Journal Pre-proof

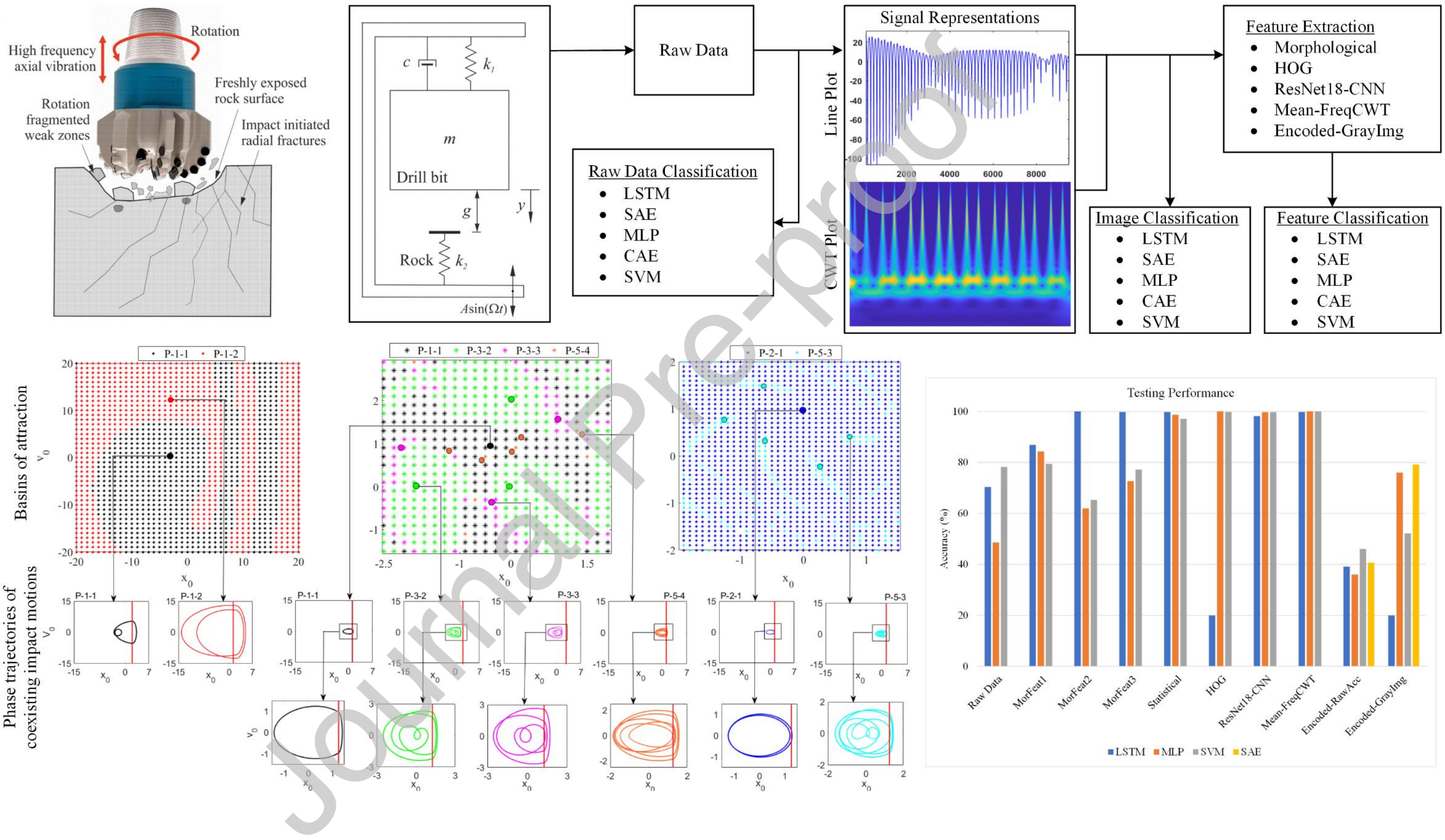
Application and comparison of feature-based classification models for multistable impact motions of percussive drilling

Kenneth Omokhagbo Afebu, Yang Liu, Evangelos Papatheou

PII:	S0022-460X(21)00277-7
DOI:	https://doi.org/10.1016/j.jsv.2021.116205
Reference:	YJSVI 116205
To appear in:	*Journal of Sound and Vibration*
Received date:	5 November 2020
Revised date:	28 April 2021
Accepted date:	8 May 2021

Highlights

- Prediction of coexisting impact motions of a percussive drilling system is studied.
- Various feature extractions are used to learn system's complex non-linearity.
- Appropriate feature extraction and suitable network selection are essential.
- Effective features and networks are obtained via simulation study and comparison.
- The work provides a means of designing a vibro-impact motion classification model.



Rotation
High frequency axial vibration
Freshly exposed rock surface
Rotation fragmented weak zones
Impact initiated radial fractures
c
k_1
m
Drill bit
g
y
Rock
k_2
$A\sin(\Omega t)$
Raw Data
Raw Data Classification
- LSTM
- SAE
- MLP
- CAE
- SVM
Signal Representations
Line Plot
CWT Plot
Feature Extraction
- Morphological
- HOG
- ResNet18-CNN
- Mean-FreqCWT
- Encoded-GrayImg
Image Classification
- LSTM
- SAE
- MLP
- CAE
- SVM
Feature Classification
- LSTM
- SAE
- MLP
- CAE
- SVM
Basins of attraction
Phase trajectories of coexisting impact motions
P-1-1
P-1-2
P-3-2
P-3-3
P-5-4
P-2-1
P-5-3
x_0
v_0
Testing Performance
Accuracy (%)
Raw Data
MorFeat1
MorFeat2
MorFeat3
Statistical
HOG
ResNet18-CNN
Mean-FreqCWT
Encoded-RawAcc
Encoded-GrayImg
LSTM
MLP
SVM
SAE

Application and comparison of feature-based classification models for multistable impact motions 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

Dynamics of the bit-rock interaction under percussive drilling often encounter multistability that produces coexisting impact motions for a wide range of drilling conditions. Some of them may be detrimental to its performance as it cuts through the inhomogeneous rock layers. A necessary mitigation is the ability to distinguish between coexisting impact motions in order to maintain a high-performance drilling. For this purpose, dynamical responses of a vibro-impact system mimicking the bit-rock interaction of percussive drilling were explored in this study by using machine learning techniques. As a fundamental approach of improving machine learning, hand-crafted and automatic feature extractions were carried out. Simulation results show that extracting appropriate features and using a suitable network are essential for characterising the vibro-impact motions. Extracting statistical, histogram of gradient, continuous wavelet transform and pre-trained convolutional network features are effective and less computationally intensive. With their high accuracies, they become the first point of consideration when designing the classification model for multistable vibro-impact motions of percussive drilling.

Keywords: Percussive drilling; Bit-rock interaction; Vibro-impact dynamics; Multistability; Machine learning.

1. Introduction

Vibrating machinery, such as milling [1], downhole drilling [2–4], produce signals that are often analysed not only to ascertain their health condition, but also their performance and productivity, e.g. gears [5] and cutting [6], especially when such vibrations are normal to the machine operation [7–10]. As illustrated in Fig. 1(a), the vibro-impact drilling systems introduced in [11, 12] use adjustable high frequency oscillations to generate high amplitude axial vibrations at the bit-rock contact interface. These are used alongside the drill-string's rotation to propagate fractures within the drilled rock layer. Previous studies on the resulting bit-rock impact motions have shown that some periodic motions are more efficient for the drill-bit in cutting through the rock layers compared to others [12–15]. This finding initiated the work carried out by Afebu *et al.* [10] in which an attempt was made at developing a neural network-based optimisation strategy for percussive drilling via multistable motion classification. A bi-directional Long-Short Term Memory (LSTM) network was used as a deep learning model to differentiate between two coexisting periodic motions. However, due to the inhomogeneity that often characterise downhole rock layers, wider categories of impact motions are often encountered. This, thus makes it necessary to

[*] 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)

extend the introduced classification model [10] to include more categories of coexisting motions. In this present study, five categories of impact motions which were simulated from three basins of attraction using an impacting system shown in Fig. 1(b) and categorised using machine learning techniques.

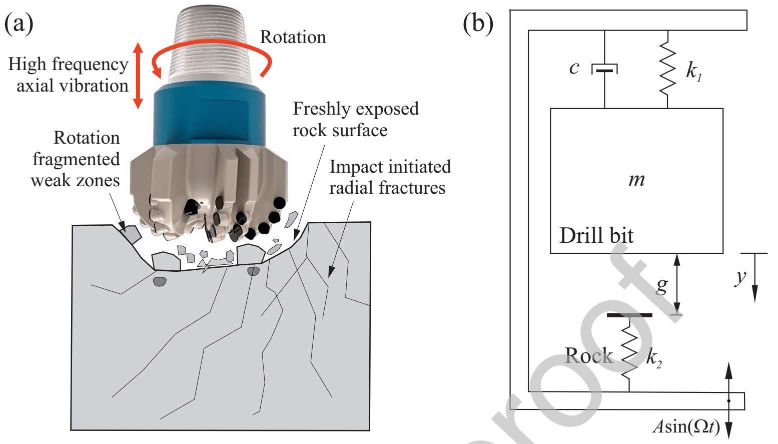

(a)
Rotation
High frequency axial vibration
Freshly exposed rock surface
Rotation fragmented weak zones
Impact initiated radial fractures
(b)
c
k1
m
Drill bit
g
y
Rock
k2
Asin(Ωt)

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

Developing a robust and less erroneous data-driven pattern recognition algorithm for classifying different impact motions of percussive drilling requires several pre-processing techniques, one of which is data reduction [16]. Data reduction approaches are usually applied to obtain a reduced representation of the original data while still retaining their temporal information. In neural network-based models, data reduction helps to create a more manageable representation of the original data. This often reduces the complexities and storage space of the data, and also minimises the computational cost in terms of the simplicity of the network architecture, its training time and accuracy. Feature selection and feature extraction are two main strategies for data reduction [17, 18]. Given an original data set X with dimension D, during feature selection, a subset X_s with dimension D_s is selected such that $X_s \subset X$ and $|D_s| \ll |D|$ while the remaining redundant $D - D_s$ dimensions are discarded. Feature extraction on the other hand uses functional mapping to produce a new set of data Y of dimension D_e from the original data X,

$$Y = f(X) : X \to Y. \tag{1}$$

Unlike feature selection, the extracted data set is not a subset of the original data ($Y \not\subseteq X$). Hence, the semantics of the original data are not preserved, and the dimensionality of the extracted new data D_e can either be higher or lower depending on the mapping function of the extraction technique. However, for the purpose of machine learning, the selected or extracted features should have better discriminating power compared to the original data set.

In view of the above, this study embarks on different feature extraction tactics for data reductions. The feature extraction techniques were first used to extract and select features from the measured impact motion signals which were later used as input data into pattern recognition models. However, due to the nonlinear dynamics and the similarities exhibited by the multistable impacting motions, designing a robust feature-based classification model is far from being an easy task, hence various feature harvesting tactics were investigated. These include (a) supervised extraction of hand-crafted features from the raw data, time-amplitude and time-frequency representation, and (b) unsupervised extraction of gradient-based and neural network-based features from image representation of the signals. The vibration images were generated as 2D colour images using continuous wavelet transform (CWT) while the unsupervised feature extraction was carried out using the histogram of oriented gradients (HOG), a pre-trained convolutional neural networks (CNN) and stacked autoencoders. The performance self-supervised learning networks, such as stacked autoencoders (SAE), convolutional autoencoders (CAE) and LSTM on the images was also explored.

The remainder of this paper is structured as follows. Section 2 presents the preliminaries and theories of the feature extraction methods and the network classifiers used in this study. The generation of vibro-impact motions and extraction of vibro-impact features are described in Section 3 while the results of the feature-based learning and classification models are presented in Section 4. The paper is rounded-off with some discussions and conclusions in Section 5.

2. Preliminaries and theories

2.1. Feature extraction models

The analysis of data for feature selection and extraction, most especially time series data, can be done either in time domain, frequency domain or time-frequency domain [19, 20]. Features are distinctive characteristic representations or structural measurements extracted from segments of data. Desirable features are usually those that capture the overall trend of the data while minimising loss of information. Such features can be manually crafted or automatically derived from the original data representation. Manual crafting of features can be time consuming and cumbersome, but to get around this limitation, the use of unsupervised automatic feature extraction methods have been proposed and utilised for feature extraction. Sections 2.1.1 and 2.1.2 illustrate manually extracted features and Sections 2.1.3-2.1.6 describe automatically extracted features.

2.1.1. Morphological time-interval features

In the time domain, intrinsic temporal information in the form of features can be hand-crafted from structural components of a signal waveform. Peaks and troughs are readily observable structural components of most waveforms which have been used in studies involving chromatographic signal [21], proteomic data [22], electrocardiogram [23], electroencephalography [24], electrophysiology [25] and X-ray diffraction signal analysis [26]. The attributes defining the analysed feature can be represented both in value and position within the time series. Peaks can be described based on their amplitude, locations, width and distance apart. These properties are useful for describing the magnitude and time of occurrence of specific events within time series. For example, Inan *et al.* [27] used morphological features as an input of a neural network classifier for classifying heart beats.

