2017 - Railway applications – Track – Track geometry quality – Part 5: Geometric quality levels – Plain line, switches and crossings, 2017.
- [41] K. Knothe, S. Stichel. *Rail Vehicle Dynamics*, Second Edition, Springer Berlin Heidelberg, Berlin, 2016.
- [42] K. Rombach, G. Michau, O. Fink, Contrastive learning for fault detection and diagnostics in the context of changing operating conditions and novel fault types, *Sensors* 21 (2021) 3550.
- [43] A. De Rosa, R. Kulkarni, A. Qazizadeh, et al., Monitoring of lateral and cross level track geometry irregularities through onboard vehicle dynamics measurements using machine learning classification algorithms, *Proc. Inst. Mech. Eng. Part F J. Rail Rapid Transit.* 235 (2021) 107–120.
- [44] S.B. Kotsiantis, Supervised machine learning: a review of classification techniques, *Informatica* 31 (2007) 3–24.
- [45] H. Tsunashima, Condition Monitoring of Railway Tracks from Car-Body Vibration Using a Machine Learning Technique, *Appl. Sci.* 9 (2019) 2734, <https://doi.org/10.3390/app9132734>.
- [46] R. Kulkarni, A. Qazizadeh and M. Berg, Identification of vehicle response features for onboard diagnosis of vehicle running instability, 2022 IEEE International Conference on Prognostics and Health Management (ICPHM), Detroit (Romulus), MI, USA, 2022, pp. 52–57, <https://doi.org/10.1109/ICPHM53196.2022.9815828>.
- [47] EN 13674-1:2011+A1 Railway Applications - Track - Rail Part I: Vignole railway rails 46 kg/m and above. 2017.
- [48] EN 13715:2020 - Railway applications - Wheelsets and bogies - Wheels - Trade profile. 2020.
- [49] R. Kulkarni, et al., Monitoring of Alignment Level (AL) and Cross Level (CL) Track Geometry Irregularities from Onboard Vehicle Dynamics Measurements Using Probabilistic Fault Classifier, in: A. Orlova, D. Cole (Eds.), *Advances in Dynamics of Vehicles on Roads and Tracks II. IAVSD 2021. Lecture Notes in Mechanical Engineering*, Springer, Cham, 2022, https://doi.org/10.1007/978-3-031-07305-2_48.