# Feature-based intelligent models for optimisation of percussive drilling

Kenneth Omokhagbo Afebu, Yang Liu*, Evangelos Papatheou

*College of Engineering, Mathematics and Physical Sciences, University of Exeter, North Park Rd, Exeter, EX4 4QF, UK*

**Abstract**

As a rotary-percussion system, the vibro-impact drilling (VID) system utilises resonantly induced high frequency periodic impacts alongside existing drill-string rotation to cut through the rock layers. Due to the inhomogeneous nature of underlying rock layers, the system often experiences multi-stability which generates different categories of impact motions as drilling continues dowhnhole. Some impact motions yield better drilling performance in terms of rate of penetration (ROP) and bit life-span when compared to others. As an optimisation strategy, the present study adopts feature-based neural networks including multi-layer perceptron, support vector machine and long short-term memory as intelligent models in categorising impact motions from a one-degree-of-freedom impact oscillator representing the percussive bit-rock impacts of the VID system. This way, high-performance impacts can be easily detected and maintained while undesirable low-performance impacts are well avoided to increase ROP, improve bit lifespan and save cost. In this study, scarce and limited classes of experimental impact data are merged with inexhaustibly simulated impact data to train different network models. By means of cross-validation, the trained networks were tested on separate sets of only simulation and only experimental data. Results show that extracting appropriate features from raw impact data is essential for optimising the performance of each network model. About 42% of the feature-based networks yield accuracies greater than 91% while about 67% yield accuracies greater than 77% on both simulation and experimental impact motion data.

*Keywords:* Vibro-impact drilling; Rotary-percussion; Bit-rock interaction; Impact motions; Multistability; Machine learning.

## 1. Introduction

Rock drilling involves boring holes into rocks using custom arrangement of pipes and bits driven by a load force. It is extensively used in mining, construction, groundwater development and the oil and gas industry. Drilling techniques can be summed-up into two major types including rotary and percussive techniques. The rotary drilling relies on the continuous circular movement of the drill bit and the drilling mud to cut through the rock while the percussive drilling relies on intermittent heavy blows delivered by a hammer head to crush underlying rock. If compared, the impact load accumulated by the falling bit during percussive drilling is much higher and fractures rocks better than the axial loading achieved by its rotation during rotary drilling [1]. Also, the intermittent impact actions of the drill-bit during percussive drilling reduces its total rock-contact time to about 2% of the entire drilling time [2]. However, compared to rotary drilling, material removal in percussive drilling is slower, thus resulting

*Corresponding author

*Email addresses:* ka396@exeter.ac.uk (Kenneth Omokhagbo Afebu), y.liu2@exeter.ac.uk (Yang Liu), e.papatheou@exeter.ac.uk (Evangelos Papatheou)

in re-compaction of previously fractured rock materials by subsequent percussions. In an attempt to combine the aforementioned advantages of both rotary and percussive drilling, rotary-percussive drilling technologies were invented [3]. A more recent invention is the high frequency vibro-impact drilling (VID) [4] also referred to as the resonance enhanced drilling [5]. The system, alongside the existing rotation of the drill-string during rotary drilling generates high frequency periodic impacts very close to the drill-bit via the effect of resonance between the drill-bit and the drilled rock. The percussive impacts ensure that the rock area underlying the bit is intensely weakened while the bit's rotation ensures that the weakened zone is timely removed and that fresh rock surface is continuously exposed for fracturing. This thus prevents the re-compaction of the already fractured rock materials and increases ROP. The intermittent impact actions instead of the continuous grinding action of conventional rotary drilling reduces the contact time between the bit and the rock thus reducing its tearing and wearing [2, 6, 7]. Fig. 1(a) shows the rock fragmentation mechanism of the VID system as a typical rotary-percussive drilling system. Being a periodically impacting system, the categories impact motions obtainable from the VID system are annotated as P-i-j, where "i" is the number of period(s) expended before the typical impact motion repeats itself and "j" is the number of impacts made within the expended period(s).

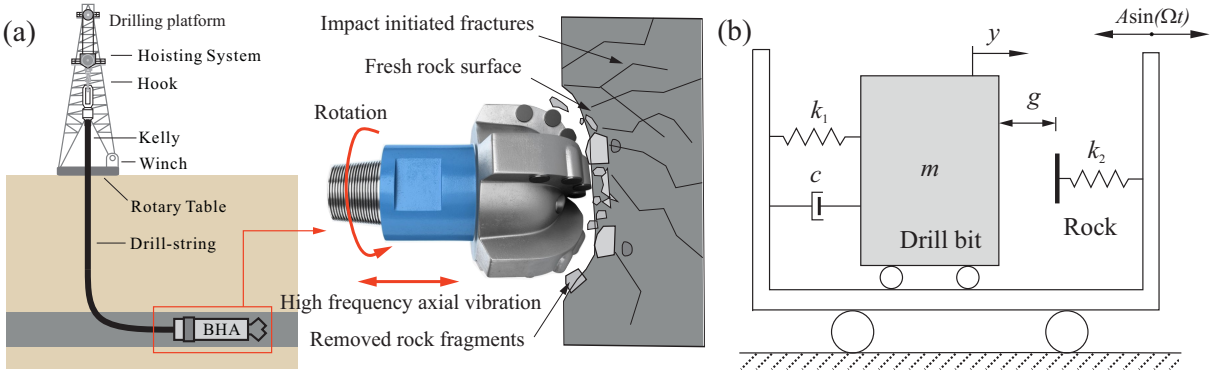


Figure 1: (Colour online) (a) Rock fragmentation mechanism of a typical rotary-percussive drilling system and (b) physical model of the impact system representing bit-rock interaction during percussion.

For conventional drilling systems, weight on bit and rotary speed are two key parameters that are usually adjusted to improve ROP [8, 9], however, this is not the case for the VID system. While studying the VID system and its optimal performance conditions, it was observed that ROP increases significantly with increasing forcing amplitude of the system provided the impact motion is of P-1-1 category [4, 10, 11]. Also, it can be observed from [4, 12, 13] that impact motions with the same number of impact(s) as their period(s) of excitation gave higher ROP compared to those with non-identical number of impacts and excitation periods. These observations imply that, despite fulfilling the conditions for resonance in the VID system, the categories of resulting impact motions is crucial to its performance, and that only high performing impacts should be maintained during drilling operations. However, changing downhole conditions which is synonymous to changing system parameters, influenced by the inhomogeneous rock layers often throws the system into different categories of impact motions. This has been numerically established via bifurcation analysis [4, 13] which also showed cases of multistability, i.e., two or more impact motion categories co-existing for the same set of system parameters. This thus implies that system parameters need to be constantly tuned to maintain desirable impact motions while avoiding undesirable ones as soon as possible, especially when the system is exhibiting multistability.

With the VID system obscurely hidden down the borehole and the need to constantly monitor its

exhibited impact motions, it has become necessary to develop a real-time impact motion characterisation system. Previous experimental [14, 15] and theoretical [4, 13, 16] studies of the VID system have revealed that complex nonlinear relationship exist between its operational parameters, its measurable variables and its exhibited impact motion categories. For this reason, the use of artificial neural networks capable of learning complex nonlinear relationship from data [17–19] is investigated in this present paper to distinguish between downhole impact motions of the VID system. The impact motions data were obtained as long time series data of measurable vibration variables. As a suggestion for improving the use of long time series data in machine learning, feature extractions was carried out [18, 20]. Self-defined features including morphological waveform features and statistical features were crafted from the data. Automatic feature extractions using autoencoders, histogram of oriented gradients (HOG) and pre-trained convolutional network were also carried out. The use of direct image classification models, such as convolutional neural networks (CNN), long short-term memory network (LSTM) and stacked autoencoders (SAE) was also explored. The outcome of these approaches were compared to that of using original raw data. In order to meet the large data requirement for training supervised machine learning algorithms and to also reduce the disparity associated with validating simulation models with experimental data, the practice of merging both simulation and experimental data for network training was adopted [21–23]. The training data were thus composed of both simulation data which were inexhaustible and experimental data which were limited and scarce. The fully trained networks were tested on two sets of out-of-training sample data. The first set consisted of a five-categories impact data from simulation while the second set consisted of a three-categories impact data from experiment.

As a new drilling technology, the present study forms an integral part of the VID optimisation strategies and has demonstrated the practicality of a machine learning-based real-time impact motion categorisation system using readily measurable impact dynamics, and we believe that this is the first time it is being attempted. The remaining sections of this paper are organised as follows. The impact system as a representation of bit-rock interaction during percussive drilling alongside its mathematical and experimental models are presented in Section 2, while a brief description of the feature extraction and the network classifiers is presented in Section 3. Section 4 contains the results and discussions from the raw data and feature-based learning. Finally, conclusions are drawn in Section 5.

## 2. Mathematical description and experimental apparatus

### *2.1. Mathematical model of the impact system*

Fig. 1(b) shows the physical model of the impact oscillator system representing the bit-rock impact actions of the VID. In this model, it is assumed that the discontinuity boundary for impact is fixed at $x = e$, with $e > 0$ being the nondimensional gap. The equations of motion of the impact system can be written in a nondimensional form as:

$$\begin{cases} x' = v, \\ v' = a\omega^2 \sin(\omega\tau) - 2\zeta v - x - \beta(x-e)H(x-e), \end{cases} \tag{1}$$

where $x'$ and $v'$ are the derivatives of displacement and velocity with respect to the dimensionless time $\tau$, respectively, and $H(\cdot)$ stands for the Heaviside step function. The variables and parameters in Eq. (1)

are nondimensionalised as:

$$x = \frac{y}{y_0}, \quad \beta = \frac{k_2}{k_1}, \quad \zeta = \frac{c}{2m\omega_n}, \quad \omega_n = \sqrt{\frac{k_1}{m}},$$
$$\omega = \frac{\Omega}{\omega_n}, \quad a = \frac{A}{y_0}, \quad e = \frac{g}{y_0}, \quad \tau = \omega_n t, \tag{2}$$

where $y_0 > 0$ is an arbitrary reference distance, $\beta$ is the stiffness ratio, $\zeta$ is the damping ratio, $\omega_n$ is the natural frequency, $\omega$ is the frequency ratio, and $a$ is the nondimensional amplitude of external excitation.

### *2.2. Experimental apparatus and set-up*

The experimental apparatus of the impact system is presented in Fig. 2(a), where a wheeled aluminium holding frame is excited harmonically by a linear DC servomotor on a fixed basement. A mass ($m$) which is restricted to only horizontal displacement is attached to the frame via two parallel leaf springs with stiffness $k_1$ and damping coefficient $c$. $k_1$ and $c$ were identified through free vibration tests, and both of them can be adjusted by varying the length of the leaf springs. Impacts occur when the mass hits the bolt on the elastic beam attached to the opposite arm of the frame with stiffness $k_2$ at a distance $y > g$. The stiffness of the beam can be changed by varying its length to reflect different rock layers during VID. Mass displacement ($y$) and acceleration ($\ddot{y}$) are measured by an eddy current probe and an accelerometer, respectively. As can be seen from the schematics of the rig in Fig. 2(b), the position of the motor's rod is controlled by a PID motion controller and is displayed together with the mass displacement and acceleration through a graphic user interface in Labview. So, the rod connected to the holding frame can be harmonically excited at the desired frequency ($\Omega$) and amplitude ($A$). The identified physical parameters of the experimental rig are given in Table 1. Samples of obtained periodic time histories including mass displacement and acceleration smoothened with Savitzky-Golay algorithm are presented in the internal panel of Fig. 2(a), where impact regimes are characterised with sharp spikes of acceleration.