2.1.2. Statistical features

Statistical features, especially the descriptive categories, seek to create a compact representation of raw data in a summarised way that may allow for pattern recognition. The compact representations contain sets of statistical measures calculated from raw data that often describe the central tendency and spread of time series data. Previous usage include audio signal classification [28], critical drilling conditions recognition from drilling time series [29], drilling time series classification [30] and human activity recognition [31]. Table 1 describes the statistical measures explored in this study which have been crafted with respect to previous works listed in [31]. No further feature selection was carried out on the chosen statistical measures as doing that may annul the contribution of the feature interaction within and between each of the classes [32].

Table 1: Statistical measures used as statistical features.

S/N	Statistical Feature	S/N	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	Average cumulative maximum element	23	Skewness factor
12	Average cumulative minimum element		

2.1.3. Continuous wavelet transform

Time-frequency representation of data has proved useful in isolating temporal and localised information embedded in the data. Aside short time Fourier transform [33], Cohen's class [34], WignerVille transform [35], ChoiWilliams distribution [36] and empirical mode decomposition [37], CWT, a form of wavelet transform [38], is common method used in decomposing univariate non-stationary time series into a two-dimensional timefrequency space, thus enabling the frequency component of the signal to be observed as a function of time. CWT uses finer discretisation to produce more redundant representation where a 1-by-N samples of data gets transformed to M-by-N matrix of coefficients with "M" representing the number of scales. The main idea of CWT is to apply a selected mother wavelet as a bandpass filter to the time series. The Morlet wavelet was selected in this study being the most used mother wavelet for CWT decomposition [39]. The mother wavelet is shifted at small intervals on the signal along its x-axis, and the correlation coefficient is calculated for each shift. The procedure is repeated at different dilation or scaling factors on the y-axis to capture the frequency-based variation. The Morlet wavelet can be expressed as

$$\psi(\tau) = \frac{1}{\sqrt[4]{\pi}} \left(e^{i\omega_o \tau} - e^{-\dfrac{\omega_o^2}{2}} \right) e^{-\dfrac{\tau^2}{2}}, \tag{2}$$

where ω_o and τ are nondimensional frequency and time, respectively, and $e^{-\frac{\omega_o^2}{2}}$ is the correctional term that corrects for the non-zero mean of the complex sinusoidal function. It becomes negligible for $\omega_o \geq 5$, thus giving

$$\psi(\tau) = \frac{1}{\sqrt[4]{\pi}} e^{i\omega_o \tau} e^{-\dfrac{\tau^2}{2}}, \qquad \omega_o \geq 5. \tag{3}$$

The CWT of an N-sampled time series, $(x_i, i = 1, 2, 3, , N)$, with a uniform time-step δt is given as

$$T(a, b) = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} x(\tau) \psi * \left(\frac{t - b}{a} \right) \delta t, \tag{4}$$

where $\psi * (\tau)$ is the complex conjugate of the mother wavelet function $\psi(\tau)$, a and b are the dilation and location parameters of the wavelet. The signal energy at a specific a scale and b location is given as a two-dimensional wavelet energy density function given as

$$E(a,b) = |T(a,b)|^2, \tag{5}$$

which is referred as the scalogram. By integrating the scalogram at a specific scale a and across the b location, the total energy contained in the signal can be recovered as

$$E(a) = \frac{1}{C_g}\int_{-\infty}^{+\infty} |T(a,b)|^2 \delta b, \tag{6}$$

where C_g is the admissibility constant given as:

$$C_g = \int_{0}^{+\infty} \frac{|\hat{\psi}(f)|^2}{f}\delta f < \infty, \tag{7}$$

and $\hat{\psi}(f)$ is the Fourier transform of $\psi(\tau)$ given as

$$\hat{\psi}(f) = \int_{-\infty}^{+\infty} \psi(\tau) e^{-i(2\pi f)\tau}\delta\tau. \tag{8}$$

Generally, CWT computation is carried out over a finely discretised time-frequency grid, where the b location is discretised at the sampling interval and the a scale at a logarithmic scale. This way the wavelet transform is approximated for each time step over a range of wavelet scales thus resulting in a relatively large output of higher dimension and with redundant information. The output often requires dimension reduction to remove the redundancy and reduce computational cost during classification [40]. Selected band of wavelet coefficients can be saved in their raw vector form or as scalograms representing vibration images that can be further analysed using image processing techniques.

2.1.4. Histogram of Oriented Gradient

HOG is a principal tool in computer vision and image processing which has been widely adopted for feature extraction [41]. It relies on the idea that the local textures of an image can be characterised by series of histograms of local intensity gradients describing the magnitude and direction of change in intensity of the image. Firstly, a gray-to-scale normalisation is performed on the images to reduce illumination variances that keeps them with the same intensity range. Secondly, the image is divided into multiple cells of m-by-m pixels. For each pixel in each of the cells, the spatial gradient information for a pixel at location (x, y) is given as

$$\begin{cases} G_y = I(x, y+1) - I(x, y-1), \\ G_x = I(x+1, y) - I(x-1, y)t, \end{cases} \tag{9}$$

where G_y and G_x represent the vertical and horizontal directions, and the magnitude and orientation are obtained as

$$M = \sqrt{G_y^2 + G_x^2}, \tag{10}$$

and

$$\theta = \arctan\left(\frac{G_y}{G_x}\right), \tag{11}$$

respectively. Thirdly, the HOG for each cell is constructed by concatenating the magnitudes by their orientations. The resulting outcome is an assemblage of L1-normalized vectors in a single 1-D vector array representing the integral HOG descriptors. The length of resulting HOG features is dependent on the image size and the extraction function parameters.

2.1.5. Convolutional Neural Network

CNN form part of the major deep neural networks that learn both linear and nonlinear transformations directly from images. CNN architecture can be divided into the input section consisting of an image input layer, an intermediate feature extraction section consisting of several hidden layers and the classification section consisting fully connected layer, a softmax layer for classification problems or a regression layer for regression problems and a final output layer. The hidden layers automatically reconstruct the input data with the intent of hierarchically extracting deep level features that are specific to the input images. The intermediate fully connected layers combine the features learnt by the previous layers to identify general patterns in the images while the last fully connected layer, containing same numbers of neurons as the number of classes, combines the feature to arrive at a class for the image. The output layer which also has the same number of neurons as the number of classes and uses loss functions such as mean square error, softmax cross entropy and sigmoid cross entropy to estimate loss (discrepancy) between the estimated and the actual. During training, the parameters are tuned towards minimising the loss. The spatial size of the output volume from a given convolutional layer is given as

$$O = \frac{W - K + 2P}{S} + 1, \tag{12}$$

where $O :=$ the output volume, $W :=$ the input volume size, $K :=$ the filter size, $P :=$ the padding and $S :=$ the stride. With the availability of robustly pre-trained CNNs, recent procedures of applying CNNs include (i) training a new CNN from the scratch and (ii) using pre-trained CNNs to extract features from analysed images. Extracted features are fed as input data into other classifier networks. Considering the huge data, time and system requirement for training a new CNN, the latter approach of transfer learning is easier and faster, and also serve as an unsupervised feature extraction procedure [42]. Hierarchical discriminating features are extracted from the images using convolutional operation while still maintaining the spatial correlation between pixels [43]. The procedures of using newly trained CNN and that of transfer learning were implemented in this study. Based on the classification accuracy, the relative prediction time and size on disk reported in [44], pre-trained Resnet18 network with 18 layers and trained on ImageNet database [45] were utilised here.

2.1.6. Autoencoders

Autoencoders, especially the SAE, are another form of neural network-based methods for unsupervised feature extraction. First introduced in late 1980s [46], they consist of an encoder and a decoder network. The output from autoencoders is usually of the same size as the input. In early usage, their application in data reduction was similar to that of principal component analysis [47] as linear feature extractors, although much more computationally intensive. The advent of nonlinear activation functions, pushed their usage to learning more complex nonlinear relationship [48]. The encoder part, $f(X)$, tries to map an original data X with D dimension into a simpler representation Y given as

$$Y = f(X) = s_f(WX + b_X), \tag{13}$$

whose dimension is D_e, such that $D_e \ll D$. The decoder part, $g(Y)$, on the other hand aims to reconstruct the original data X from the reduced data Y while minimising the differences between original data X and reconstructed data X' such that

$$X' = g(Y) = s_g(W'Y + b_Y), \tag{14}$$

where s_f and s_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 this difference.

Training an autoencoder entails tuning the model parameters, including W, b_X and b_Y, to a state denoted as Φ where they yield the least reconstruction errors between Y and X such that

$$\min_{\Phi} \frac{1}{N} \sum_{i}^{N} \|x_i - f_\Phi(x_i)\|_2^2, \tag{15}$$

where x_i is the ith data and N is the total numbers of data. The final step in training and using autoencoders is the process of fine-tuning to improve the network performance. During this process, the network, consisting of autoencoders and a softmax or regression layer is re-trained on labelled training data in a supervised manner. However, compared to other nonlinear feature extraction methods, autoencoders require higher computer memory and lots of time to train. Autoencoders are also used as unsupervised feature extractors when the auto-encoded features are used as input data into other networks. Higher level features can be extracted by stacking up multiple SAE, however, important lower level features may be omitted in this way. Recent development of autoencoders include the CAE currently used in image data analysis. Unlike the SAE which forces image features to be global (i.e. spanning through the entire visual field), the CAE takes into account the $2D$ structure of the image data. CAE architectures are similar to the conventional autoencoders only that their weights are shared among all locations in the input image data, thus preserving spatial locality of the learnt features [49].

2.2. Network classifiers

2.2.1. Multi-layer perceptron

Being universal approximators, Multi-layer perceptron (MLP) networks have been used in modeling nonlinear problems including classification [50] and regression [51] tasks. Assuming the input data x_i $(i = 1, 2, 3, , N)$, the network output y is defined in terms of a mapping function shown as below [52]

$$y = f_{output}\left(\sum_{j=1}^{M} W_j f_{hidden}\left(\sum_{j=1}^{M} W_{ji} x_i,\right) + W_o\right) \tag{16}$$

where N is the total number of input data, M is the number of hidden neurons, x_i is the ith input data, W_{ij} is the weight parameter between the ith input data and jth hidden neuron and W_j is the weight parameter between the jth hidden neuron and the output neuron. The activation function f_{output} is given as a sigmoid function for classification problems and as a linear activation function for regression while f_{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 networks prediction (y) and the actual target (tt) is defined as a cross-entropy error,

$$E_{xr} = -\sum_{p=1}^{P} \{tt_p \ln(y_p) + (1 - tt_p) \ln(1 - y_p)\}. \tag{17}$$

2.2.2. Support Vector Machines

Support vector machines (SVMs) initially developed for binary (two-class) classification problems analyses input data set alongside their corresponding labels to establish an optimal separating hyperplane between the two classes. They are adapted for multi-class classification by creating multiple binary-class SVMs for pair-wise combination of classes. The binary pair-wise classifiers are then combined to create the final multi-class classifier using any of the existing combination strategies as found in [53]. In this study, the Error-Correcting Output Code (ECOC) combination technique [54] as presented in MATLAB software [55] was used in fusing multiple binary SVM decisions. This operation involves a coding and a decoding process using a matrix of codewords (usually $\{1, 0, 1\}$) of size $k \times l$ [56], where k is the number of classes and l is the numbers of SVM binary classifiers or the number of codewords codifying each class given as

$$l = \frac{k(k-1)}{2}. \tag{18}$$

Matrix C shown in Table 2 is a code matrix for a five class problem using ten learners as carried in this present study. An instance belonging to class k_i is positive for a classifier l_j if and only if $C_{ij} = 1$ and it is negative if and only if $C_{ij} = -1$. For a classifier, $C_{ij} = 0$ means that the class was not used in training the classifier. Table 2 indicates that the multi-class SVM trains its first four binary learners ($l_1 - l_4$) using observations that are classified as class k_1 as it returned them as positive class against k_2, k_3, k_4 and k_5 which are respectively returned as negative class.

Table 2: 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

Earlier studies have shown multi-class SVMs to be very accurate in their performance, however, they were found to be less effective for online application [57]. A major contributor to this drawback is the amount of computation needed to done within a very short period of time, especially when it involves input data that can easily scale-up as found in sequential text pattern recognition and phrase text classification [58, 59]. To circumvent this problem, recent studies have adopted strategies such as feature extraction to maximise inter-class margin and also minimise computational time and memory requirement [57, 60, 61]. In line with this, minimal and highly discriminating features were extracted and explored in building multiclass SVM models in this present study.

2.2.3. Long-Short Term Memory classifier

LSTM networks are improved forms of recurrent neural networks having memory cells and gates in place of the usual inter-connecting hidden neurons. This improvement makes the networks immune to the vanishing gradient problem when learning from long time series data [62]. At each time step t in the time series, the network calculates a hidden vector h_t and a memory vector m_t which uses to update its state and output at that time-step and later carried to the next time-step. The calculations at a time-step are as follows.

$$\hat{i}_t = \sigma_g(W_i \hat{x}_t + R_i h_{t-1} + b_i), \tag{19}$$

$$\hat{f}_t = \sigma_g(W_f \hat{x}_t + R_f h_{t-1} + b_f), \tag{20}$$

$$\hat{g}_t = \sigma_g(W_g \hat{x}_t + R_g h_{t-1} + b_g), \tag{21}$$

$$\hat{o}_t = \sigma_g(W_o \hat{x}_t + R_o h_{t-1} + b_o), \tag{22}$$

$$m_t = \hat{f}_t \odot m_{t-1} + \hat{i}_t \odot \hat{g}_t, \tag{23}$$

$$h_t = \hat{o}_t \odot \tanh(m_t), \tag{24}$$

where $\hat{i}_t$, $\hat{f}_t$, $\hat{g}_t$ and $\hat{o}_t$ are state vectors of $[0, 1]$, σ_g and $\odot$ represent a sigmoid and an element-wise multiplication function, respectively. In the present study, three approaches were employed in using LSTMs for classifying the dynamical responses of the impacting system. These include (i) their direct application on the measured raw data, thereby exploiting their ability to self-extract temporal information from the data, (ii) their application on already extracted features from the raw data [10], and (iii) their application on image representations of the data. For the image data based LSTM model, the images are fed into the network as sequences of images. A sequence folding and a sequence unfolding layers are respectively used to convert the incoming sequence of images to image arrays and vice versa, and a flatten layer is used to convert the images into vector features.

