Wheelflat detection and severity classification using deep learning techniques

Sresakoolchai, Jessada; Kaewunruen, Sakdirat

Wheelflat detection and severity classification using deep learning techniques

Jessada Sresakoolchai [1], Sakdirat Kaewunruen [2,*]

[1] School of Engineering, University of Birmingham, Birmingham B15 2TT, UK; jss814@student.bham.ac.uk
[2] School of Engineering, University of Birmingham, Birmingham B15 2TT, UK; s.kaewunruen@bham.ac.uk
* Correspondence: s.kaewunruen@bham.ac.uk; Tel.: +44 (0) 121 414 2670 (S.K.)

Received: date; Accepted: date; Published: date

Abstract: Wheelflats are one of the most common defects found in railway systems. Wheelflats can result in decreasing passenger comfort and noise if they are slight, or serious incidents such as the derailment if wheelflats are severe. With the increasing demand for railway transportation, the speed and weight of rolling stocks tend to increase, which results in relatively rapid deterioration. Wheelflats are also affected by this increasing demand. To perform preventive maintenance for wheelflats, to keep wheelsets in a proper condition and minimize maintenance cost, the ability to detect and classify wheelflats is required. This study aims to apply deep learning techniques to detect wheelflats and classify wheelflat severity. Deep learning techniques used in this study are Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN). 1,608 samples are used to develop machine learning models. The models are evaluated in 3 different aspects consisting of the overall accuracy, the ability to detect wheelflats, and the ability to classify wheelflat severity. Results from the study show that DNN has the highest overall accuracy of 96%. In addition, DNN can be used to detect wheelflats with 100% accuracy. CNN performs better than RNN in overall accuracy and wheelflat detection. However, RNN has better performance than CNN in wheelflat severity classification. Therefore, DNN is the best approach for detecting wheelflats and classifying severity.

Keywords: Wheelflat Detection; Wheelflat Severity Classification; Machine Learning; Deep Learning; Convolutional Neural Network, Recurrent Neural Network

1. Introduction

Wheelflats are a type of wheel defect mainly caused by sliding and slipping [1]. The sliding may be a result of low adhesion, frozen surface, leaf film, deteriorated brakes, or too high braking force. Wheelflats are common problems that rolling stocks suffer especially in autumn and winter, which significantly increases the chance of leaf film [2] and slip. They can potentially cause service delay or even rolling stocks needing to be removed from service for maintenance.

Wheelflat occurs at the location where a wheel stops rotating and is dragged along the rail. When wheelflats occur, an impact between the wheel and the rail repeatedly happens when the rolling stocks run. This also affects passenger comfort and railway acoustics. From the impact, a sharp corner of wheelflat is worn and makes it more difficult to be detected by humans.

If a wheelflat is detected quickly when its size and severity are small, it can be removed by the wheel turning process before it causes a serious incident. Although small wheelflats are less serious than larger ones, they can cause a high impact even if the speed of the vehicle is low [3]. The impact of rolling stock can damage both wheels and rails by increasing other defects. For a large wheelflat, the wheel turning process may not be sufficient to treat this issue and the wheel needs to be replaced. In the worst case, wheelflats are large and not detected, and they can damage the track, break the rails, or even cause a derailment. Therefore, both small and large wheelflats should be managed properly.

Different techniques can be used to detect wheelflats, including ultrasound, impact force measurement, wheel radius variation measurement, and flaw detection [4]. In this study, the aim is to develop predictive models to detect wheelflats and classify the severity of wheelflats by using machine learning techniques. It is expected that the developed predictive model can detect wheelflats and classify their severity without an in-depth understanding of wheelflat mechanisms and subjective issues. The study uses forces between wheels and rails as a key parameter to detect and classify wheelflats, which can be collected even though rolling stocks run at full speed. This will save time and costs for detailed investigation.

The objectives of this study are: 1) to develop predictive models for detecting wheelflats and classifying their severity by using deep learning techniques consisting of deep neural networks (DNN), 1D-convolutional neural networks (1D-CNN), and recurrent neural networks (RNN) and 2) to compare the performance of each predictive model by comparing the accuracy. The scope of this study will focus on wheelflats, while there are no other defects in both wheels and rails. Data is obtained by reliable simulations for numerical data.

The key contribution of this study can be summarized as follows;

- This study presents the potential for applying deep learning techniques to detect and classify wheelflats and their severity of which there is no previous study. The detection and classification of wheelflats tend to be a difficult task without a detailed diagnosis. Wheelflats may be able to be detected by passenger comfort when the size of wheelflats is large. However, with the advanced suspension system of rolling stocks in the present, detecting small wheelflats without a tool is difficult.
- DNN is used to develop a predictive model. Features of the prediction are carefully selected to ensure that they are significantly related to wheelflat detection and classification. This study demonstrates which features should be used to detect or classify wheelflats.
- CNN is used to present the accuracy of the prediction when the machine detects and classifies wheelflats by itself without feature engineering by humans. The accuracy of the CNN model will demonstrate the

performance of machine learning in capturing the pattern in the data and using the pattern for the prediction.

- RNN, which is normally used for sentence or speech recognition, is applied in this study. This study will show whether RNN is appropriate for engineering prediction when compared to other deep learning techniques.
- The predictive models developed in this study can be used for the maintenance plan of railway systems to manage wheelflats efficiently. The maintenance plan can be developed depending on risk and cost management.

This study is presented as follows; the literature review and the research gap are presented in Section 2 and the background of deep learning techniques used in this study is presented in Section 3. Section 4 presents the data used in this study to develop predictive models and related software. Techniques and model structures are presented in Section 5. Results from the model development and test are presented in Section 6. The discussion and conclusion are presented in Section 7 and 8 respectively.

2. Literature review

Wheelflats are a common and serious problem in railway systems. Various techniques are used to study their effect on railway systems. Dong et al. [5] applied finite element to determine dynamic force between wheels and rails and strains. Wheelflats were included in the study to study how they affected dynamic forces. They found that the axle load and speed were critical parameters which affected dynamic forces. From the increasing demand for railway transportation, the axle load and speed of rolling stocks tend to increase according to the demand which results in more severe wheelflats. To prevent serious incidents and perform preventive maintenance, a capability to detect wheelflat and classify their severity is required.

To detect wheelflats, various approaches are used. An acceleration was basically used to detect wheelflats by calculating energy and Fourier transform [6]. It showed that acceleration could identify wheelflats. Yue et al. [7] used wavelet transform to detect wheelflats and mainly used vibration as an indicator to detect wheelflat. From their study, it is apparent that the use of time-domains might lead to mistaken detection. They claimed that using data based on FFT (Fast Fourier transform) was more precise than time-domains especially when peak values were used. Belotti et al. [8] also applied wavelet signals to detect wheelflats. They also found that using frequency-domains was more precise. They set a speed range of 10-100 km/h. Using their approach, wheelflats can be detected with 94% accuracy. A tool to collect frequency-domain data could be Fiber Bragg Gratings installed at tracks [9]. Ultrasound was another approach used to detect wheelflats although it required time and money to do [4].

Although vibrations or accelerations are commonly used to detect wheelflats, there was no efficient processing tool to analyze collected data [10]. Various studies have proposed methods to interpret the data. Gao et al. [11] applied neural networks and genetic algorithms to detect wheelflats. The main parameter in their study is the mean of noise. They claimed that the proposed technique could improve the accuracy of detection by 10% compared to the traditional methods used in China. However, their study used noise and speed as the main parameters meaning models of rolling stocks could affect the performance of the detection. Li et al. [12] state that although vibration was mainly used to detect wheelflats, the data was usually contaminated with other noises such as background noise and deteriorated track structure. Therefore, they used adaptive multiscale morphological filtering (AMMF) to detect wheelflats and found the result was satisfying. Krummenacher et al. [13] applied machine learning techniques to detect wheelflats. They used a wagon as the case study. Vertical forces were used as the main parameter. The techniques they used are support vector machine and neural networks. They applied 2D-convolutional neural networks with time-domain data. In their study, the accuracy of the prediction was 81-92%. Deep learning techniques were used for various purposes. For example, they are used for automating inspection in industries [14]. Deep learning has the potential to be applied in railway systems for defect detection, which this study will conduct.

From the literature review, it is apparent that various techniques have used to detect wheelflats. However, studies on wheelflat severity classification are still limited. In addition, features used to detect and classify wheelflats are not highly varied. In terms of raw data, frequency-domain data has not been used for neural networks which expected to provide a satisfying outcome. It can be seen that the use of deep learning techniques to detect and classify wheelflats is new in this field. This study aims to fill this research gap and demonstrate the potential of deep learning in railway systems.

3. Background

This study applies deep learning techniques to develop predictive models for detecting and classifying wheelflat and its severity. In this case, deep neural network (DNN), convolutional neural network (CNN), and recurrent neural networks (RNN) are used. For DNN, the characteristics of the model are similar to traditional neural networks. The details of CNN and RNN are demonstrated in this section.