Table 1: Identified physical parameters of the experimental rig.

| Parameters | Symbols | Values | Units |
|---|---|---|---|
| Mass | $m$ | 0.503 | kg |
| Stiffness of the leaf spring | $k_1$ | 0.42 | kN/m |
| Damping coefficient of the leaf spring | $c$ | 0.032 | Ns/m |
| Stiffness of the elastic beam | $k_2$ | 3.36 | kN/m |
| Gap | $g$ | 4.65 | mm |
| Excitation frequency | $\Omega$ | [3.8, 5.5] | Hz |
| Excitation amplitude | $A$ | [0.5, 1.1] | N |

### *2.3. Simulated and experimental periodic motions*

Typical categories of impact motions used in this present study were obtained from both the mathematical and experimental model of the impact oscillator. As earlier stated, the long existing practice of augmenting scarce experimental dataset with inexhaustible simulation data to improve machine learning was adopted [21–23]. A mixture of all categories of impact motions from the simulation and experiment were used to train the networks. However, to evaluate them, the networks were separately tested on simulation and experimental data that were never part of the training data. By adopting the system

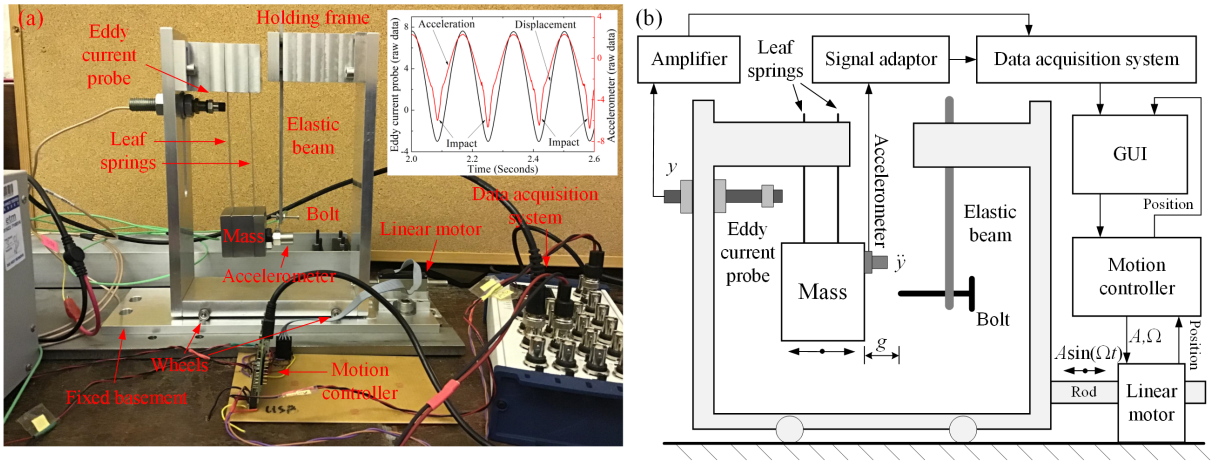


Figure 2: (Colour online) (a) Photograph and schematics of the experimental apparatus. A wheeled holding frame is driven freely by a linear DC servomotor on a fixed basement under sinusoidal excitation. A mass is connected with the holding frame via two parallel leaf springs that prevent it from rotation ensuring horizontal displacement only. Impact will occur when the mass hits the bolt that is attached to the elastic beam mounted on a separate column of the holding frame. Mass displacement and acceleration are measured by an eddy current probe and an accelerometer, respectively, and then collected by the data acquisition system. A sample of the raw data is shown in the internal panel of (a), where impact regimes are easily recognisable in the form of sharp spikes in acceleration. The motor has a movable rod which is harmonically excited using desired frequency and amplitude. Position of the rod is monitored by a PID motion controller and is displayed together with the mass displacement and acceleration through a graphic user interface (GUI) in Labview at a sampling rate of 1 kHz.

parameters stated in Table 2, Eq. (2) was solved for displacement ($x$), velocity ($v$) and acceleration ($v'$) of the mass $m$ at different initial conditions by using the Runge-Kutta fourth-order method in MATLAB. Fig. 3 shows the basins of attraction of the system which contain all the initial conditions that lead to the resulting categories of impact motions.

Table 2: System parameters for originating co-existing impact motions in simulation.

| Parameters | MBSN1 | MBSN2 | MBSN3 |
|---|---|---|---|
| $\omega$ | 0.75 | 0.935 | 0.8063 |
| $a$ | 5.6 | 0.7 | 0.7 |
| $\beta$ | 18.27 | 29 | 29 |
| $\zeta$ | 0.01 | 0.01 | 0.01 |
| $g$ | 2.1 | 1.26 | 1.26 |

In all five categories of impact motions, P-1-1, P-1-2, P-2-1, P-3-2 and P-3-3 were simulated. For the experimental investigation, only P-1-1, P-3-2 and P-3-3 impact categories were obtainable. The acceleration histories of the vibrating mass were used to categorise the resulting impact motions due to their high sensitivity to impacts. At impact, the mass acceleration starts to decrease rapidly due to the resistance imposed on it by the constraining elastic force of the impacted beam. Depending on the present stiffness of the beam, the displacement of the parallel leaf springs reaches maximum and so also its deceleration, and it returns back to its initial position. This effect results in the occurrence of high amplitude spikes along the acceleration signal. Typical acceleration time series resulting from simulation and experiment are respectively shown in Fig. 4 and Fig. 5 alongside their phase portraits.

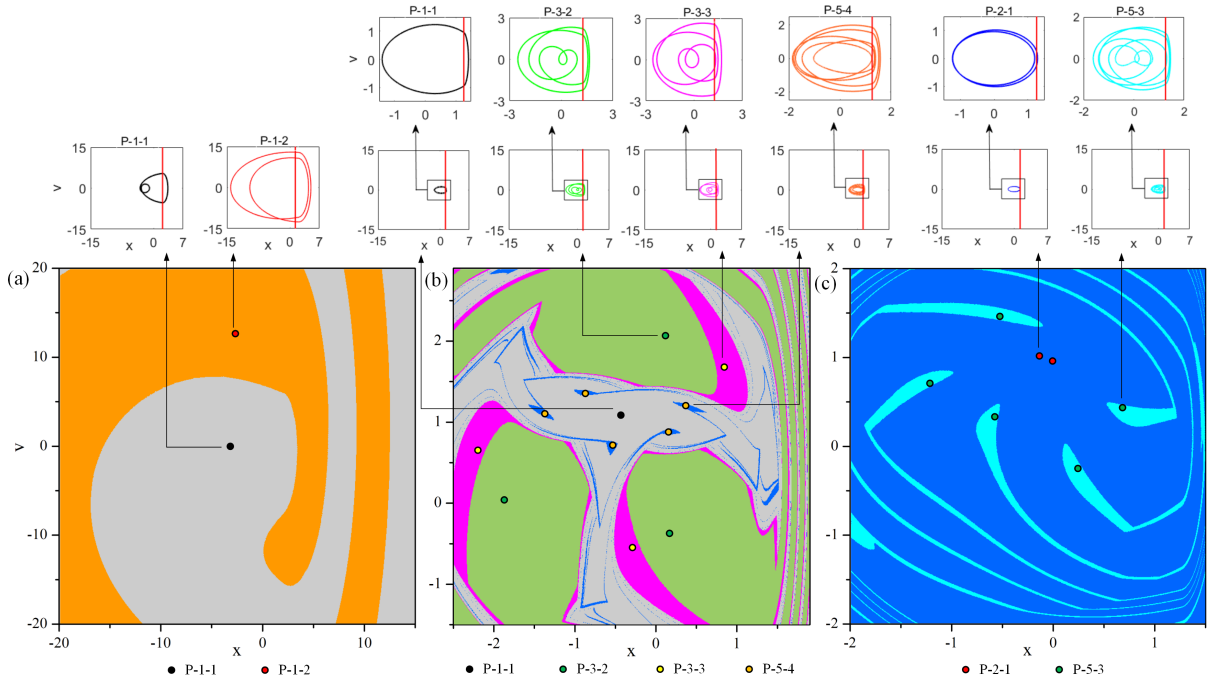


Figure 3: (Colour online) Basins of attraction for (a) MBSN1 (b) MBSN2 and (c) MBSN3 showing P-1-1 (black), P-1-2 (red), P-3-2 (green), P-3-3 (magenta), P-5-4 (orange), P-2-1 (blue) and P-5-3 (cyan). The solid red lines in the phase portraits denote the impact boundary. In the present study, P-1-1, P-1-2, P-2-1, P-3-2 and P-3-3 were used.

## 3. Feature extraction and classifier networks

### *3.1. Feature extraction*

In order to develop a robust and less erroneous classification networks, feature extraction is often encouraged, however, with little or no domain expertise, defining such features can be challenging. The advent of image processing tools has provided a comprehensive and a less interactive way of extracting features without using domain expertise. In this study, self-defined feature representations including waveform morphological and statistical features, and automatic feature representations using stacked-autoencoders, HOG descriptors and pre-trained convolutional network have been carried out. To minimise the effect of embedded noise without significantly distorting the signal, the experimental signals were smoothened using the Savitzky-Golay algorithm to fit a second order polynomial over a group of 25 data points.

#### *3.1.1. Waveform morphological time-interval features*

These are numerically coded morphological and time-tied features manually extracted from the signal waveform in its time-domain format. Common morphologies of time series signals include peaks and troughs, and studies, [24–26], have shown that their intrinsic properties and distribution along the signal can be explored to gather information about the signals. For this study peak amplitudes ($A_{\mathrm{pk}}$), peak locations ($L_{\mathrm{pk}}$) and peak gaps ($G_{\mathrm{pk}}$) were coded from the peaks of the signals while periodic locations ($L_{\mathrm{prd}}$) and period gaps ($G_{\mathrm{prd}}$) were coded from the period information of the signal. Before extracting these features, the negatively projected forward impact peaks were converted to positive peak. Also, the numerical variation between the different categories of signals and also between the mathematically simulated and experimental variants (as seen in Fig. 4 and Fig. 5) was nullified by scaling the values to

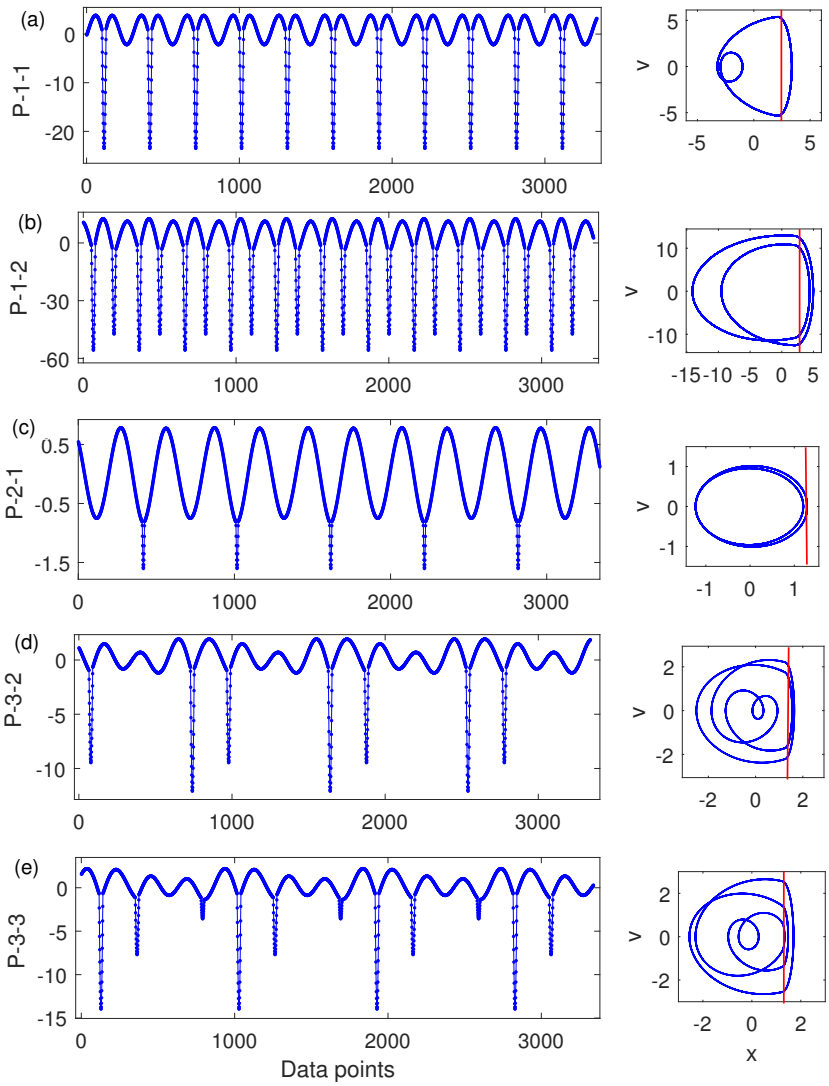


Figure 4: Representative acceleration time histories and phase portraits for typical simulated impact motions with the red line representing impact boundary.

an interval of $[0, 1]$. This procedure allows data on different scales to be compared by converting them to a common unified scale. To extract the morphological time-interval features, the period of oscillation of the signal is first calculated, and from this, $L_{\text{prd}}$ along the signal are deduced and $G_{\text{prd}}$ is calculated. Secondly, the signal is twice differentiated to reveal the start ($S_{\text{pk}}$) and end ($E_{\text{pk}}$) of each peak which also marks the loading and unloading of individual impact action. The actual peak location ($L_{\text{pk}}$) is calculated as the mid-point between $S_{\text{pk}}$ and $E_{\text{pk}}$ while the peak amplitude $A_{\text{pk}}$ is taken as the value of the signal at the peak location (i.e. $L_{\text{pk}}{}^{\text{th}}$ positions). Peak gaps $G_{\text{pk}}$ are calculated as the difference between consecutive $L_{\text{pk}}$ data. Fig. 6 shows the analysed morphological features for (a) simulated and (b) experimental acceleration signal.