3. Impact data and impact feature generation

Figure 2 illustrates a typical schematic of the stages followed in developing the vibro-impact motion classification model. This includes vibro-impact data collection, processing into 2D representation, feature selection and extraction, and finally, the development of raw data-based and feature data-based classifiers.

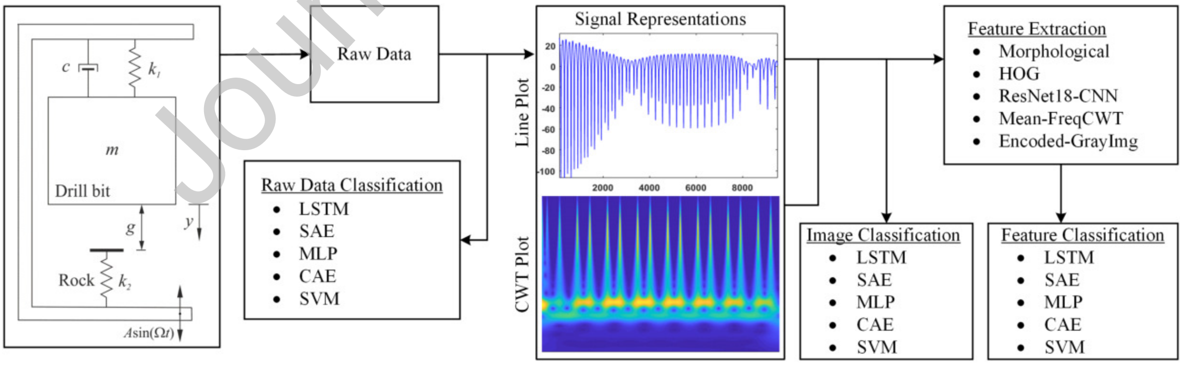


Figure 2: Schematic of the vibro-impact motion classification model.

3.1. Modelling of the vibro-impact motions

The impacting system shown in Fig. 1(b) is a one-degree-of-freedom system that mimics the bit-rock interaction for the vibro-impact drilling system. It is made up of a mass m connected to a rigid frame via a linear spring of stiffness k_1 and damping coefficient c, representing the drill-bit. Attached to the opposite end of the frame is the secondary linear spring with stiffness k_2 representing the rock media. k_2

can be varied to imitate the different types of rocks encountered while drilling. The equations of motion of the impacting system can be written as

$$\begin{cases} x' = v, \\ v' = a\omega^2 \sin(\omega\tau) - 2\zeta v - x - \beta(x-e)H(x-e), \end{cases} \tag{25}$$

where x' and v' are the differentiates of displacement and velocity of the drill-bit with respect to non-dimensional time τ, respectively, and $H(\cdot)$ stands for the Heaviside step function. The dimensionless form of the system parameters have been derived as follows.

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

where $y_0 > 0$ is an arbitrary reference distance, β is the stiffness ratio, ζ is the damping ratio, ω_n is the natural frequency, ω is the frequency ratio, a is the nondimensional external excitation amplitude, and e is the nondimensional gap between the mass and the secondary spring.

Basin of attraction is a closure of initial conditions whose long-time behaviour converges to a fixed position for a single attractor or multiple positions for coexisting attractors or multistable impact motions. It can be plotted using the corresponding initial displacements and velocities of the system. To generate multistable impact motions three basins of attraction, MBSN1, MBSN2 and MBSN3 shown in Fig. 3, were simulated using Eq. (25). It was implemented using the Runge-Kutta fourth-order method in MATLAB, and the system parameters were listed in Table 3. For simplicity, we use abbreviations P-n-m to describe periodic motion of the system, e.g. P-1-2 represents a period-one motion with two impacts per period of external excitation.

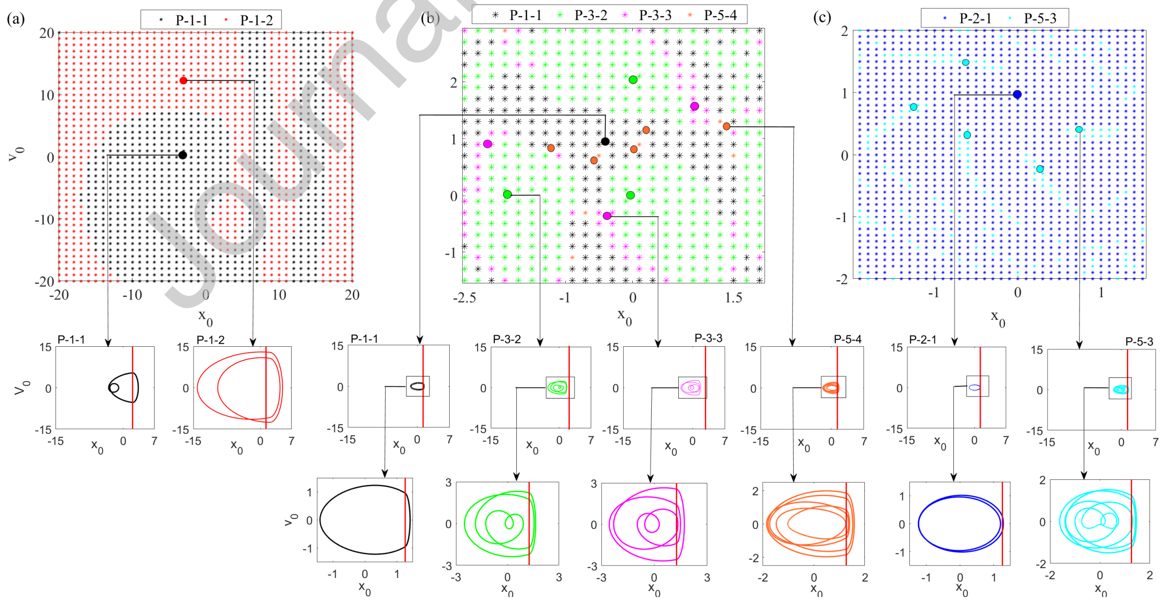


Figure 3: (Colour online) Basins of attraction for (a) MBSN1 (b) MBSN2 and (c) MBSN3 showing P-1-1 (Black), P-1-2 (Red), P-3-2 (Green), P-3-3 (Magenta), P-5-4 (Orange), P-2-1 (Blue) and P-5-3 (Cyan). The boundary of impact is denoted by the solid red line. The impact motions, P-1-1, P-1-2, P-2-1, P-3-2 and P-3-3, were extracted for training and testing the networks.

Table 3: Basin parameters for originating coexisting motions.

Parameters	MBSN1	MBSN2	MBSN3
ω	0.75	0.935	0.8063
a	5.6	0.7	0.7
β	18.27	29	29
ζ	0.01	0.01	0.01
g	2.1	1.26	1.26

Five categories of the impact motions, including period-one with one impact (P-1-1), period-one with two impacts (P-1-2), period-two with one impact (P-2-1), period-three with two impacts (P-3-2) and period-three with three impacts (P-3-3), were generated. In all, 600 samples of each impact category were generated, out of which 400 were used for training and 200 for testing. These impact motions can be represented by their measurable dynamic variables including displacement (x), velocity (v) and acceleration (v'). Among these variables, acceleration was chosen as the representative measurement for the impact motions, because (i) it is practically feasible to measure in real-life scenarios, and (ii) due to the abrupt decrease in velocity during an impact, acceleration measurements are more sensitive to impact actions compared to others.

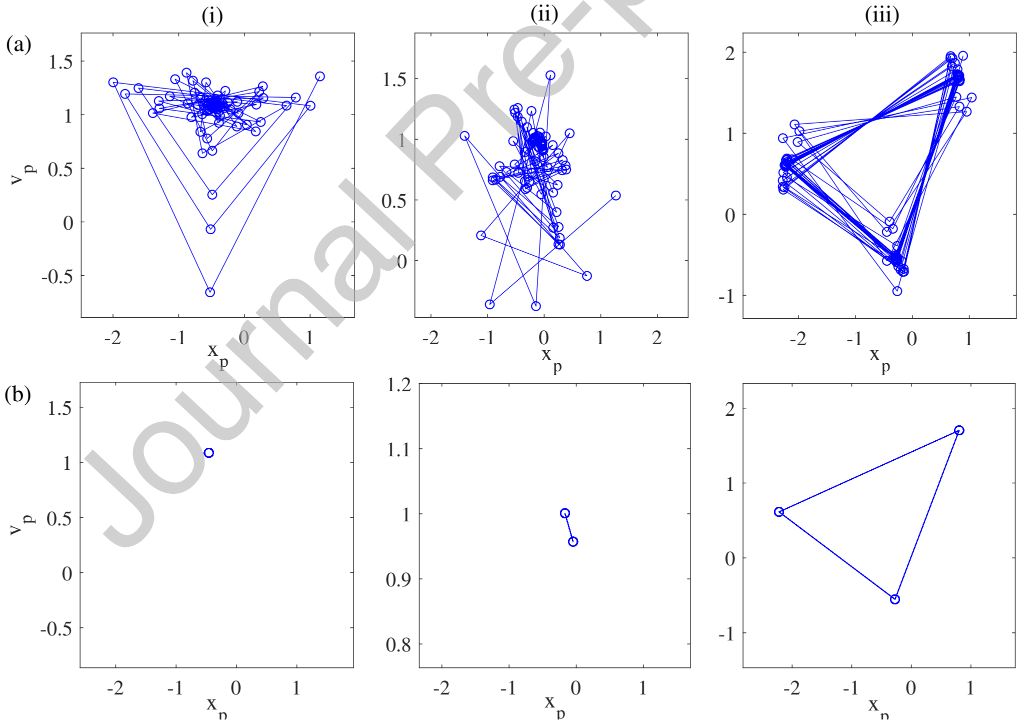


Figure 4: Oscillated points (Poincaré sections) on the phase plane for (a) stable, (b) transient, (i) period-one, (ii) period-two and (iii) period-three responses.

In this study, our aim is to explore the use of limited early but transient system responses for impact motions classification. By this, possible real life situation was mimicked and the need to wait for the system to reach stability before collecting data was minimised. To ensure that the used data are within the early transient periods of operation, a periods-to-stability calculation as suggested in [10] was carried out for each of the simulated time history data. During the transient periods of oscillation, the system erratically orbits around its points of attraction (Poincaré sections) in the x-v phase plane as presented in Fig. 4(a). But at stability, as shown in Fig. 4(b), the system maintains a relatively constant orbit of n points for a period-n motion. This implies that at stability, the distance orbited by the system in the phase plane becomes relatively constant and restricted to the path defined by the n-points. The number of periods expended before the aforementioned happens is taken as the periods-to-stability (T_i) of the system. In the present work, the estimated periods-to-stability for the training and testing data is presented in Fig. 5.

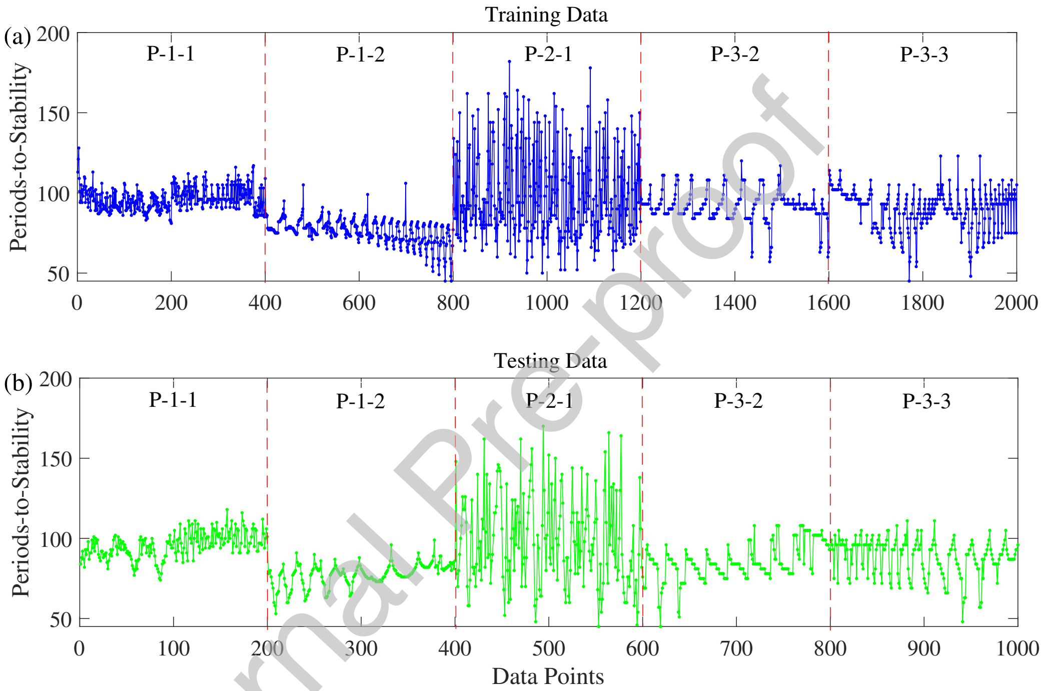


Figure 5: Estimated periods-to-stability for (a) training and (b) testing data sets.