3.1. Feature extraction

To develop predictive models, some features need to be fed into models if DNN is used. In other words, feature selection is made by humans. However, if raw data is used, CNN can be used to extract features from the data by machines or unsupervised learning [15]. Therefore, no knowledge is required to do feature extraction and it is not affected by cognitive bias. This is one of the main benefits of using CNN for classification. To conduct feature extraction, components of CNN have their own particular roles.

Like other neural networks, weights and biases are adjusted during the training process. Weights are related to features. To identify features, local dependencies of the data need to be detected. A result of feature extraction is feature mapping. For CNN, detected features are called filters. The number of features can be adjusted by filters. This is done in a convolutional layer. In a convolutional layer, there are moving windows (called kernels) which move along the data to extract features. The number of features is directly related to the number of kernels because a kernel can represent a feature. The size of the kernels can be adjusted and identifies the number of data used to calculate representatives. A set of data is calculated by an activation function.

A pooling layer is an important layer in CNN. The concept is although an image is smaller, humans can still classify the image and machines are likely to be able to do as well. The pooling layer can be placed after a convolutional layer to maintain significant data. Advantages of the pooling layer include reducing the sensitivity of the output from the convolutional layer and reducing the number of parameters required to be trained because the pooling layer will extract only significant data for the prediction. As a result, the training process will be faster when using pooling layers. One of the most notable pooling layers is a max pooling layer which extracts a maximum data in a set of data or a kernel as a representative, and will be used in this study. As convolutional layers, the size of kernels in max pooling layers can be adjusted. The output from a pooling layer is shown in **Figure 1**.

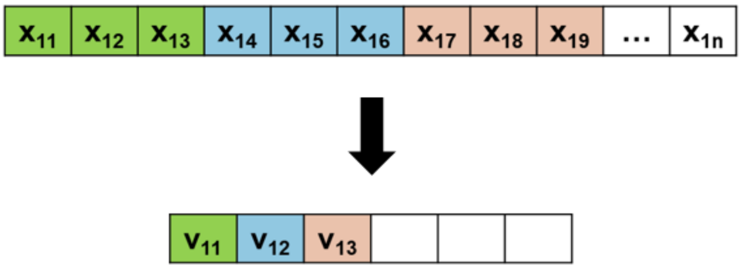


Figure 1. The output from a pooling layer

Convolutional layers can be used with or without pooling layers depending on problems or performance during model tuning. One part of the predictive model containing convolutional layers and pooling layers is feature extraction which is one of two parts of CNN models. Another part of the CNN models is the classification part which contains dense layers or fully-connected layers. To connect the feature extraction and classification parts, a flatten layer is used to transform nD-array to 1D-array to feed the input into dense layers. This process is called flattening as shown in **Figure 2**.

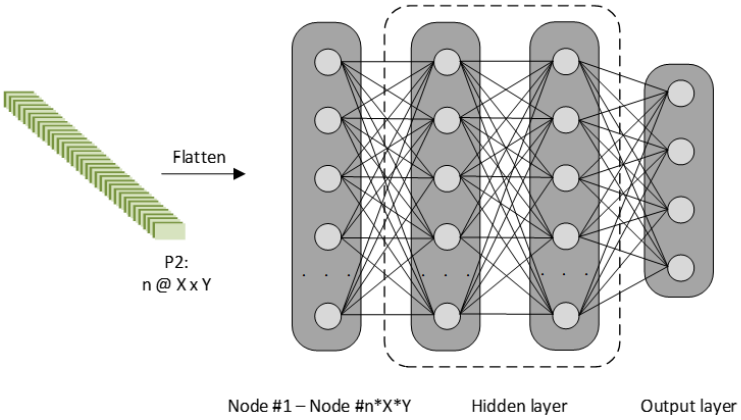


Figure 2. The flattening process and the classification part

One of the most serious weaknesses of machine learning models is overfitting. Overfitting occurs when some features are more significant compared to other features. In other words, the model tends to memorize the output instead of predict. To prevent overfitting, different models can be used to predict and outputs are used to do the prediction again. However, using many models is time-consuming and resource-consuming. Dropout is a concept to prevent overfitting. A function of dropout is that some features are randomly ignored during the training process. When relatively significant features are ignored, other features need to be used to make the prediction instead to maintain the performance of the model. Therefore, there is no feature that mainly used to predict. Dropout can improve the performance of the model through regularization or making the model general enough to handle diverse data.

3.2. Long short-term memory

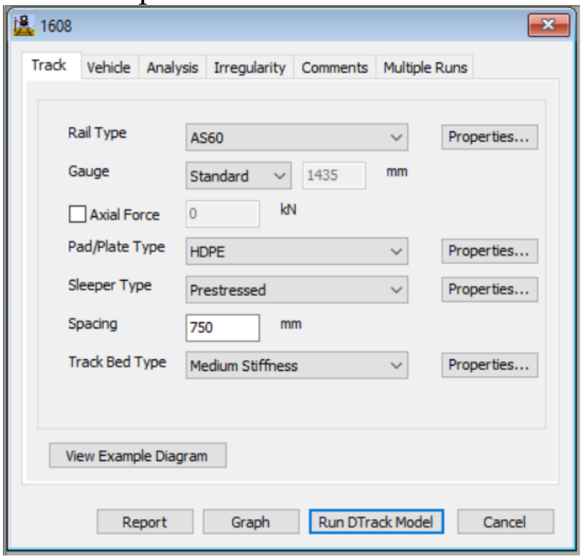
The main characteristic of RNN is it has memory units in a model. In contrast to other neural networks which features are independent, features in RNN are related to each other. At the same time, RNN can remember these relationships during the training process. Therefore, it is believed that RNN can provide better performance when used with sequential data. The sequences of data are also important to RNN because the model is trained in loops.

One of the noteworthy RNN is a concept of long short-term memory (LSTM) is used. LSTM consists of 3 gates: the input gate, forget gate, and output gate. Because RNN can memorize everything fed into the model, it can suffer from a large amount of data due to long processing time and unnecessary data. The forget gate is used to determine how much data should be memorized and removed from the model. The input gate is used to update data when the model finds significant features which will be used further. The output gate is used to determine the output which will be sent to the next stage of training. From these 3 gates, in contrast to the traditional neural network, the same input can result in different outputs depending on the previous input in the sequential data. Like CNN, the main benefit of RNN with sequential data is that it does not require knowledge or expertise to extract features because machines will do this by pattern recognition. RNN also has the capability to be fed multiple parallel sequential data. Therefore, it can be applied to predict wheelflats when sequential data from 2 wheels is used to train the model.

4. Data and Experiment

4.1. D-Track

This study uses the dynamic behavior of wheels and rails as its case study. D-track simulation is applied to simulate the dynamic behavior and generate numerical data. D-track was developed in 1994 [16]. The user interface of the simulation was developed and a validation of accuracy was conducted in 2005 [17]. However, there were issues with accuracy [18]. Then, D-track was revised to improve accuracy. From the revision, the error from the simulation was reduced to less than 15% [19]. Results from previous simulations by D-track are reliable and can be used as benchmarks for this study. The window of D-Track is shown in **Figure 3**. The software is flexible enough to adjust track profile and properties, vehicle properties, analysis, and irregularity. For the track, the rails and substructure properties can be adjusted, including the type of rail, sleeper type and spacing, and track bed properties. For the vehicle, the detail of vehicle, bogie, and wheel can be adjusted, including speed and diameter of wheels. Meanwhile, the location of the analysis can be selected. In this study, the analysis is done every 0.0005 seconds. Therefore, the results from simulations are reported on a 2,000 Hz basis. The irregularity can be input in the software both rail and wheel irregularity. In this study, wheelflats are the only irregularity analyzed. The length and depth of wheelflats can be set in this window. The output of the software can be exported in the form of reports and graphs. Different forms of information are exported. In this study, the information is exported as reports. The information available is acceleration, force, moment and shear at rails and sleepers, and displacement. In this study, accelerations are calculated from forces and mainly used for model development. Further detail is presented in Section 4.2.



Rail Type AS60
Gauge Standard 1435 mm
□ Axial Force 0 kN
Pad/Plate Type HDPE
Sleeper Type Prestressed
Spacing 750 mm
Track Bed Type Medium Stiffness

Figure 3. D-Track simulation software

4.2. Data

Simulations in this study are conducted by using D-track. Different parameters are varied to create data diversity and simulate the dynamic behavior of wheels and rails in different situations. At the same time, some of these varied parameters will be used as features in model development.