In all, three sequences of features annotated as MorFeat1, MorFeat2, and MorFeat3 were extracted. MorFeat1 are row features derived by placing '2' at every peak location ($L_{\text{pk}}$) on the signal while every location marking the end of a period of oscillation is marked with '1', and every other point on the

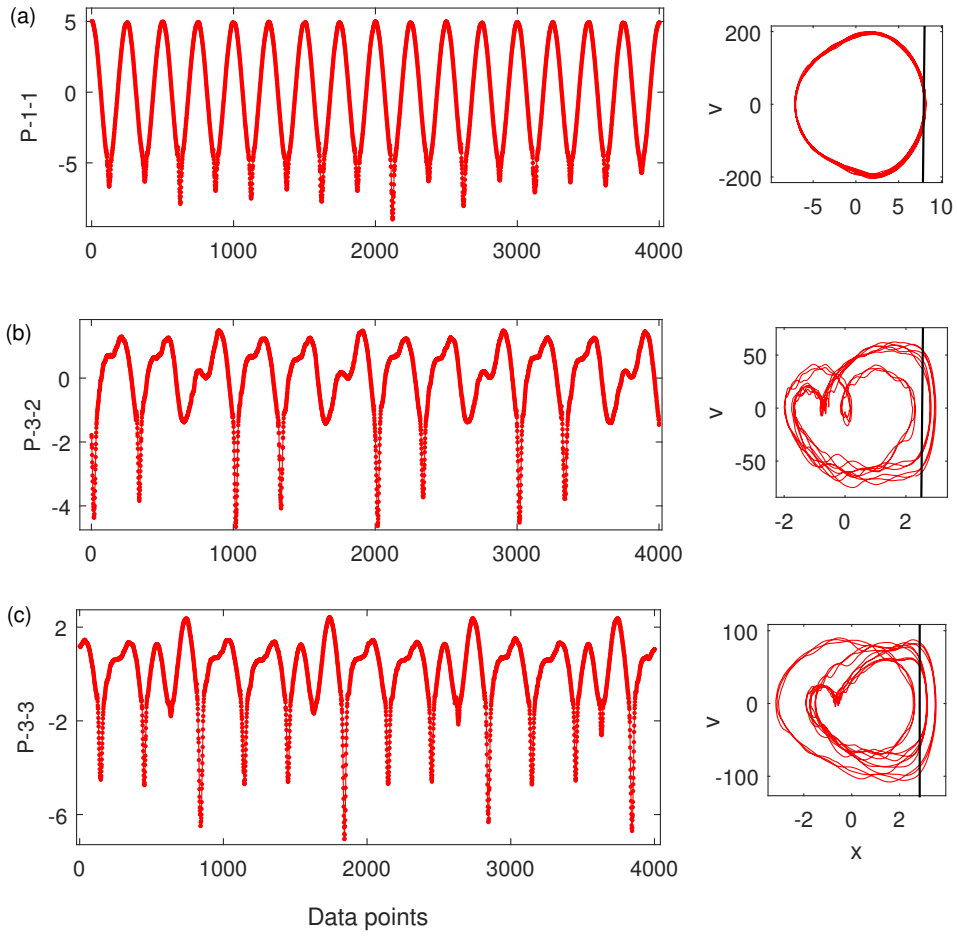


Figure 5: Representative acceleration time histories and phase portraits for typical experimental impact motions with the black line representing impact boundary.

signal is represented with '0' [18]. Like MorFeat1, the MorFeat2 and MorFeat3 are also row features, but only that the peak locations ($L_{\mathrm{pk}}$) along the signal are represented with the peak amplitudes ($A_{\mathrm{pk}}$) for MorFeat2, and with peak gaps ($G_{\mathrm{pk}}$) for MorFeat3. To further remove redundancy, the '0' within these row features were removed thus leaving MorFeat1 as row vectors of '1' and '2', MorFeat2 as row vectors of '1' and peak amplitudes ($A_{\mathrm{pk}}$), and MorFeat3 as row vectors of '1' and peak gaps ($G_{\mathrm{pk}}$). $G_{\mathrm{pk}}$ values of the MorFeat3 features were converted to durations by multiplying with the nondimensional sampling time interval textcolorred$t_{\mathrm{s}}$ given as:

$$t_{\mathrm{s}} = \frac{2\pi}{\omega \times n_{\mathrm{pt}}}, \tag{3}$$

where $t_{\mathrm{s}}$ is the nondimensional sampling time interval and $n_{\mathrm{pt}}$ is the number of sampled points per period. The resulting peak-gap durations of the MorFeat3 features were then normalised and annotated as MorFeat3$^{\mathrm{N}}$ by expressing them as a ratio to the duration of a period of oscillation,

$$\mathrm{MorFeat3}^{\mathrm{N}} = \frac{G_{\mathrm{pk}} \times t_{\mathrm{s}}}{G_{\mathrm{prd}} \times t_{\mathrm{s}}}. \tag{4}$$

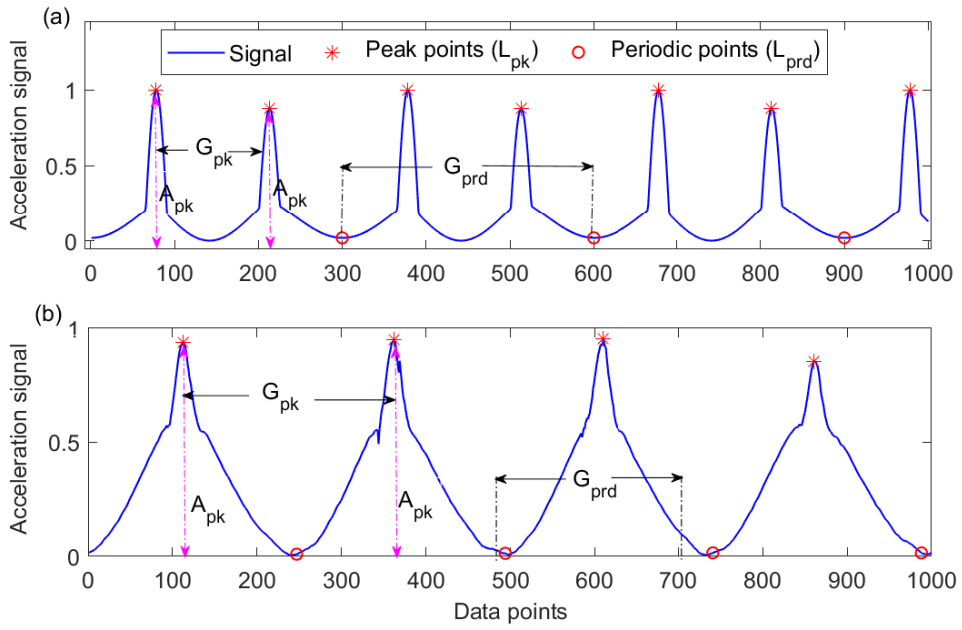


Figure 6: Waveform morphological time-interval features for processed (a) Simulated and (b) Experimental signal.

Figs. 7–12 are typical graphical representation of extracted morphological features from same periods of data for both simulation and experimental signals. It can be observed that despite using same periods of data, the resulting waveform morphological features varied in sizes depending on the impact motion category. The feature sizes varied between 40 and 60, hence all waveform morphological features were pre-padded with zeros to a uniform size of 80. It is also noticed that the MorFeat3$^{\mathrm{N}}$ features for 1-impact motion categories tend to vary along a straight line for both simulation and experimental data. They vary at a value equal to their period(s) of oscillation, hence P-1-1 is seen to vary along a value of 1 while P-2-1 varies along a value of 2 for both simulation and experimental data.

*3.1.2. Statistical features*

A compact representation of the raw data is created from a set of carefully selected descriptive statistical measures and they are used as input into the classifiers. This approach has been used for audio signal classification [27], critical drilling conditions recognition from drilling data [28] and drilling episodes classification [29]. In this study, the final set of statistical measures eventually used, Table 3, were arrived at after downsizing measures with same values from the original set of measures. Some of the measures showed larger values compared to others, hence they were normalised using Eq. (5). Fig. 13 shows the statistical features resulting from a single signal before and after normalisation.

$$X_{\mathrm{norm}} = \frac{X - \min(X)}{\max(X) - \min(X)}. \tag{5}$$

*3.1.3. Continuous wavelet transform for image generation*

Time-frequency analysis of time series data has in recent times proved very useful for extracting local and temporal features from data. Over the years, many literatures have tried to explain in details the basics and methods of converting time series data into their time-frequency domain. One of such

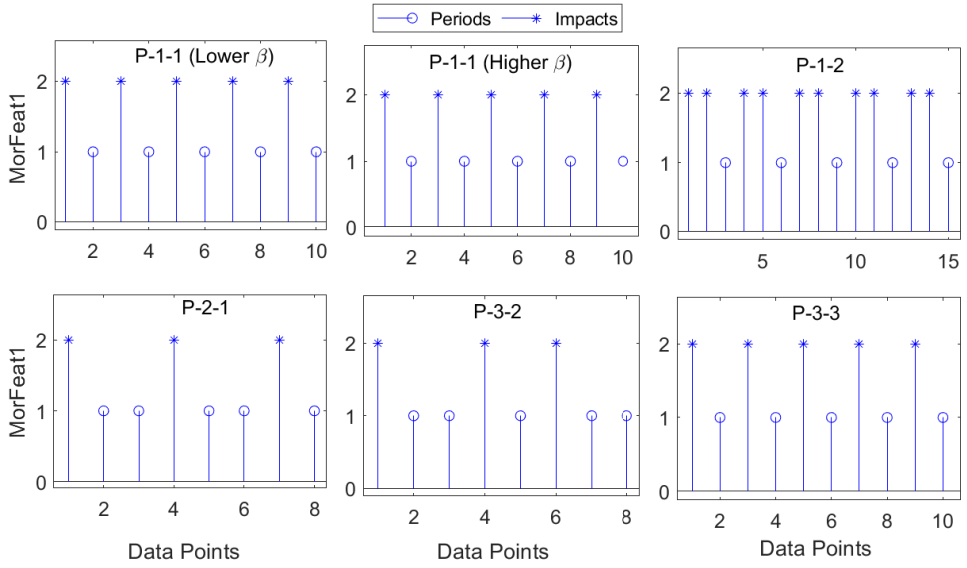


Figure 7: MorFeat1 features: Simulation data.

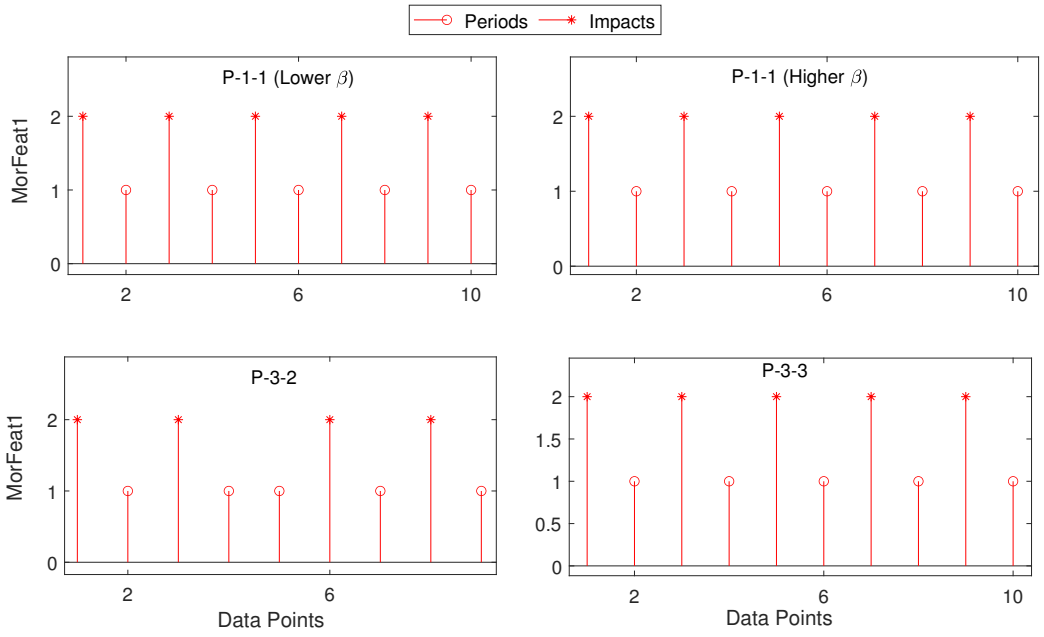


Figure 8: MorFeat1 features: Experimental data.

common method is the continuous wavelet transform (CWT). The method uses finer discretisation to decompose the time series data into a two-dimensional time-frequency space from which the frequency component of the signal is easily observed as a function of time. This is deemed useful for creating discriminating features between signals that show very little discrepancies in their original format as seen in this study and this is readily observed when the resulting coefficients are represented in 2D image formats. The acquisition and utilisation of 2D images for solving classification problems has been in practise for a while [19, 30, 31]. Fig. 14 shows the typical scalogram plots representing 2D plots of the different impact motion signals obtained from simulation and experiment. These 2D images show

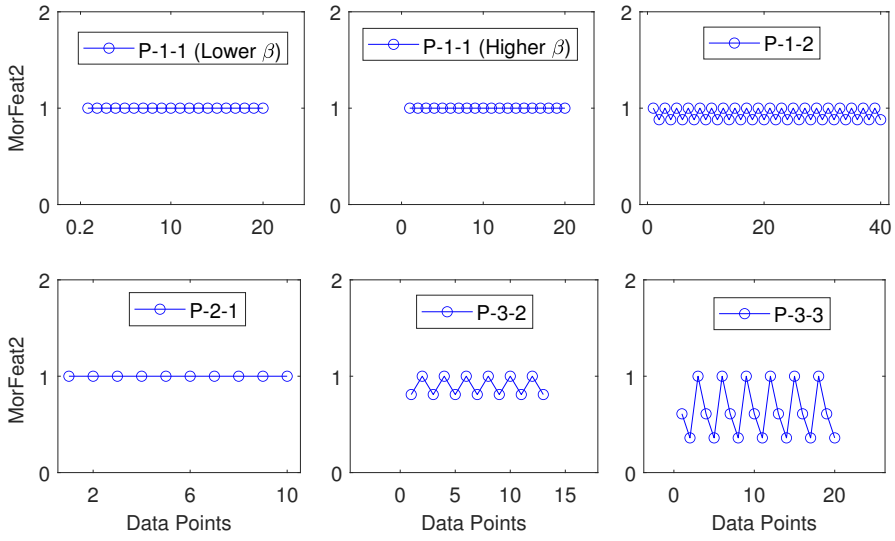


Figure 9: MorFeat2 features: Simulation data

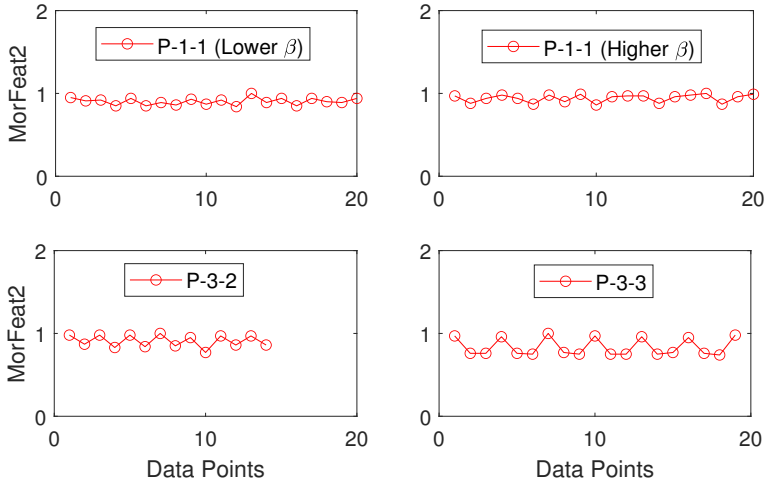


Figure 10: MorFeat2 features: Experimental data.

that the total signal energy for a particular type of impact motion tends to vary in the same manner for both its simulation and experimental counterpart. To analyse these images, analytical tools and networks capable of learning different levels of features from images were utilised. HOG descriptors and pre-trained Resnet18 convolutional network were used to automatically extract high level features from the images which were fed into other network classifiers. Aside extracting features from the vibro-impact images, their direct classification using image compatible networks such as SAE, CAE, CNN and LSTM was also explored at different image sizes.