Table 4: Statistics of the estimated periods-to-stability.

Impact Motion	Minimum	Maximum	Average	Number of Data Set (for T_i >115)
P-1-1	74	128	96	2
P-1-2	45	106	77	0
P-2-1	45	182	99	57
P-3-2	45	120	90	0
P-3-3	45	123	89	0

Table 4 shows the statistics of the estimated T_i for all the impact motion categories. The minimum and maximum T_i values are 45 and 182, and the average values are 96.08, 76.55, 98.55, 89.52 and 89.09 for the P-1-1, P-1-2, P-2-1, P-3-2 and P-3-3, respectively. The P-2-1 motions are characterised with most of the exceptionally high T_i values, hence they are considered to be the most unstable set of data in this work. Based on the minimum and average T_i reported in Table 4, the first 40 periods of data were selected from all the data samples for analysis, thus restricting the study to the use of transient unstable data. With a sampling rate of 300 data points per period, the length of each unstable acceleration signal equal 12,000, and the total number of signals simulated were 3,000. 2,000 of the signals were used for network training, and the remaining 1,000 were used as out-of-sample data for cross-validation testing. Fig. 6 shows the representative time series plot and the phase portrait of each of the impact motion categories in their transient state.

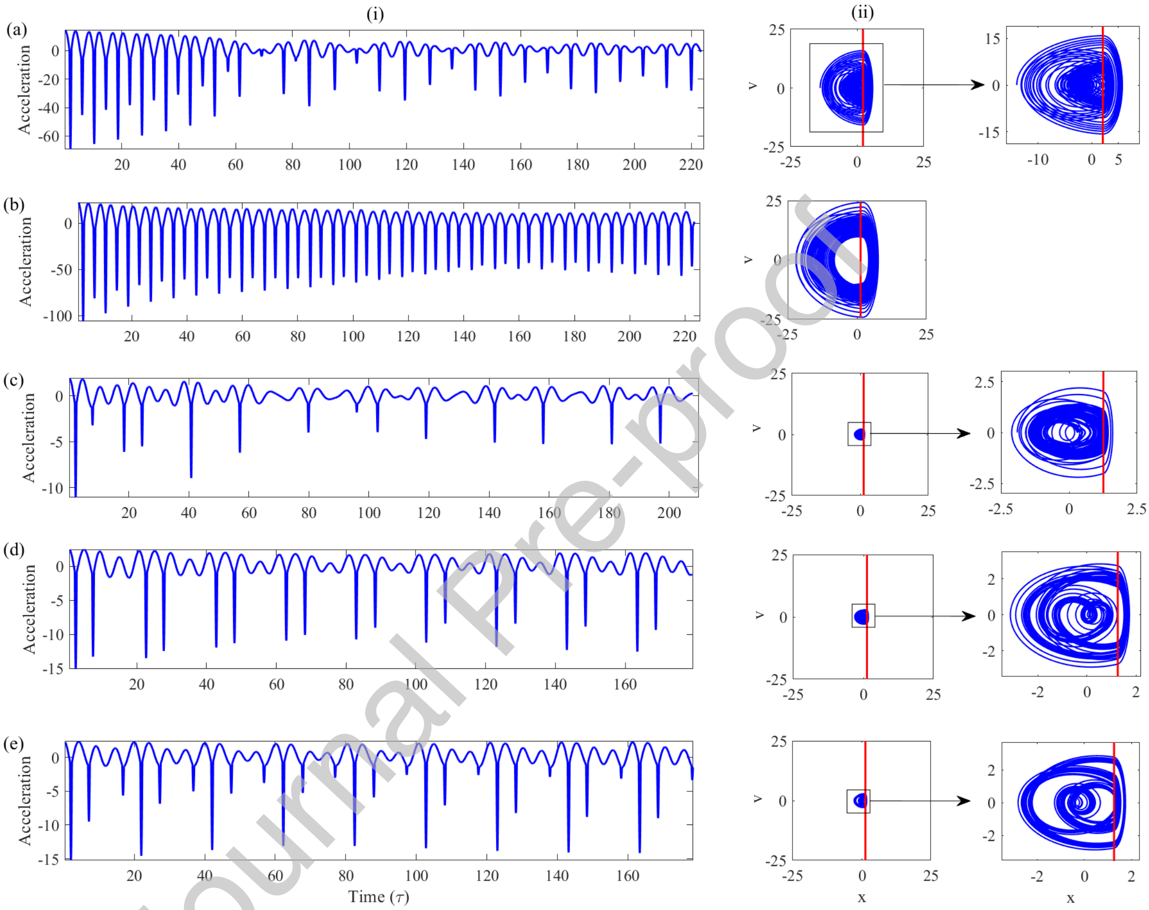


Figure 6: Representative (i) acceleration time histories and (ii) phase portraits for unstable (a) P-1-1, (b) P-1-2, (c) P-2-1, (d) P-3-2 and (e) P-3-3 impact motions. The boundary of impact is denoted by the solid red line.

3.2. Vibro-impact feature extraction

As stated earlier, the waveforms of the acceleration signals were explored to extract morphological and time-interval dependent features. The motivation for this approach is based on the fact that the categories of impact motions are defined based on the numbers of impacts per certain periods of oscillation, and also that impact actions are associated with intermittent peaks in the acceleration waveform due to abrupt velocity drop. To describe the signal in terms of the periodicity and numbers of impacts, some time-interval features were defined while the peaks were analysed for morphological features. Peaks are basic features of most time series events and picking them can be extremely time consuming and labour-intensive. It requires locating all the local minima and maxima in the time series signal in order to define the peak zones. The start (S_{pk}) and end (E_{pk}) of a peak which also marks the loading and unloading of individual impact action was delineated using the second derivative of the acceleration signal. In its second derivative, these two points, S_{pk} and E_{pk}, occur as two neighbouring peaks throughout the signal illustrated in Fig. 7 and can be easily analysed using appropriate peak finding function. The function arguments are fixed to capture only the actual peaks while avoiding background noises and rebounces. The actual peak location (L_{pk}) is calculated as the mid-point between S_{pk} and E_{pk}. The peak amplitude (A_{pk}) is taken as the acceleration data at the actual peak locations (i.e. the L_{pk}th positions) while D_{pk} is calculated as the differences between consecutive L_{pk} data. In all, three types of sequential features annotated as MorFeat1, MorFeat2, and MorFeat3 were extracted from the signal waveform as morphological features. MorFeat1 are row features in which every peak location on the signal waveform (L_{pk}) is represented as '2' while every position marking the end of each period of oscillation is marked as '1', and every other points in the signal is represented as '0's. MorFeat2 and MorFeat3 are similarly row features like MorFeat1, but only that the peak locations (L_{pk}) are represented as peak amplitude (A_{pk}) for MorFeat2, and peak distances apart for MorFeat3. To further simplify the data and remove redundancy, the '0's within the three sets of features were removed thus leaving MorFeat1 as row vectors of '1's and '2's, MorFeat2 as row vectors of '1's and amplitude values (A_{pk}), and MorFeat3 as row vectors of '1's and peak distances apart (D_{pk}). Typical representation of the three wave-form morphological features data for the five categories of impact motions are presented in Fig. 8.

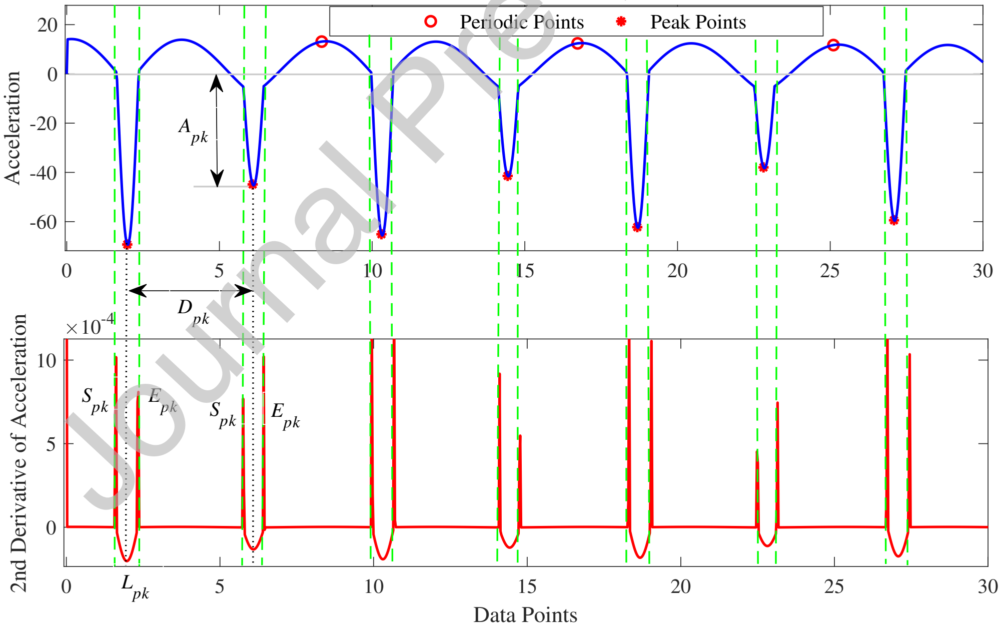


Figure 7: Peak detection and analysis using the second derivation of acceleration signals.

Statistical features which are basically a set of descriptive statistical measures (as shown in Table 1) were calculated for each of the signals in order to generalise about them in time domain. The average value of these measures with respect to each impact motion category is presented in Fig 9, and their order follows P-1-2 > P-1-1 > P-2-1 > P-3-3 > P-3-2. The wide variation of these values as noticed in the figure necessitated them to be normalised before being used as input data for network training. In terms of R and R^2, a moderate correlation level was found to exist between each of the estimated statistical features and the targeted impact motion categories as presented in Fig. 10.

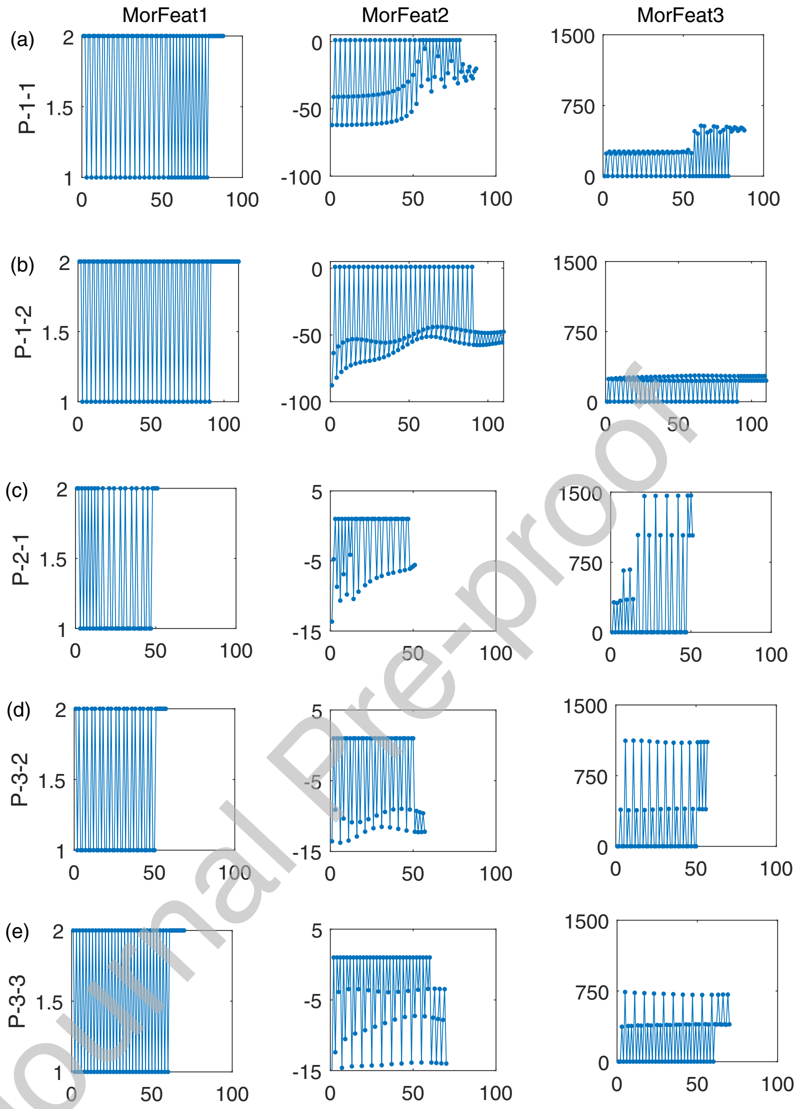


Figure 8: Graphical representation of the wave-form morphological features for the five impact categories.

The CWT time-frequency analysis was carried out using the Morlet mother wavelet and a CWT filter bank which was designed based on the signal length, sampling frequency and a discretised scale at 12 wavelet bandpass filters per octave. The analysis yielded a 128×12,000 matrix of complex conjugates with each row corresponding to one scale, and each column to one time-step. Representative two-dimensional time-frequency plots of the matrices in their absolute value form (scalogram) are shown in Fig. 11(a). The plots show areas of abrupt discontinuities (i.e. impacts) to be characterised with high coefficients (energy level) which are mostly concentrated around scale 30102 representing a frequency range of 0.045-3.62. The procedure of feature selection was implemented by selecting coefficients within these informative frequency range as shown in Fig. 11(b), thus reducing the data to 73×12,000 matrix.