The varied parameters consist of sleeper spacing, which ranges from 600 to 750 mm, speed of the vehicle which ranges from 20 to 200 km/hr, and the weight of vehicle which ranges from 40-80 tons, and size of wheelflats. Locations of wheelflats are assumed to be at sleepers and between sleepers. For the size of wheelflats, the relationship between the depth and length of the wheelflat can be explained by Equation (1), where R is the radius of wheels (mm), d is the depth of wheelflat (mm), and L is the length of wheelflat (mm). In the simulations, the function of the wheelflat when a vehicle runs along the track is explained in Equation (2) where x is the position of the vehicle related to the track.

$$R^2 = (R-d)^2 + \left(\frac{L}{2}\right)^2 \tag{1}$$

$$f(x) = \frac{d}{2} * \left[1 - cos\left[\frac{2\pi x}{L}\right]\right] \tag{2}$$

In this study, the sizes of wheelflats are set at 4 levels based on the depth and length of wheelflat which are level 0: no wheelflat, level 1: depth of 0.1 mm and length of 19.18 mm, level 2: depth of 0.55 mm and length of 44.98 mm, and level 3: depth of 1 mm and length of 60.63 mm when the radius of wheels is 460 mm. These numbers are set based on previous studies which mention that wheelflats should be limited to a length of 52-60 mm and a depth of 0.9-1.0 mm [20, 21]. Therefore, this study uses this criterion as the maximum value of wheelflat deflect and the less severe levels use lower numbers. Other parameters related to the simulations are fixed. 1,608 simulations are run to generate numerical data which can be categorized into 4 groups: level 0-3. The number of samples in each group is 888 at level 0 and 240 samples each at levels 1-3. The summary of parameters for simulation is shown in **Table 1.**

Table 1. Summary of varied parameters for simulations

Parameter	Variation	Step of variation
Sleeper spacing	600-750 mm	50 mm
Speed of vehicle	20-200 km/h	5 km/h
Weight of vehicle	40-80 tons	20 tons
Location of wheelflat	Above sleepers and mid-span between sleepers	N/A
Wheelflat depth	0, 0.1, 0.55, and 1 mm	Related to wheelflat length
Wheelflat length	0, 19.18, 44.98, and 60.63 mm	Related to wheelflat depth

This study uses accelerations at the axle box as the main outcomes of the simulation. The accelerations at the axle box are also used as features for the predictive models. However, results from simulations are forces which need to be calculated for accelerations. An example of forces at the wheel-rail contact is shown in **Figure 4**. In this figure, there are 3 peak values and 3 bottom values. These values will be used as features for the prediction. This process requires data filtering to detect data which is significantly different between samples that the severities are different. From **Figure 4**, forces from samples with wheelflat are presented. Peak and bottom values will be different depending on the size of wheelflat. For samples without wheelflats, peak and bottom values are quite constant because the occurring forces are very smooth which is different to samples with wheelflats where there are impacts from wheel defects. To obtain features for the prediction, data needs to be filtered. The main differences between samples with and without wheelflats or samples with different sizes of wheelflat are the peak and bottom values. Data filtering is used to obtain peak and bottom values. At the same time, data filtering is used to remove unnecessary data which has the same characteristics in samples with and without wheelflats. Data filtering is done by using Visual Basic for Applications (VBA) for automated data preprocessing. Please note the data filtering is used for the DNN model only because CNN and RNN models use raw data as an input. An example of data with different parameters is shown in **Figure 5**.

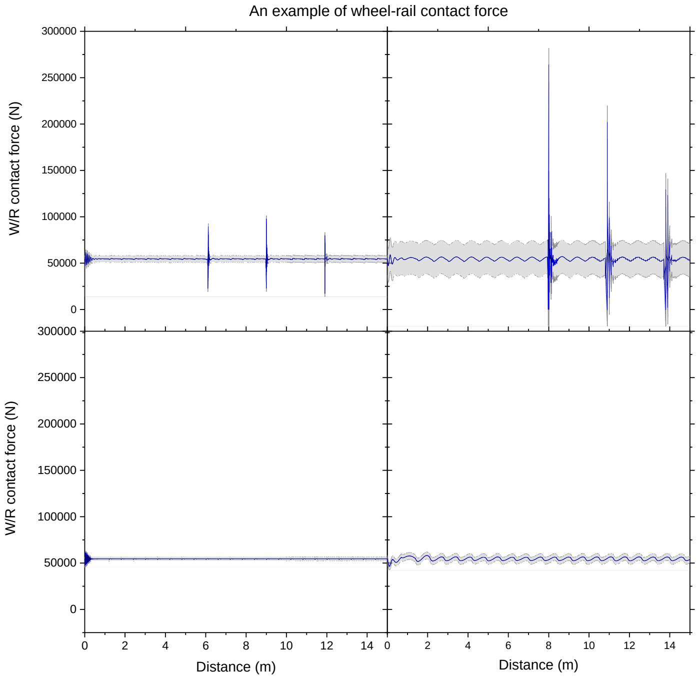


Figure 4. An example of forces at the wheel-rail contact

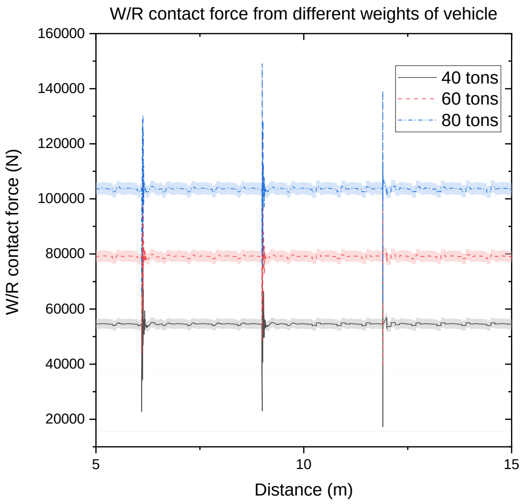
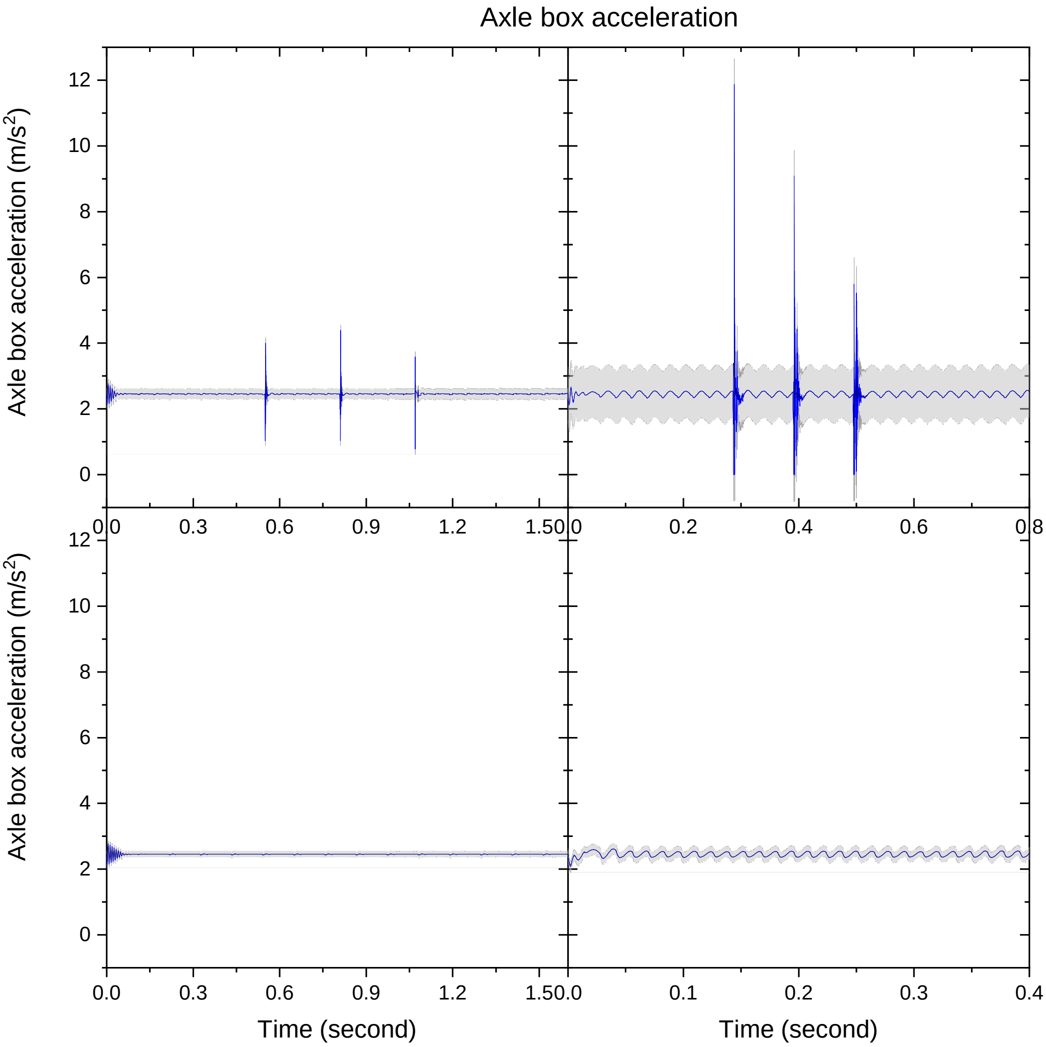