### *3.1.4. Autoencoder features*

Autoencoders are neural networks capable of unsupervisedly learning to efficiently encode data into a reduced dimensional space [32]. They consist of an encoding and a decoding part and the output is

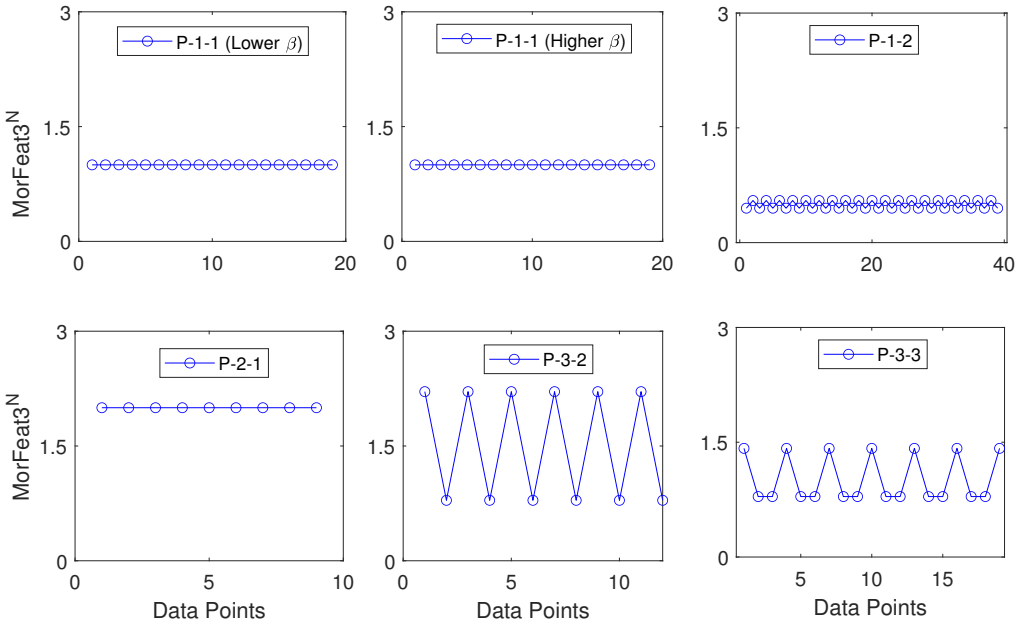


Figure 11: MorFeat3$^{\mathrm{N}}$ features: Simulation data

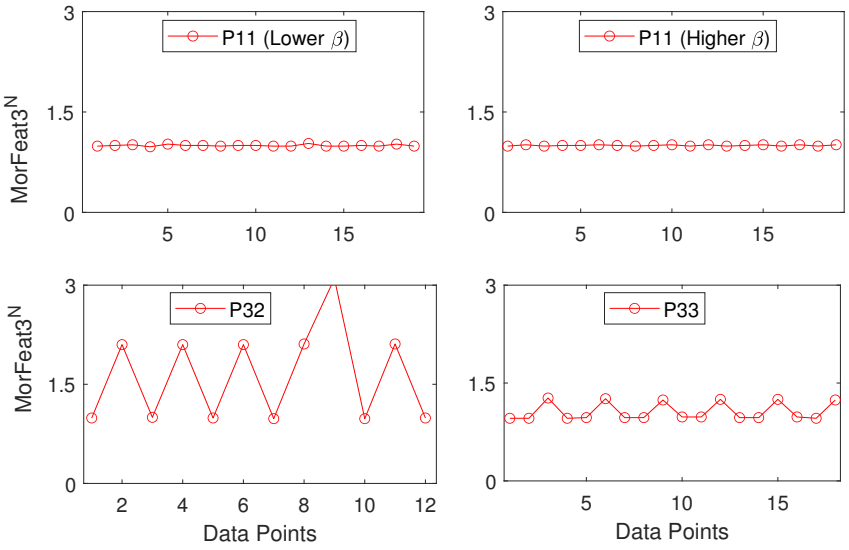


Figure 12: MorFeat3$^{\mathrm{N}}$ features: Experimental data.

usually of the same size as the input. The encoder part, $f(X)$, maps the original $D$ dimensional data $X$ to a lower dimension $Y$ given as

$$Y = f(X) = s_{\mathrm{f}}(WX + b_{\mathrm{X}}). \tag{6}$$

On the other hand, the decoder part, $g(Y)$, reconstructs the original data $X$ from the reduced data $Y$ with minimal mean square error such that

$$X' = g(Y) = s_{\mathrm{g}}(W'Y + b_{\mathrm{Y}}), \tag{7}$$

Table 3: Estimated statistical features.

| Index | Statistical Feature | Index | Statistical Feature |
|---|---|---|---|
| 1 | Mean | 13 | Root-mean-square level |
| 2 | Minimum | 14 | Absolute maximum value to rms ratio |
| 3 | Maximum | 15 | Root-sum-of-squares level |
| 4 | Standard deviation | 16 | Crest factor |
| 5 | Range | 17 | Absolute mean |
| 6 | Kurtosis | 18 | Form factor |
| 7 | Variance | 19 | Impulse factor |
| 8 | Skewness | 20 | Mean square root of absolute data |
| 9 | Sum | 21 | Kurtosis factor |
| 10 | Mean frequency of signal power-spectrum | 22 | Margin factor |
| 11 | Cumulative maximum element | 23 | Skewness factor |
| 12 | Average cumulative minimum element | | |

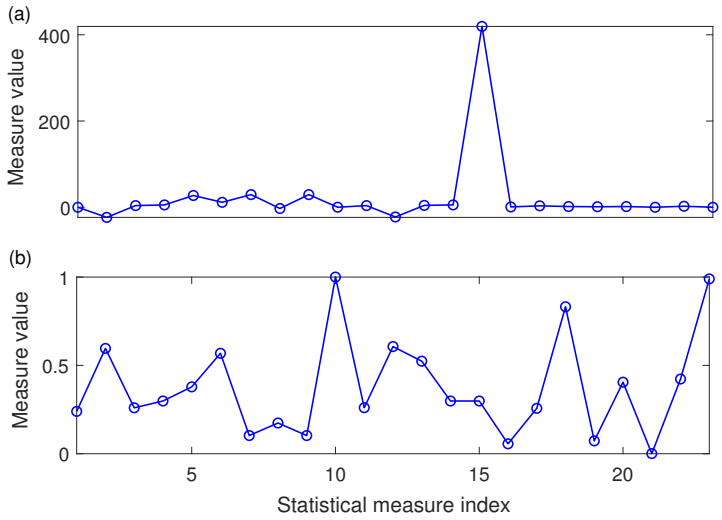


Figure 13: Representative (a) unnormalised (b) normalised statistical features of a signal.

where $s_{\mathrm{f}}$ and $s_{\mathrm{g}}$ are the activation functions for the encoder and the decoder, and $W$ and $b$ represent the weight matrices and bias vectors, respectively. Here, mean square error is used to measure the difference between $X$ and $X'$. The possibility of adapting autoencoders with nonlinear activation functions [33], makes it possible to use them for automatic feature extraction in complex nonlinear problems as used in this study. Fig. 15 shows auto-encoded raw acceleration data from the last layer of the stacked autoencoders shown in Fig. 16. Before being fed into the autoencoders, all the raw data were resized to equal lengths of 3340 data points. These were later resized to 1000 and then 500 using a two-layers stacked-autoencoders.

### *3.1.5. Histogram of oriented gradient features*

Aside requiring a domain knowledge of analysed signal, constructing self-defined features like the waveform morphological features can be cumbersome and time-sapping. Formatting data for use with networks like the autoencoders can as well be very tasky especially for high dimensional and non-identical size datasets. These challenges has attracted the use of descriptors like the HOG, which requires minimal human interaction, to extract high level textural features from image representation of data [19, 34].

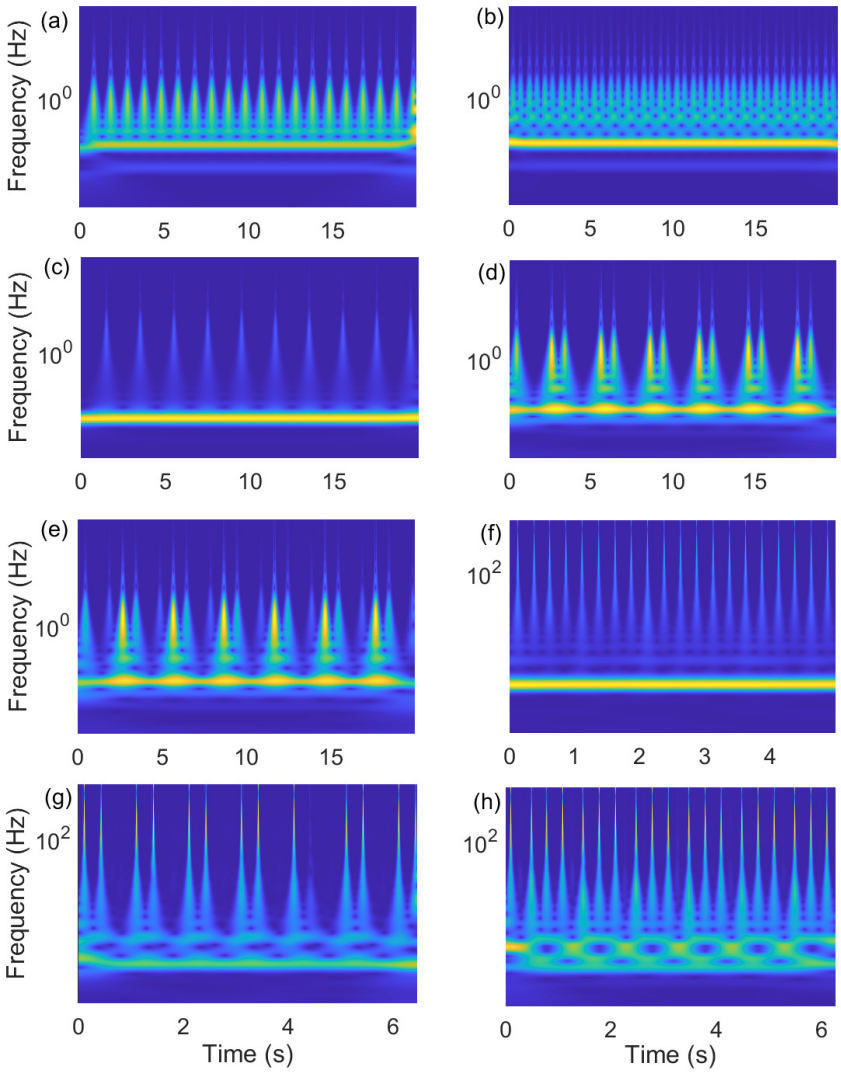


Figure 14: (Colour online) Scalogram plots from CWT analysis for simulated (a) P-1-1 (b) P-1-2 (c) P-2-1 (d) P-3-2 (e) P-3-3 and experimental (f) P-1-1 (g) P-3-2 (h) P-3-3. The images show that the total signal energy tend to vary similarly for same impact categories whether it being simulation or experimental.