To further reduce the dimension of the CWT data, two approaches were adopted in utilising the extracted CWT coefficients. The first approach was to compute the mean of the coefficients (Mean-

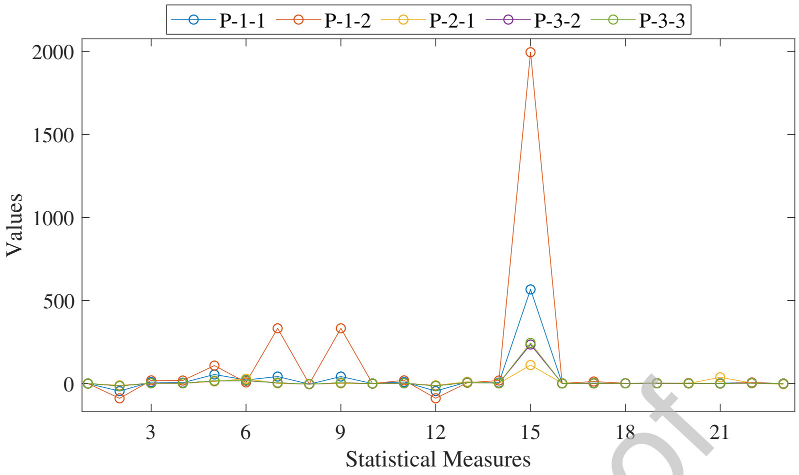


Figure 9: (Colour online) Average variation of the statistical features across the impact motion categories.

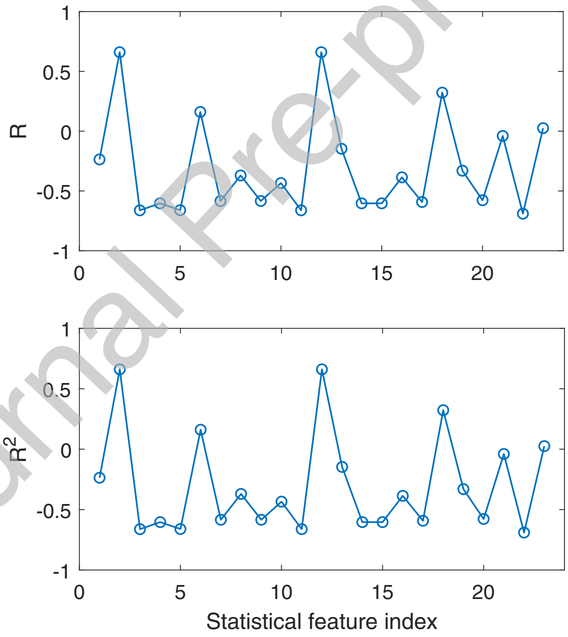


Figure 10: Correlation between each statistical feature and the target impact categories

FreqCWT) along each of the frequency axis for use as input data into the network classifiers. By averaging the CWT coefficients along the frequency axis, the cumulative concentration of impact actions represented as high positive or negative coefficients within a low coefficient background was computed. The second approach was to save the scalograms as images which were later resized and analysed with HOG, CNN and Autoencoders to extract features that were fed into the network classifiers. Direct classification of the images using self feature learning networks, like the deep CNN and LSTM, was also carried out. Fig. 12 are typical plots of the mean CWT coefficients along each of the frequency axes for the different impact motions. P-1-2 with the highest ratio of impacts-to-periods compared to others showed the highest mean CWT values, while P-2-1 with the least ratio of impacts-to-periods showed the smallest values. P-3-2

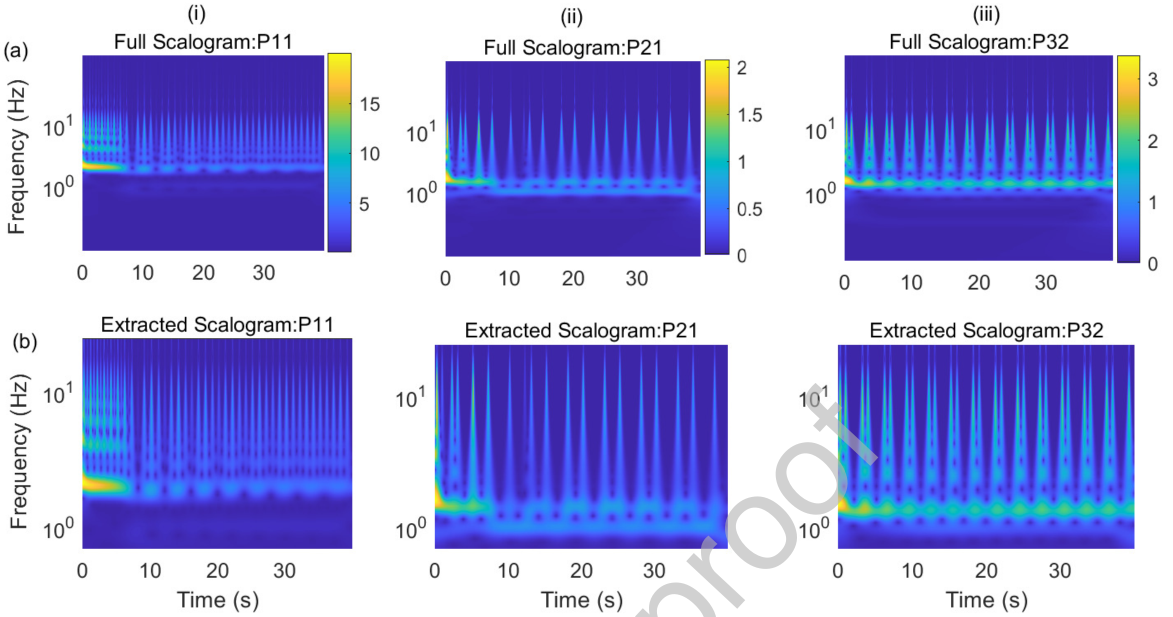


Figure 11: (Colour online) Diagrammatic representation of (a) full and (b) selected portion of (i) P-1-1 (ii) P-2-1 and (iii) P-3-2 scalograms.

and P-3-3 have almost the same number of periods and impacts, hence their close trend. It is worth noting that impact constraints of higher stiffness produce impacts of shorter duration compared to the low stiffness constraints, hence the Mean-FreqCWT values computed for stiff system will be lower than those from soft system even if they are of the same impact category. This can be demonstrated in the Mean-FreqCWT values calculated for P-1-1 in Figs. 12(a) and (b) where $\beta = 18.27$ and 29, respectively.

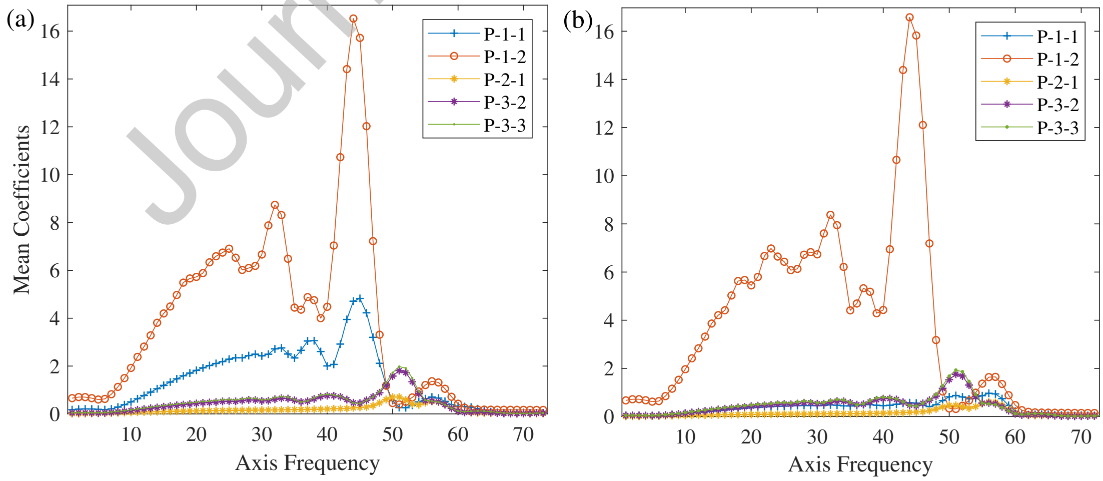


Figure 12: Variation of the mean CWT coefficient along each of the frequency axes for different impact motions with the P-1-1 impact motion originating from (a) soft and (b) stiff impact constraints.

To retain enough information in the resized images, the images for HOG analysis and auto-encoding were resized to 227-by-227 while those for the transfer learning were resized to 224-by-224 as required by the Resnet18 input layer. Prior to the HOG analysis, the true colour CWT images were converted to grayscale images and binarized to remove hue and saturation information while still retaining luminance for efficient feature representation. The binarization helps to set image vectors above a preselected threshold to ‘1’s and others to ‘0’s, thus minimising intraclass variance and permitting easy boundary definition in the form of transition from ‘1’s to ‘0’s or vice versa. Fig. 13 shows the levels of shape information encoded by the HOG descriptor using cell sizes 15 × 15, 30 × 30, 60 × 60 and 120 × 120 produced HOG features of length 86184, 20160, 4212 and 864, respectively, with a 2-by-2 block size and 9 orientation histogram bins. To ensure that sufficient amount of information was encoded about the images without significantly increasing computational cost, the 60-by-60 cell size was selected and utilised for the feature extraction. Fig. 14 shows the HOG feature vectors on the coloured image alongside its concatenated histogram of the cell gradients. To automatically encode the imagery data, two versions of autoencoders were developed. The first encodes the binarized image vectors from their initial 227 × 227 dimension to 1 × 1000, while the second further encodes the 1 × 1000 to 1 × 500 feature space. Each image vectors were presented to the autoencoders as a matrix in which each column represented a data sample. The ability autoencoders to extract discriminative features from raw unprocessed acceleration data was also investigated and this reduced the acceleration data from their initial 1 × 12000 data space through 1 × 1000 to 1 × 500. Typical weights learnt by the autoencoders from individual image vectors at the 1 × 1000 and 1 × 500 feature space are presented in Fig. 15, and the features resulting from auto-encoding the raw data are given in Fig. 16. The auto-encoded image and raw acceleration data are designated as *Autoencoded-GrayImg* and *Autoencoded-RawAcc*, respectively.

Applying an activations function, Fig. 17 shows the feature maps captured by the Resnet18 network at the maximum pooling layer 1 and global pooling layer 5 located in its 2nd and 17th convolutional layers, respectively. The shallower layer is seen to capture fewer information in the image at a higher resolution while the deeper layer tends to capture more detailed but abstract (i.e. higher-level) information about the image. The latter features are most time difficult to interpret as the network abstracts them into a more general concept that it can use for classification purpose. A total of 512 feature vectors resulting from the global pooling layer (pool5) were extracted for each of the vibro-impact image. As earlier mentioned, the HOG, autoencoders and Resnet18 derived features are intended for use as input data into the considered network classifiers.

To self-learn discriminating features from the vibro-impact image representations, self-supervised learning models including the SAE, CAE, LSTM and CNN networks were developed. The two versions of autoencoder developed for feature extraction were stacked up alongside a softmax layer (see Fig. 18) to form the SAE. The images were resized to 128-by-128 for the SAE, CAE and CNN networks but 50-by-50 for the LSTM network to reduce computational costs. Image input layers of equivalent size as the images were used to feed the data into the CAE and CNN, while a sequence input layer of size 50-by-50 was used to feed the images into the image data based LSTM as sequences.

4. Network training results and discussion

Results of the various network classification are presented in Table 5-9. For all the classifiers, their performance on raw and unprocessed data was also investigated while following the conventional approach of using out of sample data which has never been fed into the network for cross-validation.