Figure 5. An example of samples with different weights of rolling stocks

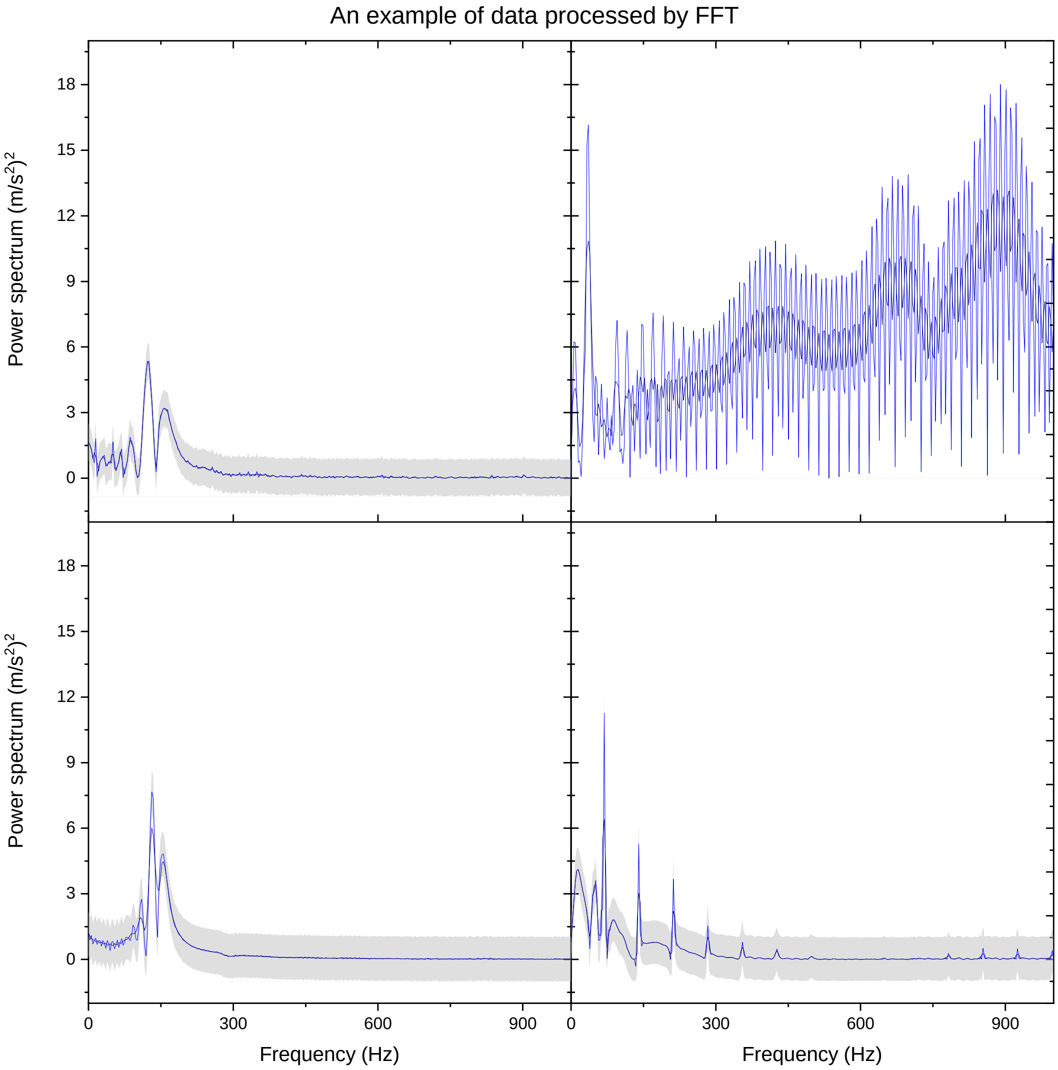
To develop predictive models, data is used in 2 ways: simplified data and raw data. Simplified data is used as features for DNN. Raw data for axle box accelerations is used as features for CNN and RNN. To clarify, for DNN, the parameters which are used as features for the prediction are sleeper spacing, speed of vehicle, the weight of vehicle, 3 peak values and 3 bottom values of axle box accelerations. Therefore, there are 9 features for DNN as simplified data and 1 label for the prediction.

As mentioned in Section 3, the main benefit of CNN is it can extract significant features through machines without human expertise. The main benefit of RNN is it can classify groups of data by sequential data. Therefore, raw data can be used in case of CNN and RNN. However, raw data cannot be used directly because the dimensions of data are different due to the speed of vehicle. From the software, the distance of simulations is fixed when the analysis is done every 0.0005 seconds. Therefore, time-domain data cannot be used directly because the duration of each simulation is different. In addition, as mentioned in Section 2, using frequency-domain data provides more accuracy than using time-domain data. Therefore, raw data is transformed by using FFT to make data in the same dimension and improve the accuracy of the prediction. An example of results of data processed by FFT is shown in



(a)

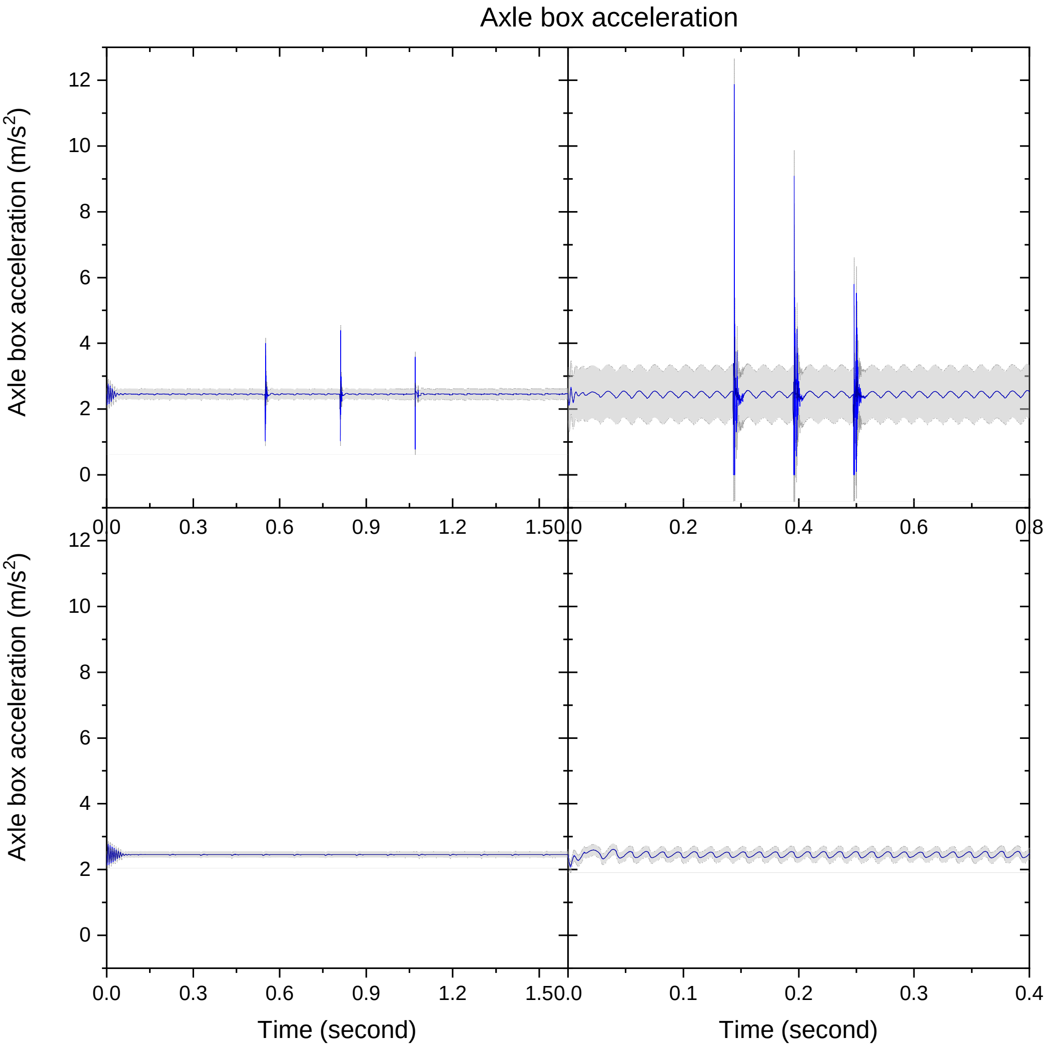
Figure 6. An example of processed data by FFT: (a) Axle box acceleration; (b) Axle box acceleration processed by FFT



(b)