The features are extracted as estimates of local intensity gradients and before estimation, a gray scale normalisation and a binarisation procedure is performed on the images to unify their intensity and scale them to $[0, 1]$. During the HOG feature estimation, the image is divided into multiple $m$-by-$m$ pixel cells and using

$$\begin{cases} G_{\mathrm{x}} = I(x+1, y) - I(x-1, y)t, \\ G_{\mathrm{y}} = I(x, y+1) - I(x, y-1), \end{cases} \tag{8}$$

$G_{\mathrm{x}}$ and $G_{\mathrm{y}}$ respectively representing the horizontal and the vertical gradient information for a pixel of intensity $I$ at location $(x, y)$ are estimated for each pixel in each cell. The size of the multiple cells is usually selected to ensure that sufficient but not redundant features are encoded by the descriptors. Then

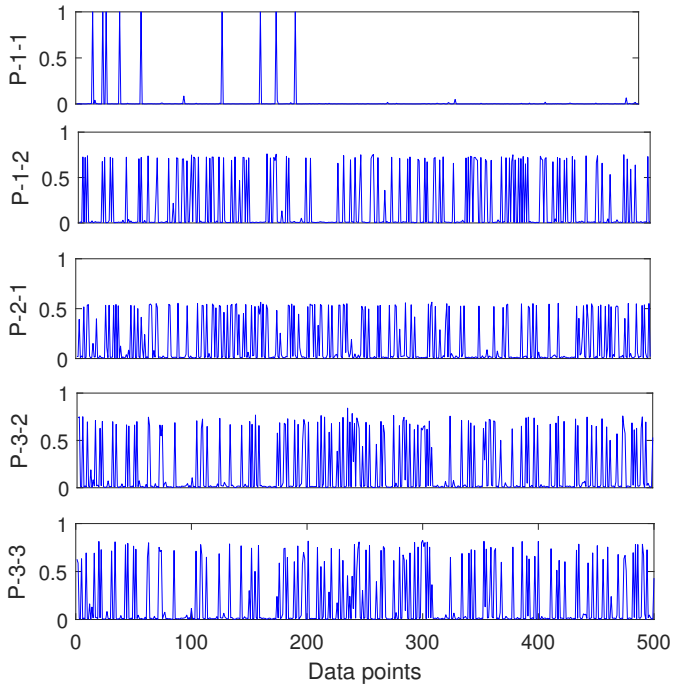


Figure 15: Representative auto-encoded features for different impact motions.

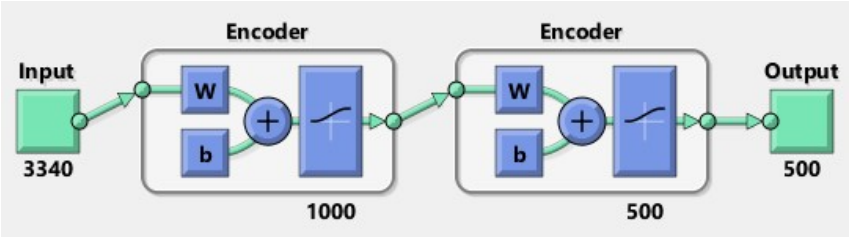


Figure 16: Two layers stacked-autoencoders for raw acceleration data encoding.

the magnitude $M$ and orientation $\theta$ are further calculated as

$$M = \sqrt{G_{\mathrm{x}}^2 + G_{\mathrm{y}}^2} \tag{9}$$

and

$$\theta = \arctan\left(\frac{G_{\mathrm{y}}}{G_{\mathrm{x}}}\right). \tag{10}$$

Thereafter, calculated $\theta$ belonging to the same cell are vote-weighted using $M$ and accumulated into bins of histogram. The bins of histograms resulting from all the cells are concatenated to construct the final L1-normalized feature vectors. Fig. 17 illustrates the HOG feature extraction process. For this study a cell size of $[60 \times 60]$ was used with a bin of 9. As earlier stated the analysed images were generated as 2D scalogram plots of the coefficients resulting from the CWT of raw acceleration data. Fig. 18 shows the scalogram of a simulated and an experimental P-1-1 and P-3-2 impact motions alongside their HOG information.

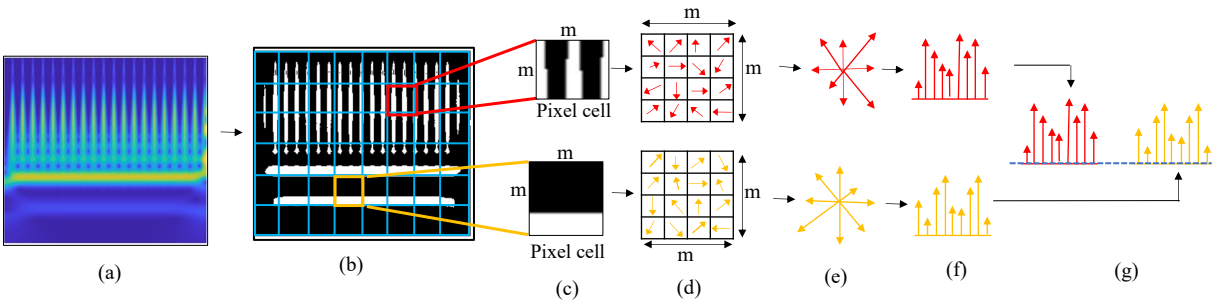


Figure 17: Process of HOG feature extraction: (a) Image generation, (b,c) gray scale normalisation, binarisation and division to m-by-m cells, (d,e) gradient magnitude and orientation computation, (f) spatial/orientation binning and (g) concatenation of binned orientation.

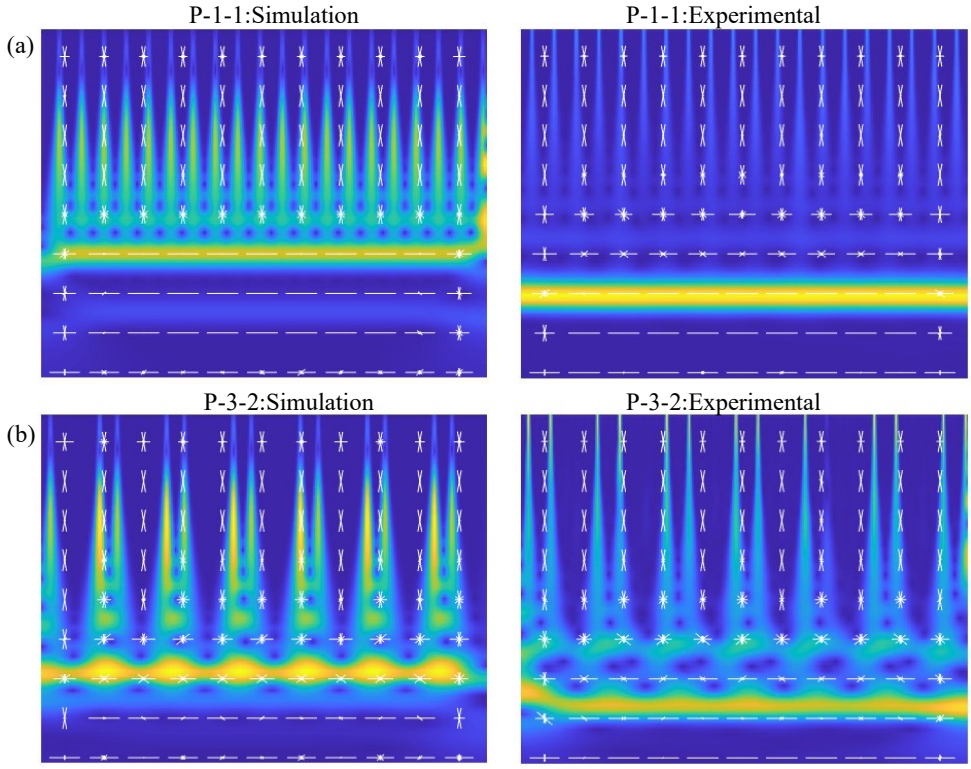


Figure 18: (Colour online) Simulated and experimental acceleration times histories with the encoded HOG information for (a) P-1-1 and (b) P-3-2.

### *3.1.6. Pre-trained convolutional neural network features*

Automatic feature extraction via the use of pre-trained CNN to effortlessly extract discriminative features from images has been around for a while [35, 36]. Architecturally, CNNs consist of an image input layer, an intermediate section of several hidden layers and a final section consisting of a fully connected layer, a softmax or regression layer for classification and regression problems respectively and an output layer. The hidden layers are able to hierarchically extract deep level features that are specific to the input images. During the automatic feature extraction, images are passed into the pre-trained network from the input layer, and by using activation functions, learnt features are intercepted at desired layers using activation functions. The resulting layer activations are saved as feature vectors which can be

used as input data into other classifiers. Several pre-trained CNN networks exist, but based on reported performance and memory requirement [37], the *Resnet*18 network with 18 layers and trained on ImageNet database [38] was selected for this study. Resnet18 accepts images of size 224-by-224 while the spatial size of the output volume from a given convolutional layer is given as

$$V_{\text{out}} = \frac{V_{\text{in}} - K + 2P}{S} + 1, \tag{11}$$

where $V_{\text{out}}$ is the output volume, $V_{\text{in}}$ is the input volume size, $K$ is the filter size, $P$ is the padding size and $S$ is the stride size. In the present study, 512 feature vectors were extracted for each of the vibro-impact images from the global pooling layer (pool5) located in the $17^{\text{th}}$ convolutional layer of the *Resnet*18 network.

### *3.2. Network classifiers*

For the purpose of developing fully trained networks that can easily be adapted into online monitoring system at minimal cost, classifiers with simple architectures and low memory requirement were explored in this study. These include multi-layer perceptrons (MLPs), multi-class support vector machines (SVMs) and LSTMs.

#### *3.2.1. Multi-layer perceptron*

MLP networks have been used as universal approximators to model nonlinear classification [39] and regression [40] problems. For an input data $x_{\text{i}}$, where $i = 1, 2, \ldots\ldots, \bar{N}$, the network output $y_{\text{out}}$ is [41]

$$y_{\text{out}} = f_{\text{output}} \left( \sum_{\text{j}=1}^{\bar{M}} W_{\text{j}} f_{\text{hidden}} \left( \sum_{\text{i}=1}^{\bar{N}} W_{\text{ji}} x_{\text{i}} \right) + W_{\text{o}} \right), \tag{12}$$

where $\bar{N}$ is the number of input data, $\bar{M}$ is the number of hidden neurons, $x_{\text{i}}$ is the $i^{\text{th}}$ input data, $W_{\text{ij}}$ is the weight parameter between the $i^{\text{th}}$ input data and $j^{\text{th}}$ hidden neuron and $W_{\text{j}}$ is the weight parameter between the $j^{\text{th}}$ hidden neuron and the output neuron. The activation function $f_{\text{output}}$ is given as a sigmoid function for classification problems and as a linear activation function for regression, while $f_{\text{hidden}}$ is a hyperbolic tangent function. The network weights and biases are iteratively adjusted during training to maximise the prediction ability of the network on the input data using backpropagation algorithms. The difference between the network's prediction $y^{\text{p}}_{\text{out}}$ and the actual target ($tt_{\text{p}}$) is defined as a cross-entropy error,

$$E_{\text{xr}} = -\sum_{\text{p}=1}^{P} \{ tt_{\text{p}} \ln(y^{\text{p}}_{\text{out}}) + (1 - tt_{\text{p}}) \ln(1 - y^{\text{p}}_{\text{out}}) \}. \tag{13}$$

#### *3.2.2. Support vector machines*

SVMs were initially developed for binary classification problems, the network takes each row of input data and label it as either belonging to a given class (positive) or not (negative). During training, the input data are analysed alongside their corresponding output labels to establish an optimal separating hyperplane between the positive and the negative samples. For this study, the SVMs have been adapted for multi-class classification by creating multiple binary-class SVM classifiers for different pair-wise combinations of classes. The decisions from the multiple binary SVM classifiers are then fused together to create the final multi-class classifier using an error-correcting output code combination technique [42]

as available in MATLAB [43]. The technique involves a coding and decoding process using a matrix of codewords (usually $1, 0, 1$) of size $k \times l$ [44], where $k$ is the number of classes and $l$ is the number of SVM binary classifiers given as

$$l = \frac{k(k-1)}{2}. \tag{14}$$

As obtained in this study, Table 4 is a code matrix $C$ for a five-class classification task using ten learners. Data entries belonging to a class, $k_{\mathrm{i}}$, is positive for the binary classifier $l_{\mathrm{j}}$ if $C_{\mathrm{ij}} = 1$, negative if $C_{\mathrm{ij}} = -1$ and zero if the class is not included in the classes used in training the binary classifier. The table shows the current multi-class SVM network, and it trains its first four binary learners ($l_1 - l_4$) on class $k_1$ data as it assigned a positive value to the class compared to $k_2$, $k_3$, $k_4$ and $k_5$ which it assigned negative values and zeros.