For the unprocessed data, the LSTM and SVM networks showed appreciable performance ranging between 70.4% for the LSTM network and 78.2% for the SVM compared to the MLP and the stacked autoencoders. This is probably due to the great partitioning power of SVMs and the ability of LSTMs to self-extract temporal and discriminating features from raw time series data. Besides the underperformance

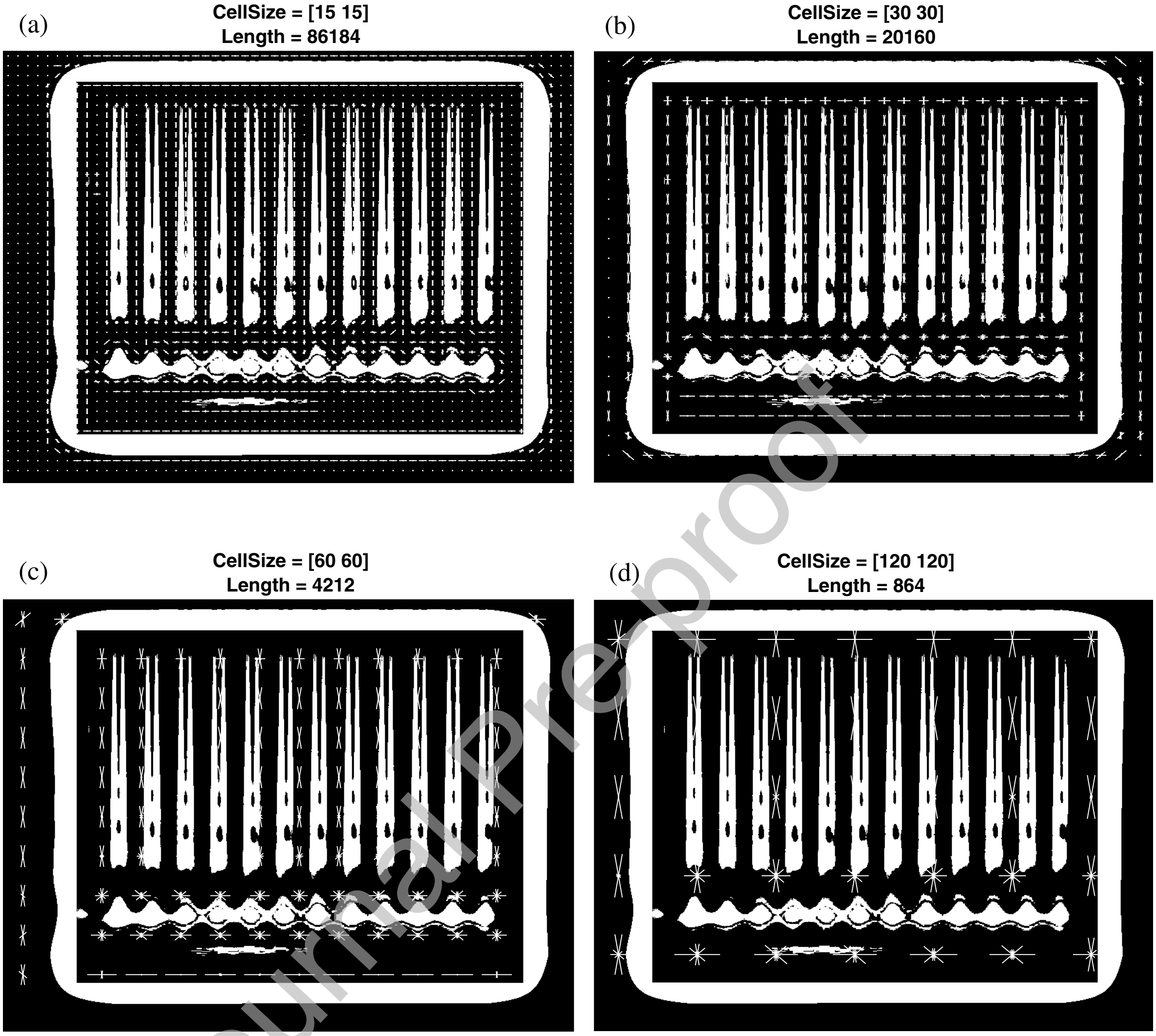


Figure 13: Encoded HOG information for cell sizes (a) 15-by-15, (b) 30-by-30, (c) 60-by-60 and (d) 120-by-120 on binarized grayscale image.

Table 5: LSTM Network performance.

Feature	Training Accuracy (%)	Testing Accuracy (%)	Hidden Units	Time (mins)
Raw Data	89.0	70.4	200	1766
MorFeat1-2-3	99.8	98.6	300	26.87
MorFeat1	89.0	86.9	300	53.95
MorFeat2	100.0	100.0	300	40.58
MorFeat3	99.8	99.8	300	33.92
Statistical	99.9	99.8	80	12.17
HOG	20.0	20.0	300	1495.85
ResNet18	100.0	98.2	80	27.55
Mean-FreqCWT	99.8	99.8	60	9.85
Autoencoded-RawAcc	39.9	39.1	200	69.1
Autoencoded-GrayImg	20.0	20.0	100	86.63

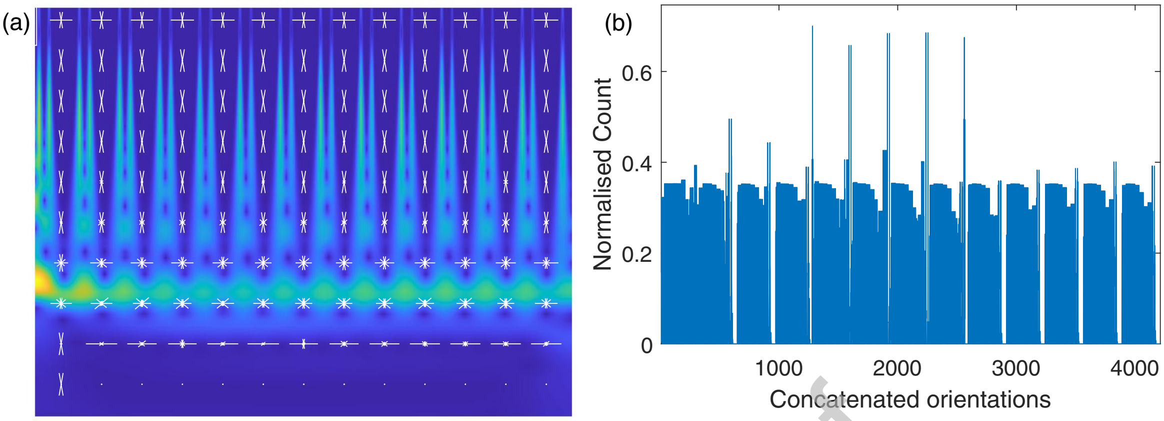


Figure 14: (Colour online) (a) Encoded HOG information on RGB image with cell size 60-by-60 alongside (b) concatenated histogram.

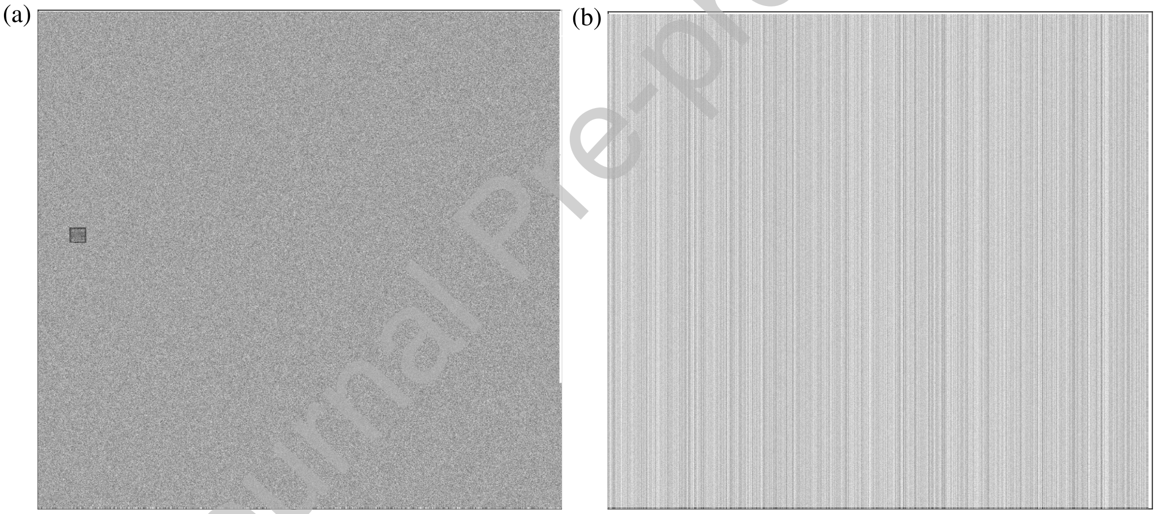


Figure 15: Learnt mappings for (a) 1 × 1000 and (b) 1 × 500 auto-encodings.

Table 6: MLP Network performance, where Tr, Val and Ts represents training, validation and testing data sets, respectively.

Feature	Training (%)			Testing (%)	Hidden Units
	Tr	Val	Ts		
Raw Data	100	100	100	48.6	30
MorFeat1	97	98.3	96.7	84.3	30
MorFeat2	99.9	99.7	99.0	62.0	30
MorFeat3	98.7	97.3	97.3	72.7	30
Statistical	99.6	99.0	99.7	98.7	30
HOG	100	100	100	100	10
ResNet18	100	100	100	99.7	10
Mean-FreqCWT	100	100	100	100	20
Autoencoded-RawAcc	50.9	49.7	48.7	36.0	30
Autoencoded-GrayImg	100	100	99.0	76.0	20

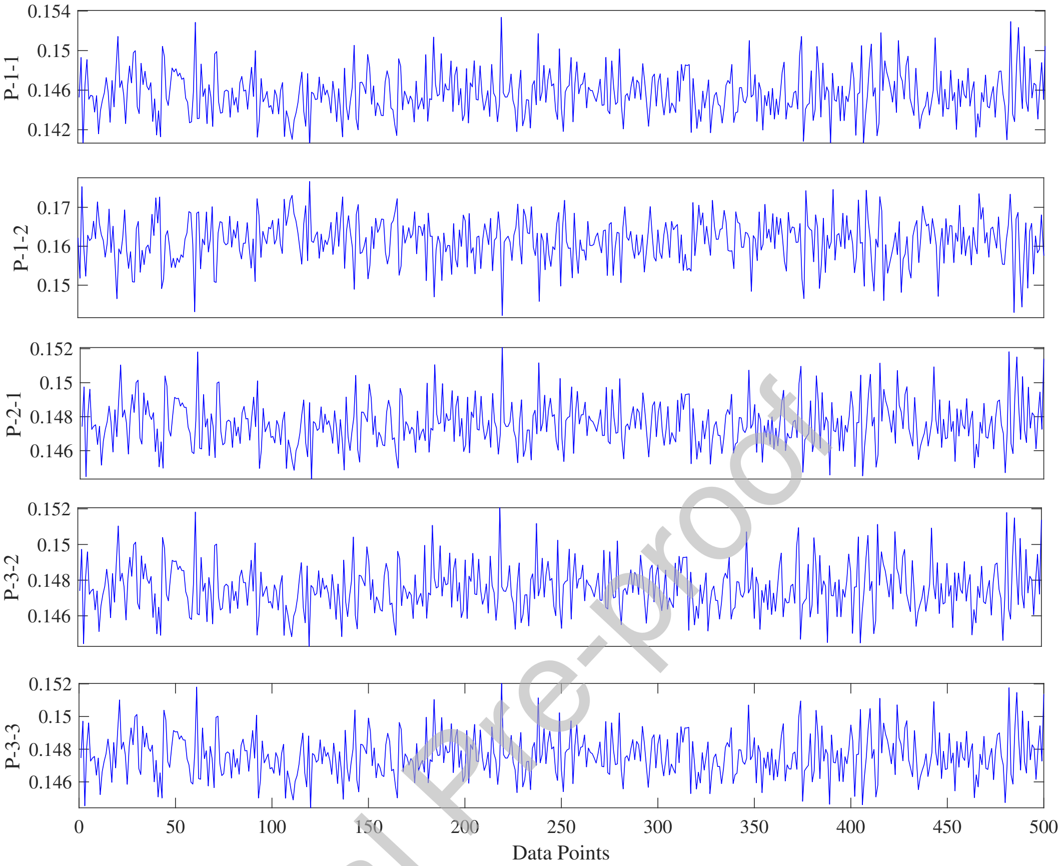


Figure 16: Representative features from auto-encoded raw acceleration data.

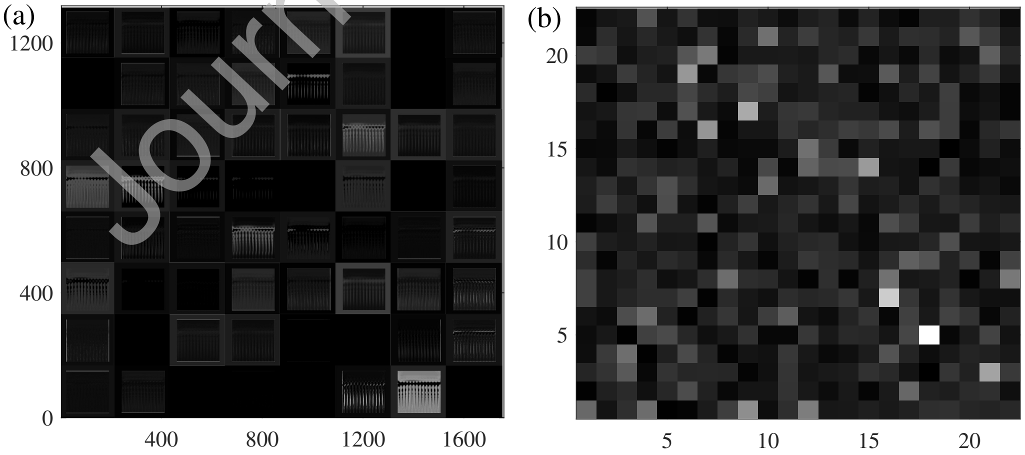


Figure 17: Representative feature maps extracted from (a) shallower maximum pooling layer 1 and (b) the deeper global pooling layer 5 of the Resnet18 network.

of using unprocessed data compared to some processed data methods, the use of LSTM on the unprocessed data set was quite time consuming (see Table 5). Fig. 19 shows the performance of the networks on the out-of-sample raw acceleration data, most of the wrong classifications occurred within the P-2-1 motions

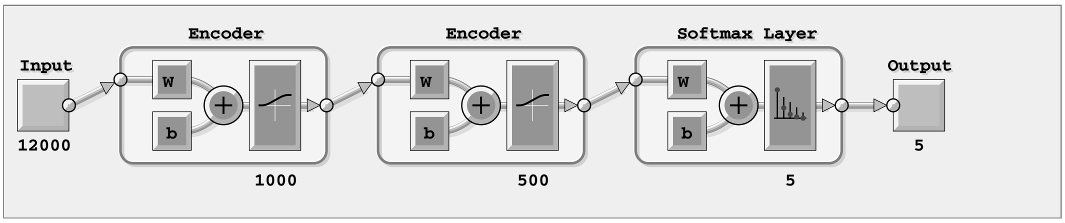


Figure 18: Stacked-autoencoders layers for classification.