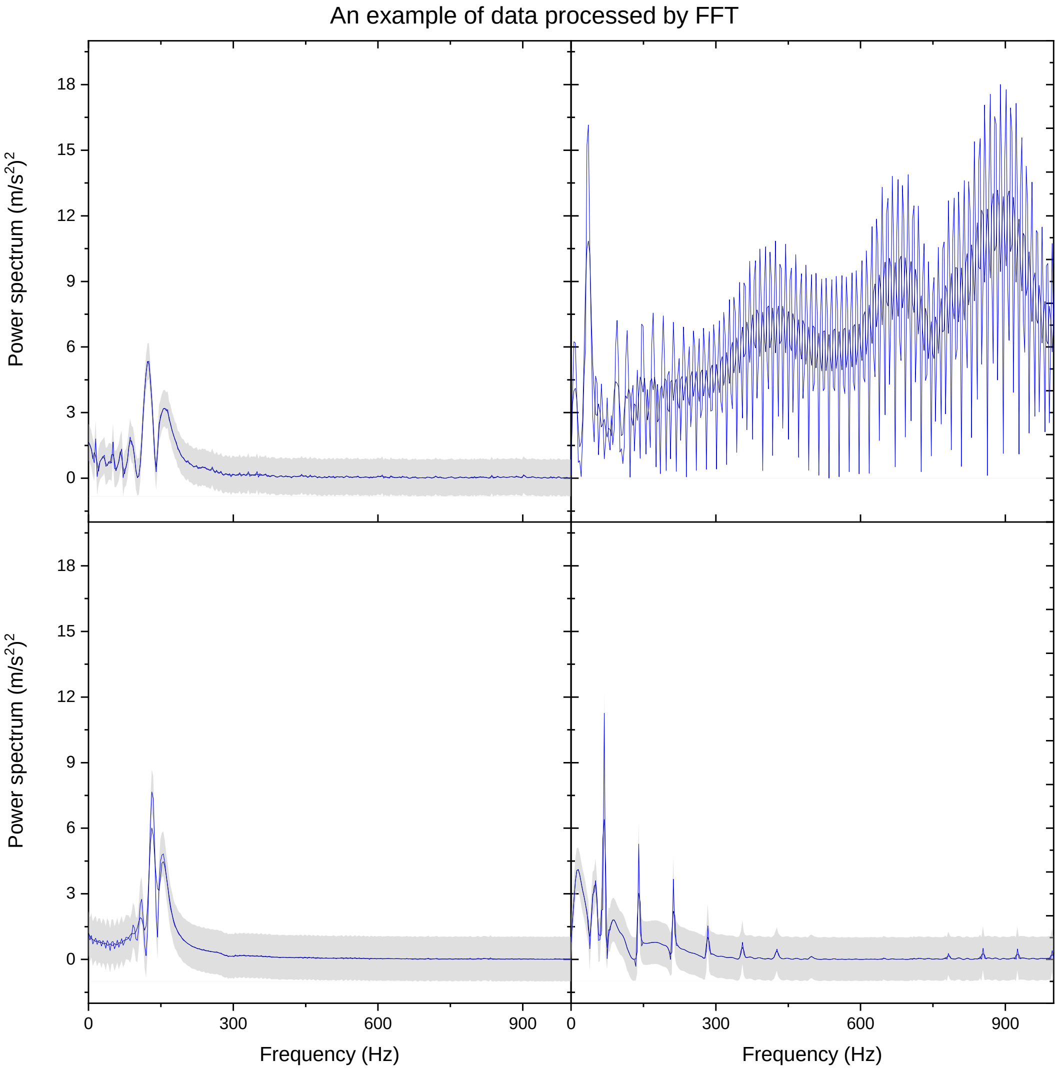
Figure 7. In this study, raw data consists of 2 sets of data due to 2 axles where accelerations are measured. These 2 sets of raw data are transformed and used to detect wheelflats and classify wheelflat severity in CNN and RNN predictive models. The software analysis is done every 0.0005 seconds so the output data is 2,000 Hz data. When data is processed by FFT, a two-sided spectrum is the result. However, these 2 sides are symmetrical. Therefore, the only single-sided spectrum

is presented in



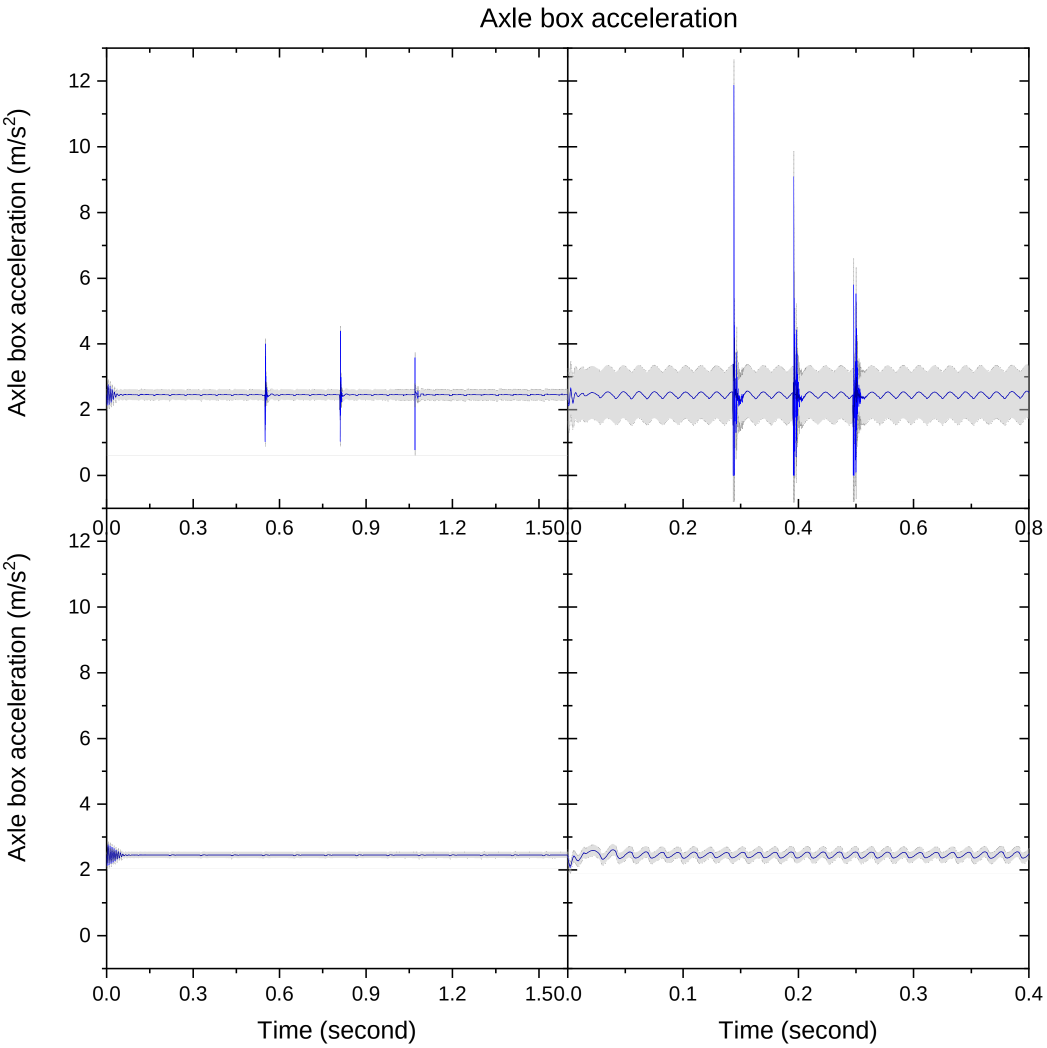
(a)

Figure 6. An example of processed data by FFT: (a) Axle box acceleration; (b) Axle box acceleration processed by FFT