Table 4: Code matrix for a five class problem using ten learners.

| | $l_1$ | $l_2$ | $l_3$ | $l_4$ | $l_5$ | $l_6$ | $l_7$ | $l_8$ | $l_9$ | $l_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $k_1$ | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| $k_2$ | -1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| $k_3$ | 0 | -1 | 0 | 0 | -1 | 0 | 0 | 1 | 1 | 0 |
| $k_4$ | 0 | 0 | -1 | 0 | 0 | -1 | 0 | -1 | 0 | 1 |
| $k_5$ | 0 | 0 | 0 | -1 | 0 | 0 | -1 | 0 | -1 | -1 |

*3.2.3. Long short-term memory networks*

LSTM networks are forms of recurrent neural networks that are adapted to learning from sequential data. They have memory cells and gates in place of the usual inter-connecting hidden neurons which makes them immune to the problem of vanishing gradient [45]. Aside from being able to learn from long time series data, LSTM networks are also able to accept data of different lengths as input. At every time step $t$ of a time series, the network uses its initial state values including, $c_{\mathrm{t}-1}$ and $h_{\mathrm{t}-1}$, alongside its sequence value $x_{\mathrm{t}}$ to calculate a new learnt information $h_{\mathrm{t}}$ and an updated cell state $c_{\mathrm{t}}$. At any time step $t$, the calculations are

$$\hat{i}_{\mathrm{t}} = \sigma_{\mathrm{g}}(W_{\mathrm{i}}\hat{x}_{\mathrm{t}} + R_{\mathrm{i}}h_{\mathrm{t}-1} + b_{\mathrm{i}}), \tag{15}$$

$$\hat{f}_{\mathrm{t}} = \sigma_{\mathrm{g}}(W_{\mathrm{f}}\hat{x}_{\mathrm{t}} + R_{\mathrm{f}}h_{\mathrm{t}-1} + b_{\mathrm{f}}), \tag{16}$$

$$\hat{g}_{\mathrm{t}} = \sigma_{\mathrm{g}}(W_{\mathrm{g}}\hat{x}_{\mathrm{t}} + R_{\mathrm{g}}h_{\mathrm{t}-1} + b_{\mathrm{g}}), \tag{17}$$

$$\hat{o}_{\mathrm{t}} = \sigma_{\mathrm{g}}(W_{\mathrm{o}}\hat{x}_{\mathrm{t}} + R_{\mathrm{o}}h_{\mathrm{t}-1} + b_{\mathrm{o}}), \tag{18}$$

$$c_{\mathrm{t}} = \hat{f}_{\mathrm{t}} \odot c_{\mathrm{t}-1} + \hat{i}_{\mathrm{t}} \odot \hat{g}_{\mathrm{t}}, \tag{19}$$

$$h_{\mathrm{t}} = \hat{o}_{\mathrm{t}} \odot \tanh(c_{\mathrm{t}}), \tag{20}$$

where $\hat{i}_{\mathrm{t}}$, $\hat{f}_{\mathrm{t}}$, $\hat{g}_{\mathrm{t}}$ and $\hat{o}_{\mathrm{t}}$ are state vectors of $[0, 1]$, $\sigma_{\mathrm{g}}$ and $\odot$ represent a sigmoid and an element-wise multiplication function, respectively. In this study, the bi-directional LSTM (BiLSTM) layer which is able to learn both forward and backward two-way relationship between sequential data was used.

## 4. Classification of the vibro-impact motions

*4.1. Feature-based classification*

The schematic layout of the designed classification models starting with the simulation and experimental generation of data is shown in Fig. 19. It involves the collection of raw acceleration data, and then

the training and cross-validation of different network models using raw acceleration data and extracted features. As stated earlier, five categories of impact motions were obtained from the simulation while only three were obtained from the experiment, however, the developed networks were trained to distinguish between five categories of impact motions using a combination of the simulation and experimental data. Due to the scarceness of some impact motion categories in experiments, the experimental data was first divided into a training and testing set, after which the scarce categories in the training set were augmented by replication as shown in Table 5. The statistics of the simulated data and the augmented experimental data by impact motion categories is presented in Table 6. The simulation and experimental training data were merged together to train the networks while the testing data from simulation and experiment were used separately to cross-validate the networks. Our expectation is for the trained network models to be able to distinguish between impact motion categories correctly even when limited classes are presented.

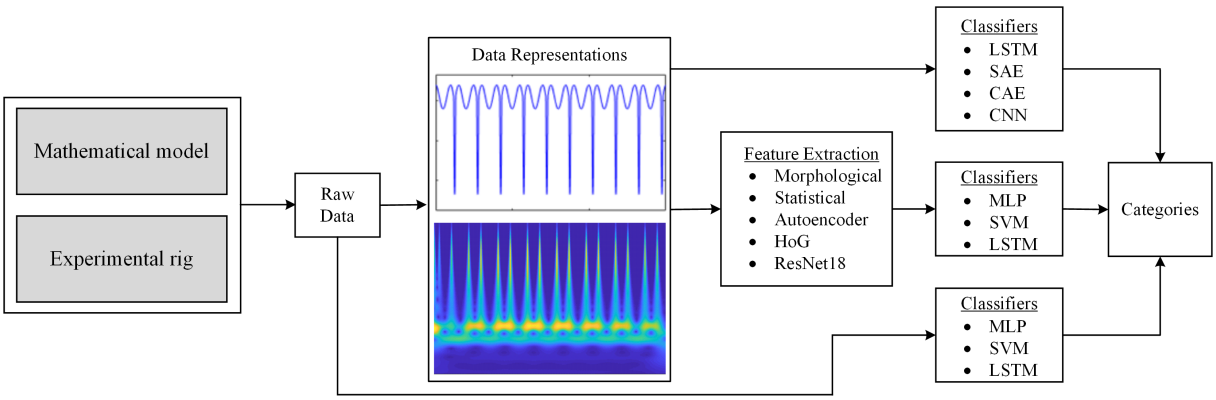


Figure 19: Schematic layout of the classification model.

Table 5: Division of experimental data.

| Motion | Total | Training | Test |
|---|---|---|---|
| P-1-1 | 597 | 200 | 397 |
| P-1-2 | 0 | 0 | 0 |
| P-2-1 | 0 | 0 | 0 |
| P-3-2 | 125 | 50 (×4) | 75 |
| P-3-3 | 274 | 100 (×2) | 174 |

Table 6: Summary of data categories: both the simulated and experimental training data were merged into one data set for training the networks.

| Motions | Simulation | | Experiment | |
|---|---|---|---|---|
| | Training | Testing | Training | Testing |
| P-1-1 | 200 | 200 | 200 | 397 |
| P-1-2 | 400 | 200 | 0 | 0 |
| P-2-1 | 400 | 200 | 0 | 0 |
| P-3-2 | 200 | 200 | 200 | 75 |
| P-3-3 | 200 | 200 | 200 | 174 |
| Total | 1400 | 1000 | 600 | 646 |

Following the conventional approach of using out-of-sample data for the networks cross-validation, the results of the training and testing of the networks are shown in Tables 7-9. The final number of

hidden units used after few trials and errors are also presented in the tables. For Table 7, 'Tr', 'Val' and 'Ts' represent subsets of train, validation and test data selected randomly by the network from the main training data in the ratio of 70%, 15% and 15%, respectively. For all the tables, 'Sim.' and 'Exp.' denote the simulated and experimental out-of-sample cross-validation data, respectively. The MLP networks were mostly trained using the Levenberg-Marquardt training function except for the raw data case when the Scaled conjugate gradient was used due to the large data size and the large memory requirement of Levenberg-Marquardt. To minimise scaling issue and to also enhance generalisation on new raw data, the raw data were scaled to $[0, 1]$ interval before being fed into the networks. The performance of each network was measured using a cross-entropy function.

Table 7: MLP network results, where 'Tr', 'Val' and 'Ts' represent subsets of train, validation and test data selected randomly by the network from the main training data in the ratio of 70%, 15% and 15%, respectively. 'Sim.' and 'Exp.' denote the simulated and experimental out-of-sample cross-validation data, respectively.

| Feature | Training (%) | | | Testing (%) | | Units |
|---|---|---|---|---|---|---|
| | Tr | Val | Ts | Sim. | Exp. | |
| Raw Data | 100 | 100 | 100 | 80.0 | 52.0 | 30 |
| MorFeat1 | 89.9 | 90.7 | 89.7 | 98.1 | 98.0 | 50 |
| MorFeat2 | 100 | 100 | 100 | 98.1 | 97.7 | 50 |
| MorFeat3$^{\mathrm{N}}$ | 100 | 100 | 100 | 79.0 | 98.3 | 50 |
| MorFeat1-2-3$^{\mathrm{N}}$ | 100 | 100 | 100 | 98.1 | 92.4 | 50 |
| Statistical | 100 | 100 | 100 | 98.1 | 95.5 | 20 |
| Autoencoder | 100 | 100 | 100 | 60.0 | 50.9 | 20 |
| HOG | 100 | 100 | 100 | 91.6 | 97.1 | 50 |
| Resnet18 | 79.0 | 79.0 | 73.0 | 37.1 | 49.5 | 30 |

Table 8: SVM network results, where 'Sim.' and 'Exp.' denote the simulated and experimental out-of-sample cross-validation data, respectively.

| Features | Training (%) | Testing (%) | |
|---|---|---|---|
| | | Sim. | Exp. |
| Raw Data | 100 | 80.0 | 52.6 |
| MorFeat1 | 90.0 | 98.1 | 98.1 |
| MorFeat2 | 100 | 98.1 | 97.7 |
| MorFeat3$^{\mathrm{N}}$ | 100 | 79.0 | 98.3 |
| MorFeat1-2-3$^{\mathrm{N}}$ | 100 | 80.0 | 98.1 |
| Statistical | 100 | 100.0 | 91.5 |
| Autoencoder | 100 | 61.9 | 46.3 |
| HOG | 100 | 98.1 | 78.9 |
| Resnet18 | 78.1 | 58.9 | 30.3 |

The results show that most of the waveform morphology based networks performed well attaining an accuracy as high as 98.1% except for MorFeat3$^{\mathrm{N}}$. MorFeat3$^{\mathrm{N}}$ showed an accuracy of 79.0%, 79.0% and 80.0% respectively on the mathematically simulated test data using MLP, SVM and BiLSTM (see Tables 7-9) and this was due to their absolute inability to distinguish between P-1-1 and P-3-3 impacts (see Fig. 20). A combination of the three morphological features MorFeat1-2-3$^{\mathrm{N}}$ showed accuracies of 98.1%, 80.0% and 100% for MLP, SVM and BiLSTM for the simulated test data, respectively. For the experimental test data, MorFeat1-2-3$^{\mathrm{N}}$ showed accuracies of 92.4%, 98.1% and 97.7% for MLP, SVM and BiLSTM, respectively. The statistical features showed accuracies greater than 91.0% for both the simulated and experimental test data using MLP and SVM, while the BiLSTM network showed an accuracy of 100% on the simulated test data and 77.9% on the experimental test data due to their confusion of P-3-3

Table 9: BiLSTM network results, where 'Sim.' and 'Exp.' denote the simulated and experimental out-of-sample cross-validation data, respectively.

| Features | Training (%) | Testing (%) | | Units |
|---|---|---|---|---|
| | | Sim. | Exp. | |
| Raw Data | 95.4 | 62.9 | 77.2 | 100 |
| MorFeat1 | 90.0 | 80.0 | 98.1 | 60 |
| MorFeat2 | 100 | 100 | 97.7 | 60 |
| MorFeat3$^{N}$ | 100 | 80.0 | 98.5 | 60 |
| MorFeat1-2-3$^{N}$ | 100 | 100.0 | 97.7 | 60 |
| Statistical | 100 | 100.0 | 77.9 | 60 |
| Autoencoder | 100 | 60.0 | 45.5 | 60 |
| HOG | 20 | 20.0 | 61.5 | 60 |
| Resnet18 | 75.8 | 58.9 | 66.9 | 60 |

impact motions for P-3-2 impact motions in the experimental data. Stacked-autoencoder features showed deficiency in their ability to sufficient code information that distinguishes between the impact motions accurately. Confusion matrices resulting from the use of stacked-autoencoder features showed P-3-2 and P-3-3 to be confused for P-1-1 in the simulated test data and P-1-1 for P-3-3 in the experimental data (see Fig. 21). This could be due to the loss of vital information during the process of encoding. For the HOG features, MLP achieved accuracies of 91.6% and 97.1%, while the SVM obtained accuracies of 98.1% and 78.9% for the simulated and experimental data, respectively. BiLSTM showed accuracies of 20.0% and 61.5% for the simulated and experimental HOG features as the network could only distinguish P-1-1 impact motions both during training and testing. This poor performance may be a result of the spatial disorientation of the HOG feature vectors.


Figure 20: Confusion matrices from using (a) MLP (b) SVM and (c) LSTM on MorFeat3$^{N}$ features from simulated test data.

Similar to the stacked-autoencoders, the Resnet18 network could not capture features that adequately distinguish between the impact motions using the scalogram images. Due to data size, using the raw data as input data was computationally intensive requiring higher memory usage and longer processing time. However, comparing the raw data networks with the feature based networks, the MLP and SVM networks performed better on the simulated raw data with an accuracy of 80% compared to BiLSTM which showed an average accuracy of 62.9%. On the other hand, the BiLSTM network showed better performance on experimental raw data compared to both MLP and SVM. This could be as a result of

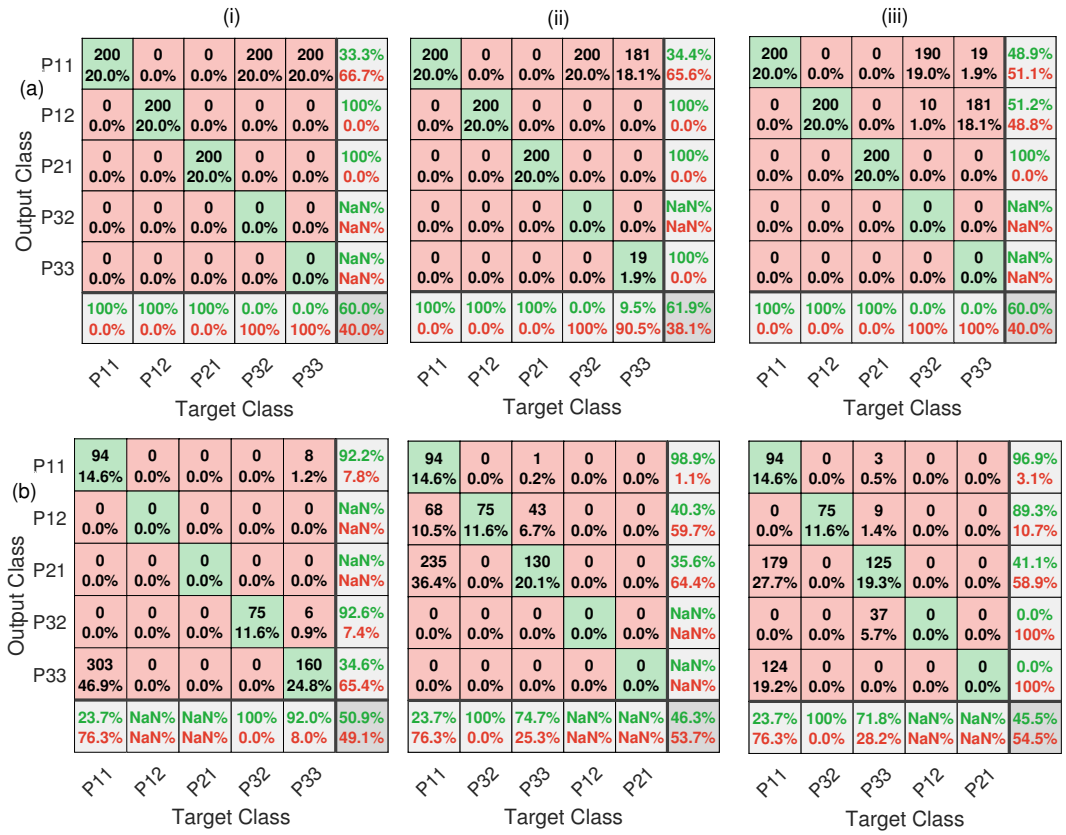


Figure 21: Confusion matrices for autoencoder features based (i) MLP (ii) SVM and (iii) LSTM networks on (a) simulation and (b) experimental test data. The networks are seen to append 'NAN' to the classes unavailable in the experimental data.

ability of the BiLSTM to self-learn distinguishing features that are consistent to both the simulated and experimental data.