Table 7: SVM Network performance.

Feature	Training Accuracy (%)	Testing Accuracy (%)
Raw Data	100	78.2
MorFeat1	97.2	79.4
MorFeat2	100.0	65.3
MorFeat3	100	77.2
Statistical	100	97.1
HOG	100	99.8
ResNet18	100.0	99.7
Mean-FreqCWT	100	100
Autoencoded-RawAcc	47.2	46.1
Autoencoded-GrayImg	68.8	52.1

Table 8: Performances of self-supervised learning models on image data.

Network Model	Training Accuracy (%)	Testing Accuracy (%)
SAE (without fine tuning)	69.9	56.5
SAE (with fine tuning)	100	90.3
CAE	100	100
CNN	100	97.5
LSTM	100	90.3

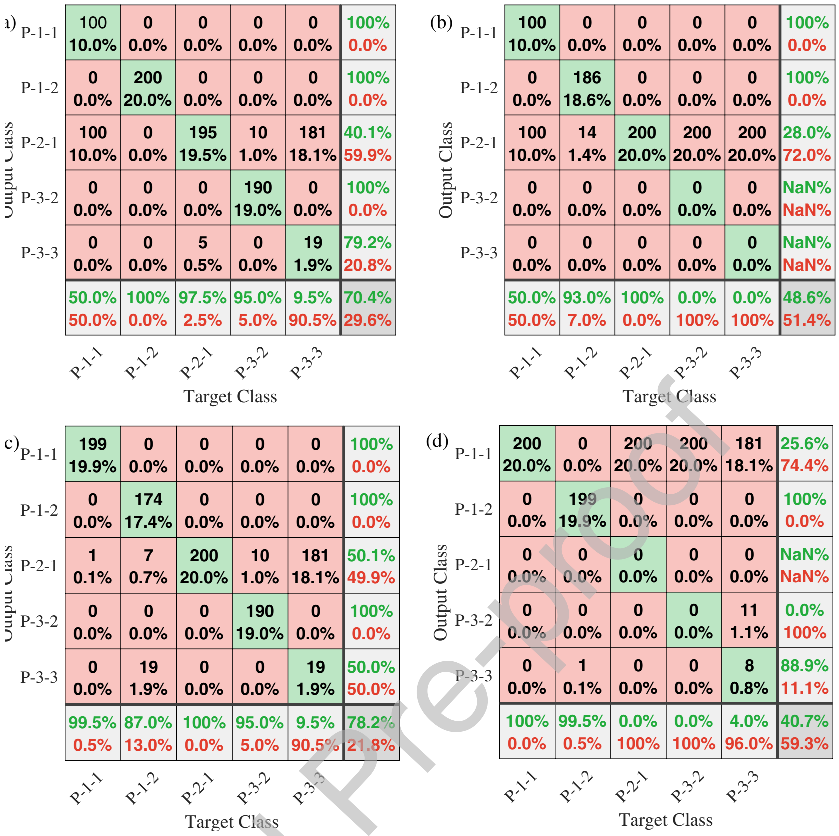
Table 9: Stacked autoencoder classification of raw acceleration data.

Feature	Training Accuracy (%)	Testing Accuracy (%)
Raw Data (without fine tuning)	86.2	39.9
Raw Data (with fine tuning)	94.6	40.7

and were completely unclassifiable for the stacked autoencoders. Having been categorised as the most unstable set of data, the results from the P-2-1 classification further confirms the dependence of network performances on the periods-to-stability of the impact motions as reported by Kenneth *et al.* [10].

The hand-crafted periods-peaks features showed better performance with the LSTM compared to the MLP and SVM as seen in Tables 5-7. This can be related to the fact that (i) the hand-crafted features came out as repeating sequence of data, thus giving LSTM networks which are known for good performances on sequential data an advantage (ii) the hand-crafted features were of different lengths due to the varying numbers of impacts per periods, hence the need to downsize the features into equal sizes before being fed into the MLP and SVM networks could have contributed to their poor performance. Aside accepting data of different sample length, LSTMs can also accept co-current multivariate data as input for a single output.

Extracted statistical features showed accuracy of 99.8%, 98.7% and 97.1% while the frequency-based mean CWT coefficients features (Mean-FreqCWT) showed accuracy of 99.8%, 100% and 100% for the



(a)

Output Class	P-1-1	P-1-2	P-2-1	P-3-2	P-3-3	
P-1-1	100 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
P-1-2	0 0.0%	200 20.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
P-2-1	100 10.0%	0 0.0%	195 19.5%	10 1.0%	181 18.1%	40.1% 59.9%
P-3-2	0 0.0%	0 0.0%	0 0.0%	190 19.0%	0 0.0%	100% 0.0%
P-3-3	0 0.0%	0 0.0%	5 0.5%	0 0.0%	19 1.9%	79.2% 20.8%
	50.0% 50.0%	100% 0.0%	97.5% 2.5%	95.0% 5.0%	9.5% 90.5%	70.4% 29.6%

(b)

Output Class	P-1-1	P-1-2	P-2-1	P-3-2	P-3-3	
P-1-1	100 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
P-1-2	0 0.0%	186 18.6%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
P-2-1	100 10.0%	14 1.4%	200 20.0%	200 20.0%	200 20.0%	28.0% 72.0%
P-3-2	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
P-3-3	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
	50.0% 50.0%	93.0% 7.0%	100% 0.0%	0.0% 100%	0.0% 100%	48.6% 51.4%

(c)

Output Class	P-1-1	P-1-2	P-2-1	P-3-2	P-3-3	
P-1-1	199 19.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
P-1-2	0 0.0%	174 17.4%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
P-2-1	1 0.1%	7 0.7%	200 20.0%	10 1.0%	181 18.1%	50.1% 49.9%
P-3-2	0 0.0%	0 0.0%	0 0.0%	190 19.0%	0 0.0%	100% 0.0%
P-3-3	0 0.0%	19 1.9%	0 0.0%	0 0.0%	19 1.9%	50.0% 50.0%
	99.5% 0.5%	87.0% 13.0%	100% 0.0%	95.0% 5.0%	9.5% 90.5%	78.2% 21.8%

(d)

Output Class	P-1-1	P-1-2	P-2-1	P-3-2	P-3-3	
P-1-1	200 20.0%	0 0.0%	200 20.0%	200 20.0%	181 18.1%	25.6% 74.4%
P-1-2	0 0.0%	199 19.9%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
P-2-1	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
P-3-2	0 0.0%	0 0.0%	0 0.0%	0 0.0%	11 1.1%	0.0% 100%
P-3-3	0 0.0%	1 0.1%	0 0.0%	0 0.0%	8 0.8%	88.9% 11.1%
	100% 0.0%	99.5% 0.5%	0.0% 100%	0.0% 100%	4.0% 96.0%	40.7% 59.3%

Figure 19: Performance of (a) BiLSTM (b) MLP (c) SVM and (d) Stacked autoencoders on the raw out-of-sample acceleration data.

LSTM, MLP and SVM networks, respectively, when tested on the never seen out-of-sample data. Image extracted features including HOG, ResNet18-CNN and Autoencoded-GrayImg features showed classification accuracy of 20%, 98.2% and 20%, respectively, for the LSTM network (see Table 5); 100%, 99.7% and 76.0%, respectively, for the MLP network (see Table 6); and 99.8%, 99.7% and 52.1%, respectively, for the SVM network (see Table 7). Imagery features extracted using the auto-encoders showed the least performance, while those from HOG descriptor and ResNet18 performed consistently well with the MLP and SVM classifiers during training and testing. The LSTM network yielded poor performance on the HOG features. This can be attributed to the fact that unlike the convolutional ResNet18 features, the HOG harvested image features are non-sequential in nature and are spatially disoriented.

The performances of the self-supervised network models are presented in Table 8 and are all seen to perform above 90% accuracy except for the yet to be fine-tuned SAE which later attained an accuracy of 90.3% after fine-tuning. The stacked-autoencoder showed better performance on the processed binary image data (see Table 8) compared to the unprocessed raw acceleration data (see Table 9), thus revealing the importance of data pre-processing. A summary of the training and cross-validation performances of the classifier networks is further presented in Fig. 20, and the confusion matrices resulting from the testing of the self-supervised models are presented in Fig. 21.

Aside the above analysis, the robustness of some of the networks to noise as would be expected in real-life scenario was also investigated. Networks with consistent high performances during training and validation were analysed by infecting their parent signals with noise before extracting their input features.

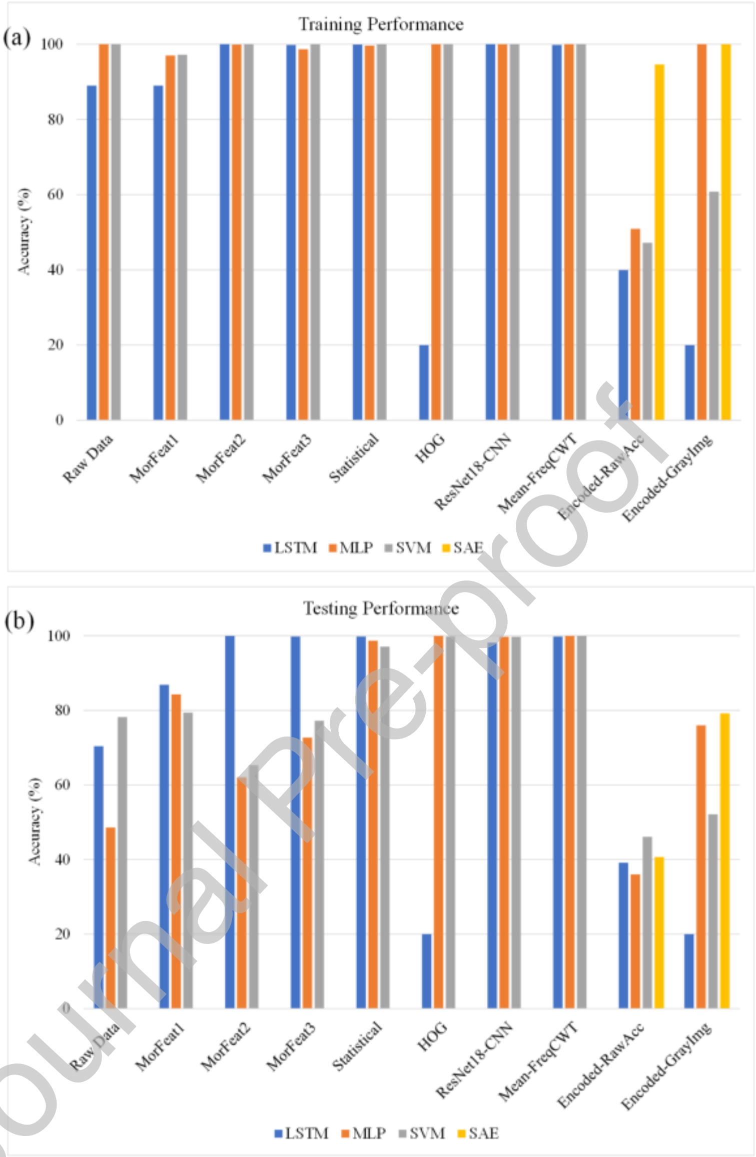
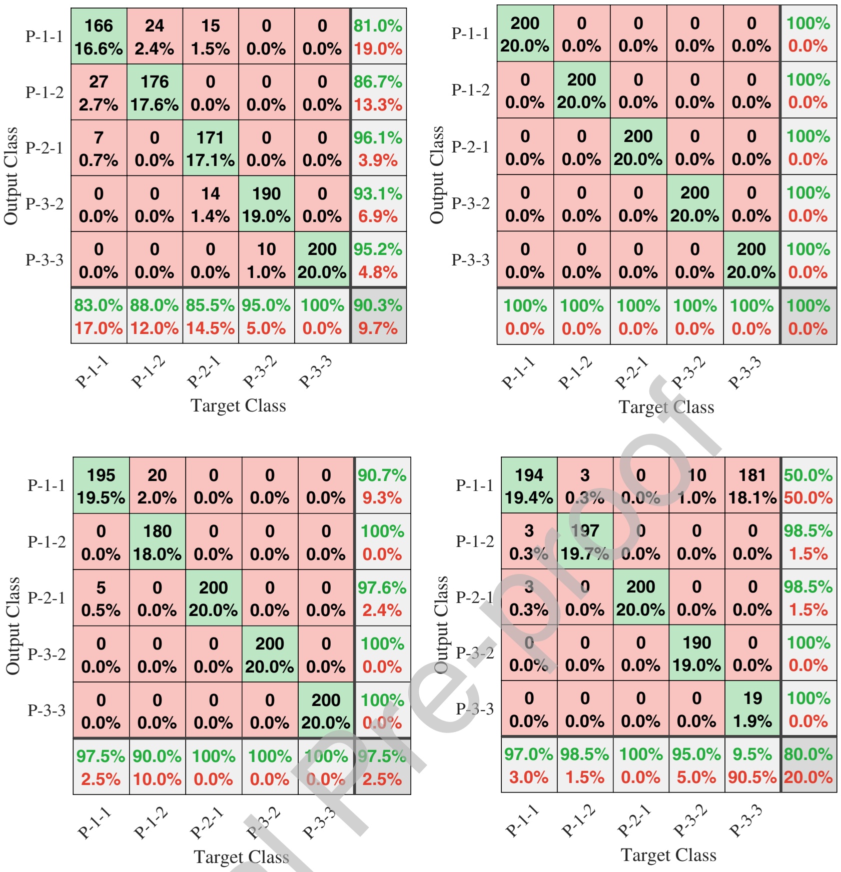


Figure 20: (Colour online) Summary of networks performances during (a) training and (b) testing.