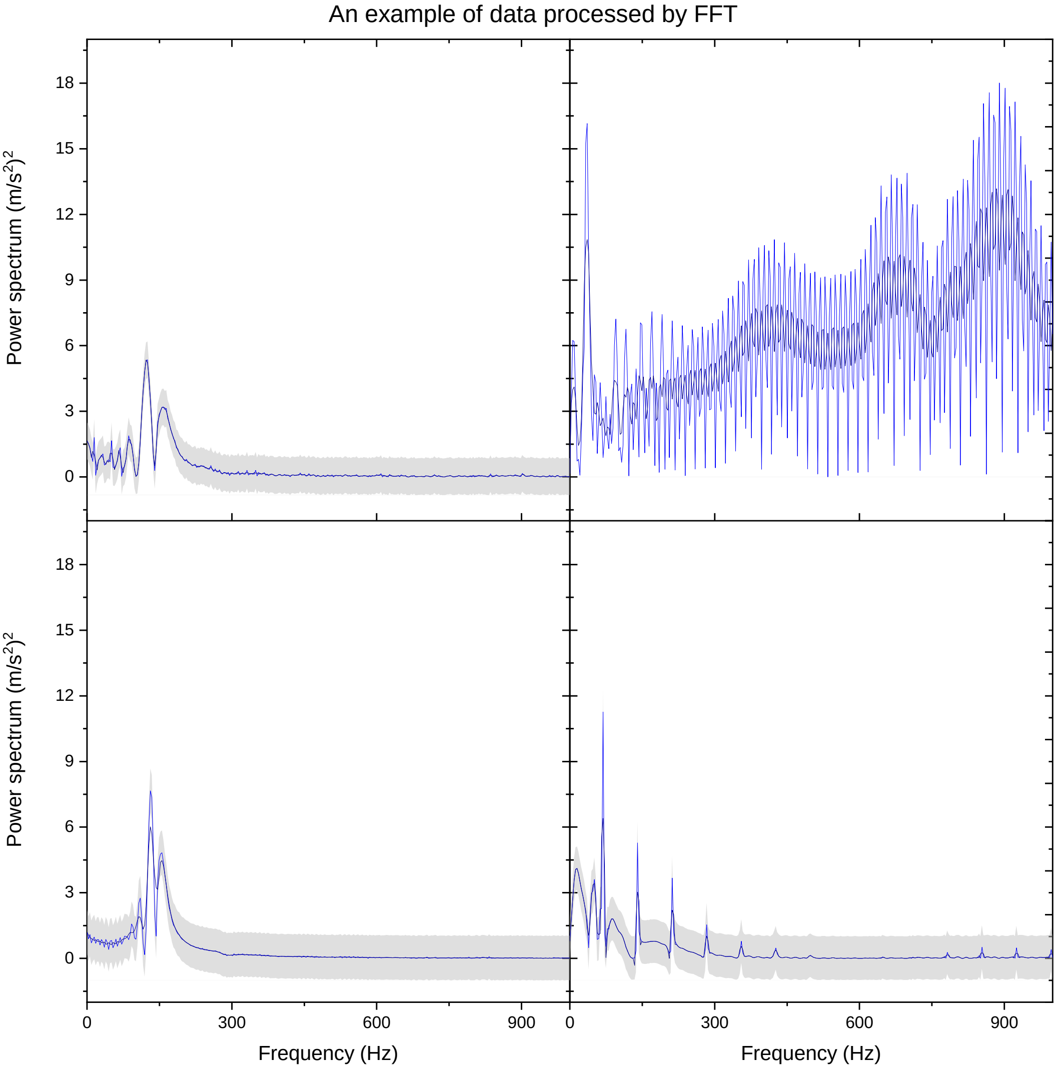
(b)

Figure 7 or the range of frequency is 0-1,000 Hz. In this figure, it can be seen that the peak power occurs when the frequency ranges from 0-200 Hz.



(a)

Figure 6. An example of processed data by FFT: (a) Axle box acceleration; (b) Axle box acceleration processed by FFT



(b)

Figure 7. An example of processed data by FFT: (a) Axle box acceleration; (b) Axle box acceleration processed by FFT

In summary, there are 2 groups of data used in this study. The first group is simplified data which consists of varied parameters from simulations and axle box accelerations from simulations. For this group, there are 9 features and 1 label for developing a DNN predictive model. The second group is raw data processed using FFT. This group of data is used to develop CNN and RNN predictive models. In 1,608 simulations, each simulation is used as a sample. 70% of samples are used as training data and 30% of samples are used as testing data. The performance is measured by using accuracy as the main criterion. Other criteria are also used to demonstrate the performance of predictive models such as precision and recall.

5. Model structures

When data is ready, model development is conducted. This study applies DNN, CNN, and RNN as deep learning techniques for classification. Data is processed to be appropriate for each technique as mentioned in Section 4.2. The next step is predictive model structure design. In Section 3.1 and 3.2, the main components from each technique were presented. Those components are the main parts of each model structure developed by each technique. However, model tuning is done while the model structures are adjusted to ensure that the accuracy from each model is the best result that the models can deliver. These model structures for each technique are presented in the next section.

5.1. DNN

The structure of the DNN model is shown in **Figure 8**. The number of hidden layers and hidden nodes are adjusted to improve accuracy. The structure consists of 1 hidden layer and the number of hidden nodes is 14. The number of nodes in the input layer and output layer is 9 and 4 respectively according to the number of features and classes. The activation function used in the hidden layer is ReLu (Rectified Linear Unit) which is 0 when the input is less than 0 and has the same value as the input when the number is more than 0. The activation function in the output layer is Softmax which returns the probability distribution or the probability of each class. The summed probability is 1 and the class with the maximum probability is selected as the predicted class.

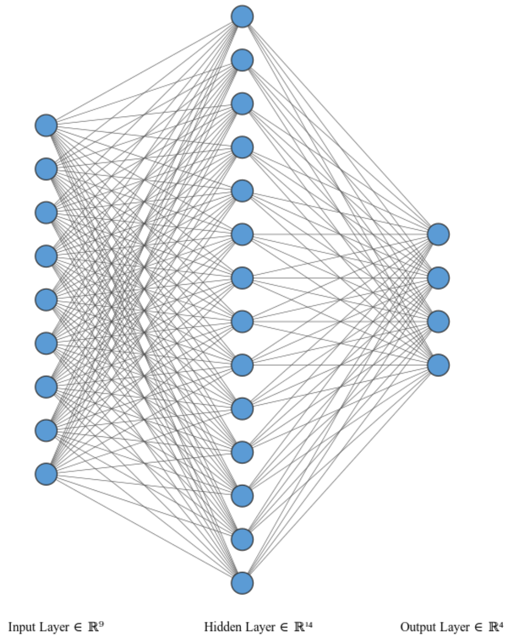

Input Layer ∈ $\mathbb{R}^9$
Hidden Layer ∈ $\mathbb{R}^{14}$
Output Layer ∈ $\mathbb{R}^4$

Figure 8. DNN structure

5.2. CNN

The structure of the CNN model is shown in **Figure 9**. The structure consists of the feature extraction part and the classification part. For the feature extraction part, 2 pairs of convolutional layers and max pooling layers are applied. A dropout layer is placed between the 2nd convolutional layer and the 2nd max pooling layer to prevent overfitting. The proportion of dropout is 50%. Then, data is passed from the feature extraction part to the classification part by a flatten layer. The classification part consists of 3 dense layers. 2 of these are used as hidden layers where the number of nodes is 100. The last dense layer is the output layer. For the feature extraction part, the number of filters in the 1st and 2nd convolutional layers is 32 and 64 respectively. For the max pooling layers, the pooling size is 2 or the size of data is reduced by 50% to reduce the complication in the structure and the training time.

From **Figure 9**, it can be seen that the input is 1D data with 2 channels. Each channel contains the axle box accelerations from a wheel processed by FFT. The dimension of the input is 1x670. After that, the input is passed through the first convolutional layer with 32 feature maps. The kernel size is 3 and the stride is 1. The activation function used in every convolutional layer is ReLu. After that, the data is subsampled by the first pooling layer. In this study, the max pooling technique is used. From the pooling layer, the data dimensions are reduced by half because the pooling size is 2 and the stride is 2. Next, the data is processed to the second convolutional layer. In this convolutional layer, the kernel size and stride are the same as the first convolutional layer. However, 64 feature maps are used to conduct feature extraction. The data is subsampled again in the second pooling layer. There is a dropout layer between the second convolutional and pooling layer to prevent overfitting and make the model perform better. These 5 layers make up the feature extraction. To pass the data to the classification part, a flatten layer is used. The flatten layer will rearrange the data in the form of a 1D-array. Each point of data is fully connected to the classification part. The classification part consists of 3 dense layers. 2 of these function as hidden layers with 100 hidden nodes and the other is the output layer with 4 nodes equal to the number of classes. The activation function in the last dense layer is Softmax, which classifies the classes.

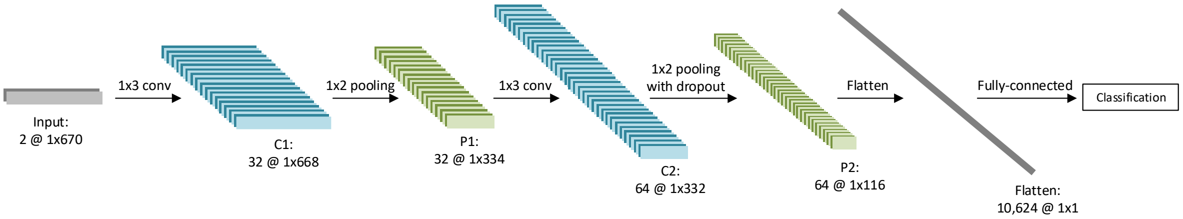


Figure 9. CNN structure

5.3. RNN

The structure of the RNN model is shown in **Figure 10**. The structure consists of an LSTM layer connected to the other 3 dense layers. The first 2 dense layers are used as hidden layers when the number of nodes is 100. The last dense layer is the output layer.