On overall, 42% of all the developed feature-based network models showed accuracies greater than 91.0% while 67% of them showed accuracies greater than 77.0%. The MorFeat2 feature-based networks showed the overall best performance with a least accuracy of 97.7% both during training and testing with simulated and experimental data as presented in Fig. 22. As expected, the confusion matrix plots showed the networks appending 'NAN' to the result rows and columns of classes not available in the experimental test data (see Figs. 21-22). A graphical summary of the performances of all the developed networks based on their input data is further presented in Fig. 23.

### *4.2. Direct image classification*

In order to adapt autoencoders for direct image classification, the images are first reshaped into a matrix by stacking up the columns of each image as a single column in a new data vectors to form a matrix which is fed into the autoencoders. Developed autoencoders are stacked up alongside a softmax layer to form the SAE image classifier. Fig. 24 shows two-layer SAE networks developed for analysing images of sizes at $[128 \times 128 \times 1]$ and $[50 \times 50 \times 1]$. In the case of LSTM networks, the input layer was replaced with an image input layer, while sequence folding and sequence unfolding layers were used respectively to convert the incoming sequence of images to image arrays and vice versa, and a flatten layer was used to convert the images into vector features.

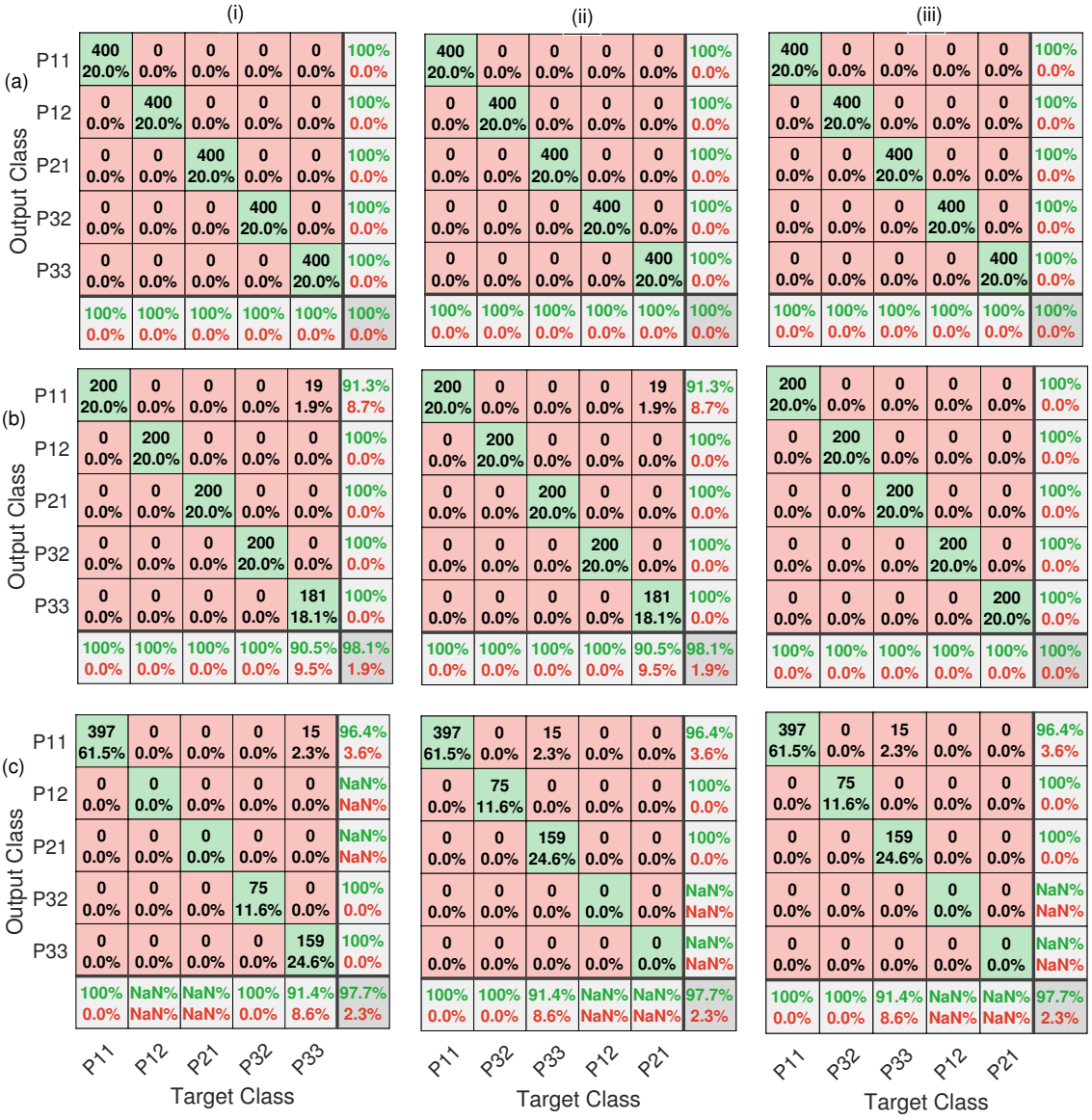


Figure 22: Confusion matrices for MorFeat2 features based (i) MLP (ii) SVM and (iii) LSTM networks on (a) merged training data (b) simulation test data and (c) experimental test data. The networks are seen to append 'NAN' to the classes unavailable in the experimental data.

The results of the direct image-based classification models are presented in Tables 10 and 11 for the vibro-impact images at $[128 \times 128 \times 1]$ and $[50 \times 50 \times 1]$, respectively. Resulting confusion matrices showed that most of the networks deficiency came from the misclassification between P-3-3 images, P-3-2 images and sometimes P-1-1 images. It is observed that using images of smaller sizes did improved the performance of CNN and tuned SAE networks on both simulation and experimental data. The smaller sized images also showed improved performance for simulation data using the LSTM network but not for the experimental data. The improved performance of the smaller sized images may be due to the reduced network computations and reduced memory usage. For the CAE, the network performance was better for the larger size images from both simulation and experiment. A summary of the reported performances is presented in Fig. 25. Based on the performances of the networks with respect to input

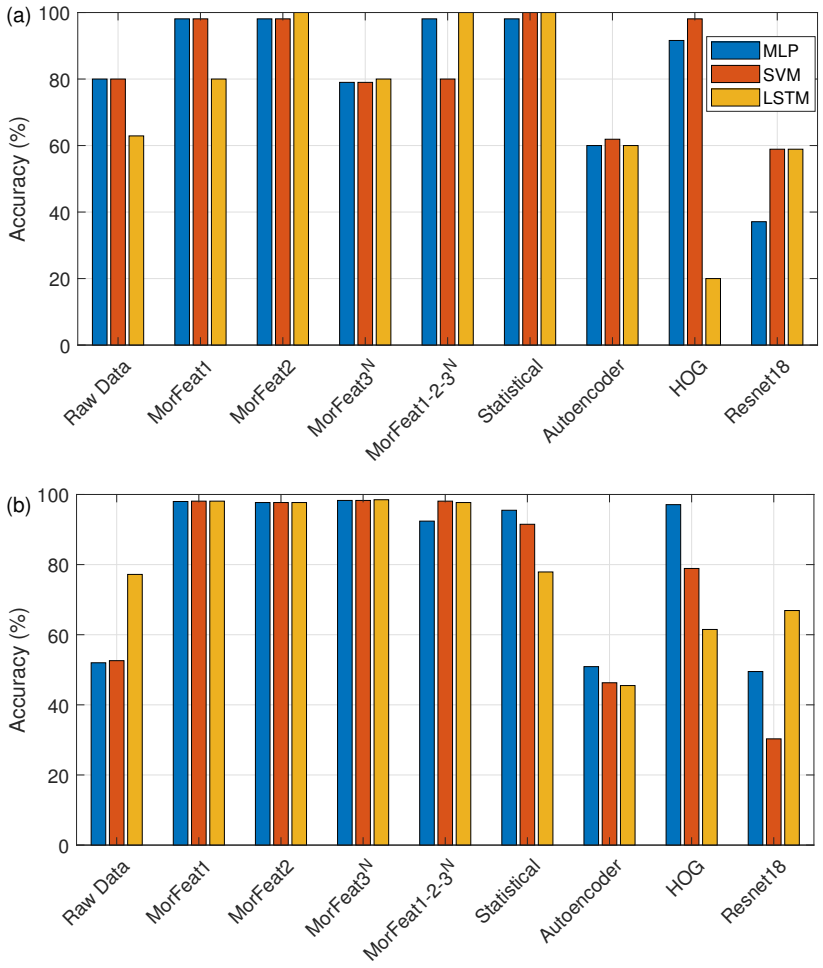


Figure 23: (Colour online) Graphical summary of all the developed network performances on (a) simulated and (b) experimental test data.

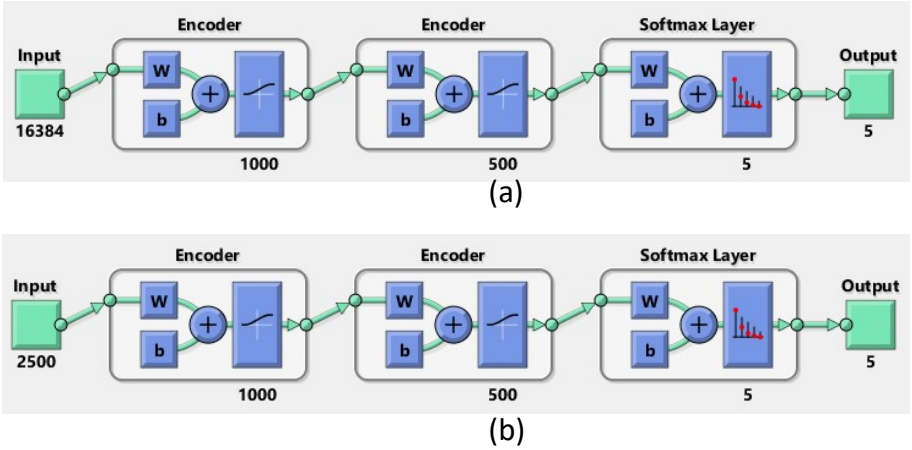


Figure 24: Stacked-autoencoders layers for (a) [128 × 128 × 1] and (b) [50 × 50 × 1] image data classification.

image sizes, it could be suggested that deducing the smallest image size wherein the discriminating image

features are still preserved is essential for building image based classification networks. This way network computations are kept to the minimal and enough memory is available for hyper-parameter tuning in order to optimise their performances.

Table 10: Results of image based classification models for [128 × 128 × 1] size images: five-class data, where 'Sim.' and 'Exp.' denote the simulated and experimental out-of-sample cross-validation data, respectively.

| Features | Training (%) | Testing (%) | |
|---|---|---|---|
| | | Sim. | Exp. |
| SAE (without fine tuning) | 100 | 80.0 | 53.1 |
| SAE (with fine tuning) | 100.0 | 80.0 | 55.9 |
| CAE | 100 | 97.1 | 100 |
| CNN | 100 | 99.0 | 85.4 |
| LSTM | 100 | 78.1 | 92.0 |

Table 11: Results of image based classification models for [50 × 50 × 1] size images: five-class data, where 'Sim.' and 'Exp.' denote the simulated and experimental out-of-sample cross-validation data, respectively.

| Features | Training (%) | Testing (%) | |
|---|---|---|---|
| | | Sim. | Exp. |
| SAE (without fine tuning) | 79.2 | 70.0 | 72.9 |
| SAE (with fine tuning) | 100.0 | 97.1 | 59.3 |
| CAE | 100 | 80.0 | 95.8 |
| CNN | 100 | 100 | 100 |
| LSTM | 100 | 97.1 | 87.8 |

## 5. Conclusions

Having concluded on the urgent need for real-time characterisation of downhole impact motions for the VID system, the potential of neural networks to learn complex nonlinear relationships has been explored to characterise impact motions from drill-bit acceleration data. Considering the large data requirement of neural networks and the need to validate their application with real life data, scarce experimental data have been augmented with mathematically simulated data to develop vibro-impact motion classification models.