Like the CNN model, the input of the RNN model is 2 sets of raw data which are processed by FFT. The time steps of the number of values in each data are 670. After that, the input is passed through the LSTM layer consisting of 200 LSTM cells.

From **Figure 10**, it can be seen that a value is processed through every LSTM cell. For example, the first value is passed through the first LSTM cell. Then, it is passed through the second LSTM cell and so on. By doing this, each value is not independent of other values but it also affects the following values. This process is used to deal with the sequential data to detect the pattern and do classification. The activation function in the LSTM layer is ReLu. After that, the LSTM layer is connected with the classification part as the CNN model. In this case, the classification part consists of 3 dense or fully-connected layers. The activation function in dense layers is ReLu except in the last dense layer which uses Softmax.

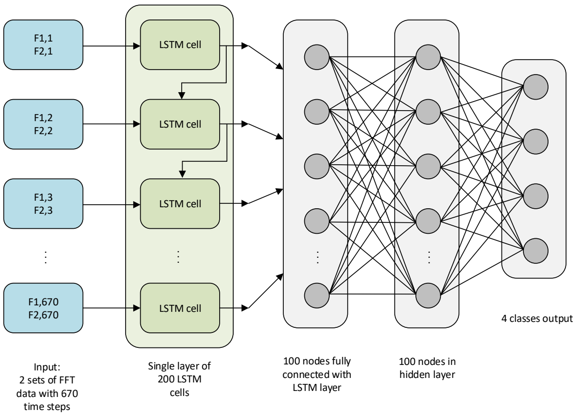


Figure 10. RNN structure

6. Results

From the model structures shown in Section 5, the number of nodes in the last layer is 4. This is because there are 4 classes mentioned in Section 4.2. In other words, the classes for wheelflat detection and wheelflat severity classification are combined together through level 0-3. This section will present the results of the overall accuracy by considering 4 classes together in Section 6.1. The performance in detection will be presented by considering level 0 and level 1-3 in Section 6.2. Last, the performance of the models in wheelflat severity classification will be presented by considering level 1-3 in Section 6.3.

6.1. Overall accuracy

The samples are separated into training data and testing data with proportions of 70/30 respectively. During the training process, the accuracy of the training data is increased to a certain value. In this case, it means the more epoch or training time does not result in higher accuracy. Therefore, the accuracy of the testing data is focused on. The accuracy is shown in Figure 11. From the figure it can be seen that accuracy can be improved when epoch is less than 30. After that, the more epochs cannot improve the overall accuracy. In addition, the accuracy varies, which is the nature of the machine learning model training where samples are selected to be trained and tested randomly. Figure 11 shows that the models with the highest accuracy are DNN.

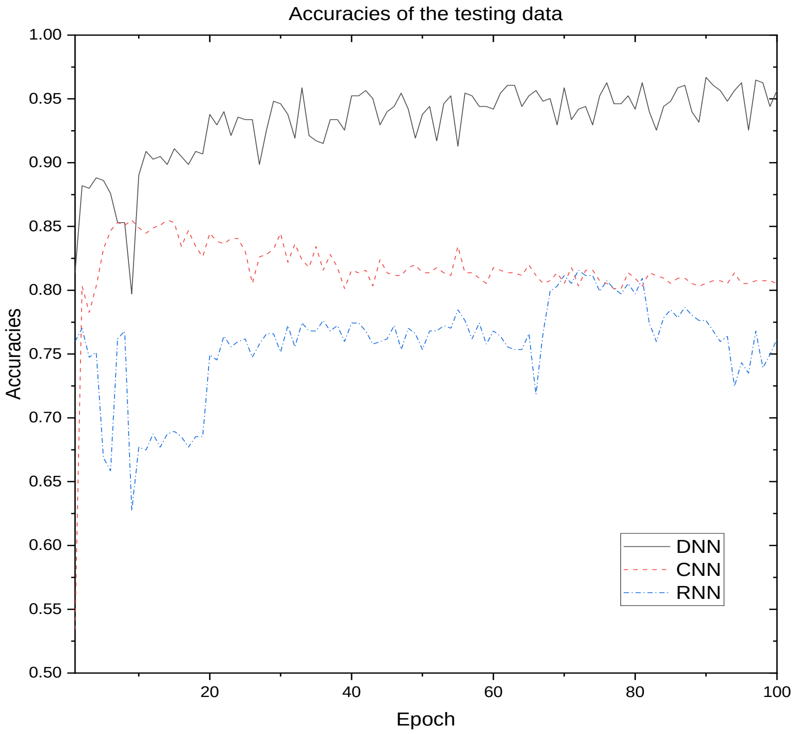


Figure 11. Accuracy of the testing data

To summarize the overall performance of the models, accuracy, precision, and recall are used. The accuracy shows the proportion of correct predictions to the total number of testing data. Therefore, one model has one value of accuracy. The precision presents the proportion of correct predictions to the total number of predictions in an interesting class. In other words, precision shows how far the prediction is reliable. The recall presents the proportion of correct predictions to the total number of true labels in an interesting class. In other words, the recall shows how well a model can detect samples. In contrast to the accuracy, precision and recall have different values depending on classes. Therefore, each class has its own precision and recall. Results from the model development are shown by the confusion matrix. The confusion matrix presents the relationship between predictions and true labels. Values in the confusion matrix are calculated for accuracy, precision, and recall. A summary of accuracy, precision, and recall is shown in **Table 2**.

Table 2. Overall accuracy, precision, and recall of predictive models

Class	Wheelflat depth (mm)	Wheelflat length (mm)	Accuracy			Precision			Recall		
			DNN	CNN	RNN	DNN	CNN	RNN	DNN	CNN	RNN
Level 0	0.00	0.00	0.96	0.80	0.77	1.00	0.94	0.73	0.99	0.91	1.00
Level 1	0.10	19.18				0.97	0.57	1.00	0.99	0.70	0.59
Level 2	0.55	44.98				0.88	0.77	0.80	0.84	0.55	0.53
Level 3	1.00	60.63				0.86	0.54	0.76	0.91	0.69	0.45

6.2. Accuracy of detection

For the performance of predictive models in wheelflat detection, wheelflat levels 1-3 are combined together to be compared with level 0. In this case, level 0 is called "without wheelflat" and level 1-3 is called "with wheelflat". The accuracy, precision, and recall of the wheelflat detection are shown in **Table 3**.

Table 3. Accuracy, precision, and recall of wheelflat detection

Class	Accuracy			Precision			Recall		
	DNN	CNN	RNN	DNN	CNN	RNN	DNN	CNN	RNN
Without wheelflat	1.00	0.91	0.81	1.00	0.94	0.73	0.99	0.91	1.00
With wheelflat				0.99	0.88	1.00	1.00	0.91	0.62

6.3. Accuracy of classification

For the performance of predictive models on wheelflat severity classification, wheelflat level 1-3 are considered while level 0 is ignored. The wheelflat depth and length are shown in **Table 2**. In this case, wheelflat level 1-3 are called light, intermediate, and severe wheelflats respectively. The accuracy, precision, and recall of the wheelflat detection are shown in **Table 4**.

Table 4. Accuracy, precision, and recall of wheelflat severity classification

Class	Accuracy			Precision			Recall		
	DNN	CNN	RNN	DNN	CNN	RNN	DNN	CNN	RNN
Light wheelflat	0.91	0.70	0.84	1.00	0.67	1.00	0.99	0.75	1.00
Intermediate wheelflat				0.88	0.84	0.80	0.84	0.60	0.81
Severe wheelflat				0.86	0.63	0.76	0.91	0.78	0.74

In terms of the training time, DNN and CNN take less than 0 second to train the whole samples for 1 epoch. For RNN, the training time is significantly higher than DNN and CNN. The training time of RNN is 11 seconds for 1 epoch.

7. Discussion

This section describes the performance of developed predictive models in different aspects consisting of the overall performance, the performance in wheelflat detection, and the performance in wheelflat severity classification.

7.1. Overall performance of predictive models

From **Table 2**, the DNN model provides the highest accuracy, of 96%, while the accuracy of CNN and RNN models is 80% and 77% respectively. From this, it can be concluded that DNN is the most appropriate technique for developing the predictive model for detecting wheelflats and classifying wheelflat severity. For DNN, 9 features are used as the inputs: sleeper spacing, the weight of vehicle, speed, 3 peak values of axle box accelerations and 3 bottom values of axle box accelerations. These features tend to be directly related to wheelflats because wheelflats result in impact when rolling stocks run. Although CNN is claimed to not require the expertise of feature extraction because the machine will take care of feature extraction itself, using raw data as an input to predict wheelflats provides a lower accuracy than

DNN. Some key features are missing when raw data in the form of axle box accelerations is used, such as speed and weight of vehicles. Therefore, if features which are significantly related to the prediction are known, using DNN can provide a better result. That is why CNN and RNN cannot detect and classify wheelflats as well as DNN. RNN provides the lowest accuracy because it is not designed for this purpose originally. In the first place, RNN is designed for tasks were sequential data is related to each other such as an order of words in a sentence. However, the sequential data of axle box accelerations seems to not be related to other values in the sequence as much as the order of words in a sentence. For precision and recall, DNN also performs well while CNN and RNN have relatively low precision and recall while some values are lower than 50%. That means the predictions from CNN and RNN in some cases are not reliable. For example, the recall of level 3 from the RNN model is 45%, which means the RNN model can detect severe wheelflats only 45% correctly while 55% of severe wheelflats are classified as lower severe wheelflats. This is critical because severe wheelflats can result in serious incidents such as derailment. Another example is the precision of level 3 from CNN which is 54%. This means only 54% of severe wheelflats are predicted correctly while another 46% of the prediction is not of severe wheelflats. This shows the model is not reliable.

7.2. Performance on wheelflat detection

Table 3 presents the performance of wheelflat detection of each model. 2 groups of samples are compared: samples with and without wheelflats. DNN has outstanding performance, with its accuracy, precision, and recall being almost 100%. Therefore, DNN is the most suitable technique to be used to detect wheelflats. CNN performs better than RNN in wheelflat detection. The accuracy levels of CNN and RNN are 91% and 81% respectively. The precision and recall of CNN are higher than 85%. However, for RNN, some values are significantly lower than other models. For example, the recall of "with wheelflat" class is 62%. There are cases when the precision and recall of RNN are equal to 1. From this, it can be said that RNN fluctuates in its wheelflat detection. From Table 3, it can be seen that if the DNN model detects wheelflats, the reliability is 100%. In conclusion, DNN is the best model for wheelflat detection while RNN is the worst for wheelflat detection.

7.3. Performance on wheelflat severity classification

For wheelflat severity classification, DNN is the best option, with an accuracy level of 91%. RNN performs better than CNN in this case. In terms of precision of detecting light wheelflats DNN and RNN are 1 which means when DNN and RNN classify wheelflats as light, they can be absolutely reliable. It should be noted that, in some cases, precision and recall of CNN are about 60% which are lower than other models. Therefore, RNN is slightly better than CNN for wheelflat severity classification. In reality, DNN is the most appropriate for wheelflat classification because its precision is higher than other models. In addition, light wheelflats can be detected correctly by using DNN before wheelflats become severe. Efficient wheelflat management when the size of wheelflat is small can reduce the subsequent defects of wheels and rails due to high impacts from small wheelflats.

In brief, the DNN model provides the best performance in every aspect. Therefore, the DNN model is the best for wheelflat detection and severity classification. This will lead to an efficient maintenance plan to deal with wheelflats. Predictive models can be applied to detect wheelflats and their severity in different ways by using the axle box acceleration which is practicable and effective. Automated software can be developed to measure real-time accelerations. These can then be fed into the models to defect and classify wheelflats.

8. Conclusions

This study uses numerical data generated by simulations to develop predictive models for wheelflat detection and severity classification. Deep learning techniques are used to develop predictive models. The selected deep learning techniques are DNN, CNN, and RNN. 1,608 simulations were run in this study and used as samples. The ratio between training data and testing data is 70/30. Data from simulations are used in 2 ways, simplified data and raw data. For simplified data, data is processed to contain 9 features and used to develop the DNN model. For raw data, axle box accelerations of 2 wheels are transformed by using FFT to develop CNN and RNN models. From the model development, DNN has the highest accuracy in every aspect. Therefore, to optimize the use of predictive models, DNN should be used in the machine learning model.

This study shows that deep learning techniques have potential in wheelflat detection and severity classification. The developed predictive model provides satisfying performance from the accuracies which higher than 90% in every aspect. Therefore, applying deep learning techniques can be integrated with railway maintenance plans.

Deep learning techniques can also be applied in railway maintenance in other aspects. The predictive models can be improved by including other defects in order to obtain a more comprehensive application. Although developing deep learning models requires data and the investment in the data collection tool, it is certain that maintenance costs in railway systems will be reduced in the long term and the capability of detection and classification will be more reliable because there is no human error in the process. The optimum goal of applying data in organizations is to be data-driven which the railway industry can achieve by applying machine learning and data science.

Author Contributions: Conceptualization, S.K. and J.S.; methodology, J.S.; software, S.K.; validation, J.S.; formal analysis, J.S.; investigation, J.S.; resources, S.K.; data curation, J.S.; writing—original draft preparation, J.S.; writing—review and editing, S.K.; visualization, J.S.; supervision, S.K.; project administration, S.K.;

Acknowledgments: Financial support from European Commission is gratefully acknowledged for H2020-MSCA-RISE Project No. 691135 "RISEN: Rail Infrastructure Systems Engineering Network," which enables a global research network that tackles the grand challenge in railway infrastructure resilience and advanced sensing under extreme events (www.risen2rail.eu).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jergéus, J., et al., *Full-scale railway wheel flat experiments.* Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit, 1999. **213**(1): p. 1-13.
2. Milne, A. *Detecting wheel flats and more*. 2017 [cited 2020 12 June]; Available from: https://www.railengineer.co.uk/2017/10/17/detecting-wheel-flats-and-more/.
3. Iwnicki, S., *Handbook of railway vehicle dynamics*. 2006: CRC press.
4. Brizuela, J., C. Fritsch, and A. Ibáñez, *Railway wheel-flat detection and measurement by ultrasound.* Transportation Research Part C: Emerging Technologies, 2011. **19**(6): p. 975-984.
5. Dong, R.G., S. Sankar, and R.V. Dukkipati, *A Finite Element Model of Railway Track and its Application to the Wheel Flat Problem.* Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit, 1994. **208**(1): p. 61-72.
6. Bracciali, A. and G. Cascini, *Detection of corrugation and wheelflats of railway wheels using energy and cepstrum analysis of rail acceleration.* Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit, 1997. **211**(2): p. 109-116.
7. Yue, J., Z. Qiu, and B. Chen. *Application of wavelet transform to defect detection of wheelflats of railway wheels*. in *6th International Conference on Signal Processing, 2002.* 2002.
8. Belotti, V., et al. *Wavelet signal processing applied to railway wheelflat detection*. in *Proc. XVII IMEKO World Congress Metrology in the Third Millennium, Dubrovnik, Croatia*. 2003.
9. Filograno, M.L., et al., *Wheel Flat Detection in High-Speed Railway Systems Using Fiber Bragg Gratings.* IEEE Sensors Journal, 2013. **13**(12): p. 4808-4816.
10. Jia, S. and M. Dhanasekar, *Detection of Rail Wheel Flats using Wavelet Approaches.* Structural Health Monitoring, 2007. **6**(2): p. 121-131.
11. Gao, R., C. Shang, and H. Jiang, *A fault detection strategy for wheel flat scars with wavelet neural networks and genetic algorithm.* Journal of Xi'an Jiaotong University, 2013. **47**(9): p. 88-91.
12. Li, Y., et al., *Fault detection method for railway wheel flat using an adaptive multiscale morphological filter.* Mechanical Systems and Signal Processing, 2017. **84**: p. 642-658.
13. Krummenacher, G., et al., *Wheel Defect Detection With Machine Learning.* IEEE Transactions on Intelligent Transportation Systems, 2018. **19**(4): p. 1176-1187.
14. Ge, C., et al., *Towards automatic visual inspection: A weakly supervised learning method for industrial applicable object detection.* Computers in Industry, 2020. **121**: p. 103232.
15. Zeng, M., et al. *Convolutional neural networks for human activity recognition using mobile sensors*. in *6th International Conference on Mobile Computing, Applications and Services*. 2014. IEEE.
16. Cai, Z., *Modelling of rail track dynamics and wheel/rail interaction.* 1994.
17. Steffens, D.M., *Identification and development of a model of railway track dynamic behaviour*. 2005, Queensland University of Technology.
18. Kaewunruen, S. and C. Chiengson, *Railway track inspection and maintenance priorities due to dynamic coupling effects of dipped rails and differential track settlements.* Engineering Failure Analysis, 2018. **93**: p. 157-171.
19. Leong, J., *Development of a limit state design methodology for railway track*. 2007, Queensland University of Technology.
20. Zhai, W., *Vehicle–track coupled dynamics: theory and applications*. 2019: Springer Nature.
21. Vyas, N. and A. Gupta, *Modeling rail wheel-flat dynamics*, in *Engineering Asset Management*. 2006, Springer. p. 1222-1231.

22. Jiang, H. and J. Lin, *Fault Diagnosis of Wheel Flat Using Empirical Mode Decomposition-Hilbert Envelope Spectrum.* Mathematical Problems in Engineering, 2018. **2018**: p. 8909031.