With the aid of feature extraction, some of the trained networks were found to have significantly performed well in distinguishing between the different categories of impact motions. Aside their low memory usage, the MLP, SVM and BiLSTM networks showed consistent high performances using the waveform morphological features, MorFeat1, MorFeat2 and MorFeat1-2-$3^{N}$. The networks also showed high performances using the statistical features except for LSTM on experimental statistical features where the accuracy was 77.9%. The MLP networks also showed good performance using the HOG features. Irrespective of the input image size, CAE, CNN and LSTM image-based networks showed accuracies ranging between 78%-100% for both simulation and experiment data.

The consistent performance of these aforementioned feature-based networks on both simulation and experimental testing data compared to the raw data-based networks shows that designing rich discriminating features is important and preferable, when augmenting experimental data with simulation data. This is because such features tend to be quite consistent for both the simulation and experimental data, thus making them the first point of consideration for an on-line impact motion categorisation system for optimising the VID system. Such a development will not only increase the ROP and the life-span of the

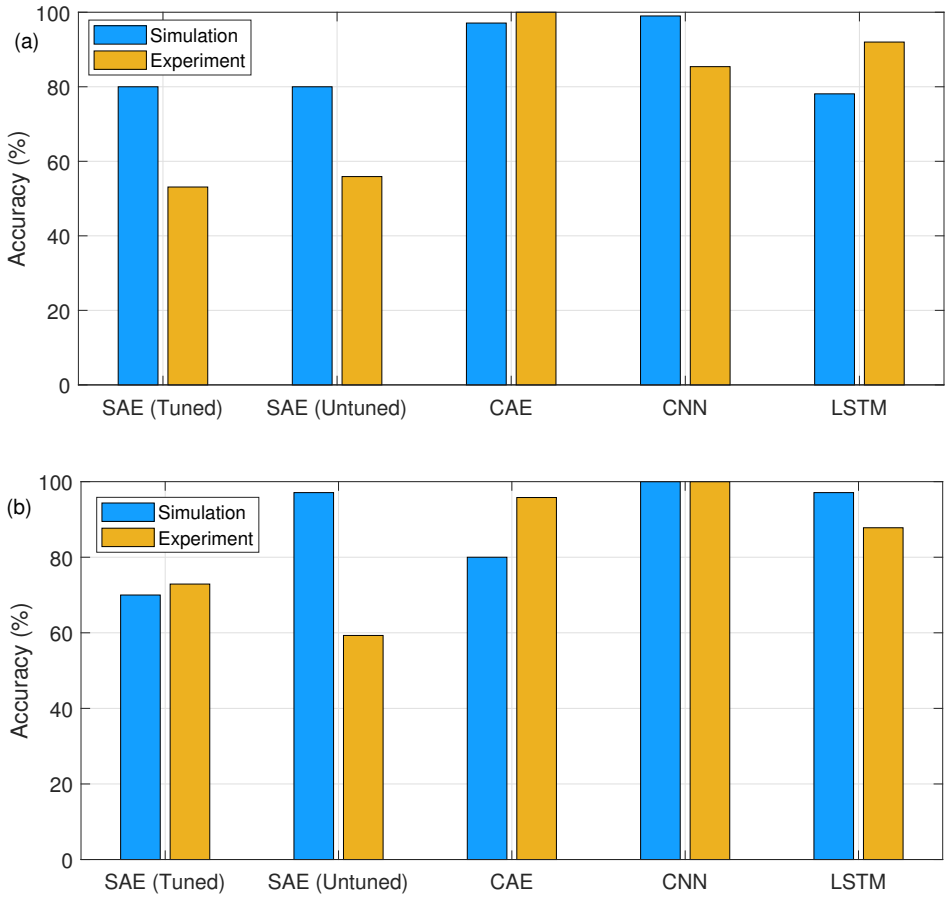


Figure 25: (Colour online) Graphical summary of the performances of the image based networks on (a) [128 x 128 x 1] and (b) [50 x 50 x 1] sized images.

drill-bit, but will also minimise energy wastage and drill-site emissions as every energy input is used to achieve greater work done when the system is consistently kept under optimal impact modes.


### Acknowledgements

Mr K. O. Afebu acknowledges the financial support from the Petroleum Technology Development Fund (PTDF) of Nigeria for his PhD scholarship (PTDF/ED/PHD/AKO/1080/17).


### Compliance with ethical standards

### Conflict of interest

The authors declare that they have no conflict of interest concerning the publication of this manuscript.

### Data accessibility

The data sets generated and analysed during the current study are available from the corresponding author on reasonable request.

## References


[1] L. Tang, X. Zhu, and H. Lin, "Effects of axial impact load on the dynamics of an oilwell drillstring," *Advances in Mechanical Engineering*, vol. 11, no. 3, p. 1687814019836876, 2019.

[2] Y. Melamed, A. Kiselev, M. Gelfgat, D. Dreesen, and J. Blacic, "Hydraulic hammer drilling technology: developments and capabilities," *J. Energy Resour. Technol.*, vol. 122, no. 1, pp. 1–7, 2000.

[3] H. G. Pyles and C. D. Albright, "Rotary percussion drill," Nov. 9 1976. US Patent 3,990,522.

[4] E. Pavlovskaia, D. C. Hendry, and M. Wiercigroch, "Modelling of high frequency vibro-impact drilling," *International Journal of Mechanical Sciences*, vol. 91, pp. 110–119, 2015.

[5] M. Wiercigroch, "Resonance enhanced drilling: method and apparatus. patent no," *WO2007141550*, 2007.

[6] M. Wiercigroch, J. Wojewoda, and A. Krivtsov, "Dynamics of ultrasonic percussive drilling of hard rocks," *Journal of Sound and Vibration*, vol. 280, no. 3-5, pp. 739–757, 2005.

[7] L. F. Franca, "A bit–rock interaction model for rotary–percussive drilling," *International Journal of Rock Mechanics and Mining Sciences*, vol. 48, no. 5, pp. 827–835, 2011.

[8] A. Abouzeid and G. Cooper, "The use of a drilling simulator to optimize a well drilling plan," *Transactions-Geothermal Resources Council*, pp. 79–84, 2001.

[9] M. S. Ramsey, *Practical Wellbore Hydraulics and Hole Cleaning: Unlock Faster, More Efficient, and Trouble-free Drilling Operations*. Gulf Professional Publishing, 2019.

[10] E. Pavlovskaia, M. Wiercigroch, and C. Grebogi, "Modeling of an impact system with a drift," *Physical Review E*, vol. 64, no. 5, p. 056224, 2001.

[11] L. F. P. Franca and H. I. Weber, "Experimental and numerical study of a new resonance hammer drilling model with drift," *Chaos, Solitons & Fractals*, vol. 21, no. 4, pp. 789–801, 2004.

[12] M. Liao, Y. Su, and Y. Zhou, "Oscillation reconstruction and bifurcation analysis of a drillbit–rock vibro-impact system," *International Journal of Bifurcation and Chaos*, vol. 27, no. 01, p. 1750013, 2017.

[13] M. Liao, Y. Liu, J. P. Chávez, A. S. Chong, and M. Wiercigroch, "Dynamics of vibro-impact drilling with linear and nonlinear rock models," *International Journal of Mechanical Sciences*, vol. 146, pp. 200–210, 2018.

[14] J. Ing, E. Pavlovskaia, M. Wiercigroch, and S. Banerjee, "Experimental study of impact oscillator with one-sided elastic constraint," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 366, no. 1866, pp. 679–705, 2008.

[15] M. Liao, M. Wiercigroch, M. Sayah, and J. Ing, "Experimental verification of the percussive drilling model," *Mechanical Systems and Signal Processing*, vol. 146, p. 107067, 2021.

[16] J. P. Chávez, E. Pavlovskaia, and M. Wiercigroch, "Bifurcation analysis of a piecewise-linear impact oscillator with drift," *Nonlinear Dynamics*, vol. 77, no. 1, pp. 213–227, 2014.

[17] D. Wang, D. Liu, Y. Zhang, and H. Li, "Neural network robust tracking control with adaptive critic framework for uncertain nonlinear systems," *Neural Networks*, vol. 97, pp. 11–18, 2018.

[18] K. O. Afebu, Y. Liu, E. Papatheou, and B. Guo, "Lstm-based approach for predicting periodic motions of an impacting system via transient dynamics," *Neural Networks*, vol. 140, pp. 49–64, 2021.

[19] K. O. Afebu, Y. Liu, and E. Papatheou, "Application and comparison of feature-based classification models for multistable impact motions of percussive drilling," *Journal of Sound and Vibration*, vol. 508, p. 116205, 2021.

[20] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," in *2014 Science and Information Conference*, pp. 372–378, IEEE, 2014.

[21] J. Hoffmann, Y. Bar-Sinai, L. M. Lee, J. Andrejevic, S. Mishra, S. M. Rubinstein, and C. H. Rycroft, "Machine learning in a data-limited regime: Augmenting experiments with synthetic data uncovers order in crumpled sheets," *Science advances*, vol. 5, no. 4, p. eaau6792, 2019.

[22] Y. Yan, Z. Tan, and N. Su, "A data augmentation strategy based on simulated samples for ship detection in rgb remote sensing images," *ISPRS International Journal of Geo-Information*, vol. 8, no. 6, p. 276, 2019.

[23] C. Tang, S. Vishwakarma, W. Li, R. Adve, S. Julier, and K. Chetty, "Augmenting experimental data with simulations to improve activity classification in healthcare monitoring," in *2021 IEEE Radar Conference (RadarConf21)*, pp. 1–6, IEEE, 2021.

[24] O. T. Inan, L. Giovangrandi, and G. T. Kovacs, "Robust neural-network-based classification of premature ventricular contractions using wavelet transform and timing interval features," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 12, pp. 2507–2515, 2006.

[25] Z. Pápai and T. Pap, "Analysis of peak asymmetry in chromatography," *Journal of Chromatography A*, vol. 953, no. 1-2, pp. 31–38, 2002.

[26] D. Jha, A. G. Kusne, R. Al-Bahrani, N. Nguyen, W.-k. Liao, A. Choudhary, and A. Agrawal, "Peak area detection network for directly learning phase regions from raw x-ray diffraction patterns," in *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2019.

[27] T. Lambrou, P. Kudumakis, R. Speller, M. Sandler, and A. Linney, "Classification of audio signals using statistical features on time and wavelet transform domains," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, vol. 6, pp. 3621–3624, IEEE, 1998.

[28] A. Arnaout, B. Alsallakh, R. Fruhwirth, G. Thonhauser, B. Esmael, and M. Prohaska, "Diagnosing drilling problems using visual analytics of sensors measurements," in *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pp. 1750–1753, IEEE, 2012.

[29] B. Esmael, A. Arnaout, R. K. Fruhwirth, and G. Thonhauser, "A statistical feature-based approach for operations recognition in drilling time series," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 5, pp. 454–61, 2015.

[30] G. L. Sciuto, C. Napoli, G. Capizzi, and R. Shikler, "Organic solar cells defects detection by means of an elliptical basis neural network and a new feature extraction technique," *Optik*, vol. 194, p. 163038, 2019.

[31] P. Jadhav, G. Rajguru, D. Datta, and S. Mukhopadhyay, "Automatic sleep stage classification using time–frequency images of cwt and transfer learning using convolution neural network," *Biocybernetics and Biomedical Engineering*, vol. 40, no. 1, pp. 494–504, 2020.

[32] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[33] J. Zhang, S. Shan, M. Kan, and X. Chen, "Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment," in *European Conference on Computer Vision*, pp. 1–16, Springer, 2014.

[34] M. Hazgui, H. Ghazouani, and W. Barhoumi, "Genetic programming-based fusion of hog and lbp features for fully automated texture classification," *The Visual Computer*, pp. 1–20, 2021.

[35] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 806–813, 2014.

[36] A. A. Barbhuiya, R. K. Karsh, and R. Jain, "Cnn based feature extraction and classification for sign language," *Multimedia Tools and Applications*, vol. 80, no. 2, pp. 3051–3069, 2021.

[37] M. H. Beale, M. T. Hagan, and H. B. Demuth, *Deep Learning Toolbox User's Guide*.

[38] *ImageNet Large Scale Visual Recognition Challenge*, 2015 (accessed 31 October 2020).

[39] S. Savalia and V. Emamian, "Cardiac arrhythmia classification by multi-layer perceptron and convolution neural networks," *Bioengineering*, vol. 5, no. 2, p. 35, 2018.

[40] M. Darbor, L. Faramarzi, and M. Sharifzadeh, "Performance assessment of rotary drilling using nonlinear multiple regression analysis and multilayer perceptron neural network," *Bulletin of Engineering Geology and the Environment*, vol. 78, no. 3, pp. 1501–1513, 2019.

[41] C. M. Bishop *et al.*, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[42] F. Al-Shargie, T. B. Tang, N. Badruddin, and M. Kiguchi, "Towards multilevel mental stress assessment using svm with ecoc: an EEG approach," *Medical & Biological Engineering & Computing*, vol. 56, no. 1, pp. 125–136, 2018.

[43] Mathworks, *Statistics and Machine Learning Toolbox User's Guide*, 2020.

[44] E. L. Allwein, R. E. Schapire, and Y. Singer, "Reducing multiclass to binary: A unifying approach for margin classifiers," *Journal of Machine Learning Research*, vol. 1, no. Dec, pp. 113–141, 2000.

[45] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.