An additive white Gaussian noise (AWGN) with a signal-to-noise ratio (SNR) of 10 and a signal power value measured from the input signal was used in present study. Figure 22 shows the (i) raw signal and (ii) the scalogram plots for a transient P-1-1 impact motion (a) without noise infection and (b) with noise infection. As a pre-processing procedure, the noise infected signals were smoothened using the Savitzky-Golay method with a second degree polynomial over 25 data span. These smoothing parameters were carefully selected to ensure minimal distortion of the original signal. The resulting smoothened signal (in red) shows the morphology of the signal to be intact, hence waveform morphological features are less likely to be affected by the introduced noise. The resulting scalogram plots also showed no visible difference.The performance results of the networks robustness to noise are presented in Tables 10-12 and the results showed no major effect of noise on the analysed networks as the obtained results were very



	P-1-1	P-1-2	P-2-1	P-3-2	P-3-3	
P-1-1	166 16.6%	24 2.4%	15 1.5%	0 0.0%	0 0.0%	81.0% 19.0%
P-1-2	27 2.7%	176 17.6%	0 0.0%	0 0.0%	0 0.0%	86.7% 13.3%
P-2-1	7 0.7%	0 0.0%	171 17.1%	0 0.0%	0 0.0%	96.1% 3.9%
P-3-2	0 0.0%	0 0.0%	14 1.4%	190 19.0%	0 0.0%	93.1% 6.9%
P-3-3	0 0.0%	0 0.0%	0 0.0%	10 1.0%	200 20.0%	95.2% 4.8%
	83.0% 17.0%	88.0% 12.0%	85.5% 14.5%	95.0% 5.0%	100% 0.0%	90.3% 9.7%

	P-1-1	P-1-2	P-2-1	P-3-2	P-3-3	
P-1-1	200 20.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
P-1-2	0 0.0%	200 20.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
P-2-1	0 0.0%	0 0.0%	200 20.0%	0 0.0%	0 0.0%	100% 0.0%
P-3-2	0 0.0%	0 0.0%	0 0.0%	200 20.0%	0 0.0%	100% 0.0%
P-3-3	0 0.0%	0 0.0%	0 0.0%	0 0.0%	200 20.0%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%

	P-1-1	P-1-2	P-2-1	P-3-2	P-3-3	
P-1-1	195 19.5%	20 2.0%	0 0.0%	0 0.0%	0 0.0%	90.7% 9.3%
P-1-2	0 0.0%	180 18.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
P-2-1	5 0.5%	0 0.0%	200 20.0%	0 0.0%	0 0.0%	97.6% 2.4%
P-3-2	0 0.0%	0 0.0%	0 0.0%	200 20.0%	0 0.0%	100% 0.0%
P-3-3	0 0.0%	0 0.0%	0 0.0%	0 0.0%	200 20.0%	100% 0.0%
	97.5% 2.5%	90.0% 10.0%	100% 0.0%	100% 0.0%	100% 0.0%	97.5% 2.5%

	P-1-1	P-1-2	P-2-1	P-3-2	P-3-3	
P-1-1	194 19.4%	3 0.3%	0 0.0%	10 1.0%	181 18.1%	50.0% 50.0%
P-1-2	3 0.3%	197 19.7%	0 0.0%	0 0.0%	0 0.0%	98.5% 1.5%
P-2-1	3 0.3%	0 0.0%	200 20.0%	0 0.0%	0 0.0%	98.5% 1.5%
P-3-2	0 0.0%	0 0.0%	0 0.0%	190 19.0%	0 0.0%	100% 0.0%
P-3-3	0 0.0%	0 0.0%	0 0.0%	0 0.0%	19 1.9%	100% 0.0%
	97.0% 3.0%	98.5% 1.5%	100% 0.0%	95.0% 5.0%	9.5% 90.5%	80.0% 20.0%

Figure 21: Confusion matrices originating from the cross-validation of the image data based (a) SAE (b) CAE (c) CNN and LSTM classifiers.

similar to those from the non-noise infected signals. However, it should be noted that the robustness to noise has been carried out with an assumption that the same category and level of noise is applicable to all data sets. This may not often be the case for real life scenarios as there may be multiple sources and level of noises. It can also be said that the simulated impacts as well as their defining peaks were prominent enough not to be affected by the signal distortion (i.e. reduced peak height and increased peak width) arising from the smoothing and also by the remaining low -frequency noises in the signal. Cases when the signal is dominated with weak impacts may require further data analysis.

Table 10: LSTM Network performance on noise infected data.

Feature	Training Accuracy (%)	Testing Accuracy (%)	Hidden Units	Time (mins)
Statistical^{nsy}	99.9	99.9	80	12.17
HOG^{nsy}	20.0	20.0	80	494.4
ResNet18^{nsy}	100.0	96.9	80	27.55
Mean-FreqCWT^{nsy}	99.6	99.9	80	9.85

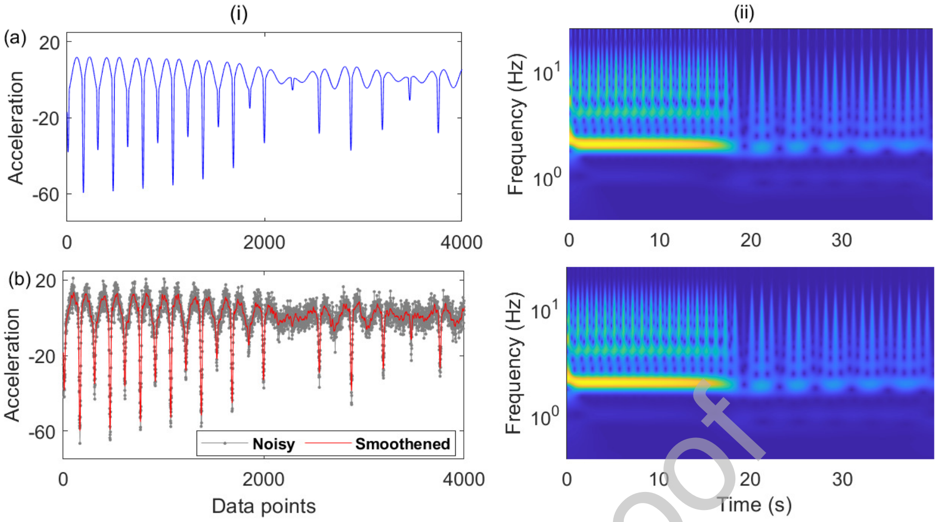


Figure 22: (Colour online) (i)Raw signal and (ii) scalogram plots for a transient P-1-1 impact motion (a) without noise and (b) with noise.

Table 11: MLP Network performance on noise infected data, where Tr, Val and Ts represents training, validation and testing data sets, respectively.

Feature	Training (%)			Testing (%)	Hidden Units
	Tr	Val	Ts		
Statistical^{nsy}	99.6	100	99.0	99.4	10
HOG^{nsy}	100	100	100	99.3	10
ResNet18^{nsy}	100	100	100	99.8	10
Mean-FreqCWT^{nsy}	99.9	100	100	99.1	10

Table 12: SVM Network performance on noise infected data.

Feature	Training Accuracy (%)	Testing Accuracy (%)
Statistical^{nsy}	100	97.2
HOG^{nsy}	100	99.3
ResNet18^{nsy}	100.0	100
Mean-FreqCWT^{nsy}	100	99.9

Table 13: Performances of self-supervised learning models on image data from noise infected data.

Network Model	Training Accuracy (%)	Testing Accuracy (%)
CAE	100	99.2
CNN	100	94.0

5. Conclusions

In this present study, both supervised and unsupervised feature extractions have been carried out on acceleration signals of five classes of vibro-impact motions as a means of improving their classification. For the supervised case, a combination of morphological time-interval features and statistical features were extracted. The morphological time-interval features describe the basic shape and position (timing) of the different waveforms occurring along the signal. Amplitudes and widths describe the shape of the waveforms while peak locations and peak separation describe the position of the waveforms. Present study shows that these features are of relevant importance when carrying out a network based classification of the signals. The classification accuracy of using the Morfeat1, Morfeat2 and Morfeat3 features with Long-Short Term Memory (LSTM) network ranged between $86.9\% - 100\%$ but between $62.0\% - 84.3\%$ with the multi-layer perceptron (MLP) and support vector machine (SVM) networks. The statistical features yielded an accuracy of $99.8\%, 98.7\%$ and 97.1% for the LSTM, MLP and SVM, respectively.

In the case of unsupervised feature extraction, the raw vibration signals were first pre-processed into a two-dimensional image representation from which features were selected and automatically extracted for dimensionality reduction and data simplification. Extracted features were fed into the network classifiers to make a prediction of the class they belong to. The ResNet18 features and the mean of the coefficients along each of the frequency axis (Mean-FreqCWT) features performed excellently well with the three network classifiers (LSTM, MLP and SVM) with accuracies that range between 98.2% and 100%. Due to their unpreserved spatial orientation, the histogram of oriented gradients (HOG) features did not do well with LSTM but did well with the MLP and SVM classifiers. Auto-encoded features from both the raw acceleration data and imagery data showed lower performances with the network classifiers. The self-supervised feature learning networks including the fine-tuned stacked autoencoders (SAE), the convolutional autoencoders (CAE), the convolutional neural networks (CNN) and the LSTM respectively showed accuracy of $90.3\%, 100\%, 97.5\%$ and 90.3% on the image vector data. The high performing models listed above showed consistent high performance during their training and testing, hence they are well-suited for an online optimisation system. Compared to others, the statistical, ResNet18 and the Mean-FreqCWT features are less laborious and less computationally intensive. With their performance being or nearing 100% for all the networks, they become the first point of consideration when designing the impact motion categorisation model. This study has also showed that extracting appropriate features and selecting the right network classifier is essential for vibro-impact motions characterisation as some of the networks showed better performances on certain features compared to other features and compared to their raw data. The auotoencoder features as automatically extracted features, despite being computationally intensive showed lesser performance in characterising the impact motions compared to others like the HOG and the pre-trained Resnet18 features. This may be because important lower level features were discarded by the autoencoders during feature learning.

In conclusion, the use of CAE, CNN and the extraction of statistical, HOG, Mean-FreqCWT and pre-trained Resnet18 features proved to be less laborious and less computationally intensive. With their accuracies reaching 100% or nearly, they become the first point of consideration for designing a vibro-impact motion classification model. Also, the compliance of their noise infected version (Tables 10-13) with the uninfected version Tables 5-8 suggests their robustness for real life applications. It should be noted that the robustness to noise has been carried out with an assumption that the same category and level of noise is applicable to all data sets. This may not often be the case for real life scenarios as there may be multiple sources and level of noises. It can also be said that the simulated impacts as well as their defining peaks were prominent enough not to be affected by the signal distortion (i.e. reduced peak height and increased peak width) arising from the smoothing and also by the remaining low-frequency noises in the signal. Cases when the signal is dominated with weak impacts may require further data analysis, which could be a future work. Future works will also involve using exceptional network models concluded from this study on experimental data from the physical impact oscillator for the purpose of impact motion characterisation and system optimisation. Impact motions categories can be used alongside other extractable features from the impact data to predict the secondary stiffness (k_2 in Fig. 1) representing the stiffness of the rock.

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] M. A. Davies and B. Balachandran, “Impact dynamics in milling of thin-walled structures,” *Nonlinear Dyn.*, vol. 22, pp. 375–392, 2000.

[2] C. M. Liao, H. Vlajic, N. Karki, and B. Balachandran, “Parametric studies on drill-string motions,” *Int. J. Mech. Sci.*, vol. 54, pp. 260–268, 2012. doi: https://doi.org/10.1016/j.ijmecsci.2011.11.005.

[3] Y. Khulief, F. Al-Sulaiman, and S. Bashmal, “Vibration analysis of drillstrings with self-excited stick–slip oscillations,” *J. Sound Vib.*, vol. 299, no. 3, pp. 540–558, 2007. doi:https://doi.org/10.1016/j.jsv.2006.06.065.

[4] X. Zheng, V. Agarwal, X. Liu, and B. Balachandran, “Nonlinear instabilities and control of drill-string stick-slip vibrations with consideration of state-dependent delay,” *J. Sound Vib.*, vol. 473, p. 115235, 2020. doi:https://doi.org/10.1016/j.jsv.2020.115235.

[5] H. Zheng, Z. Li, and X. Chen, “Gear fault diagnosis based on continuous wavelet transform,” *Mech. Syst. Signal Process.*, vol. 16, no. 2-3, pp. 447–457, 2002. doi:https://doi.org/10.1006/mssp.2002.1482.

[6] Y. Yan, J. Xu, M. Wiercigroch, and Q. Guo, “Statistical basin of attraction in time-delayed cutting dynamics: Modelling and computation,” *Physica D*, vol. 416, p. 132779, 2021. doi:https://doi.org/10.1016/j.physd.2020.132779.

[7] B. Balachandran, “Dynamics of an elastic structure excited by harmonic and aharmonic impactor motions,” *J. Vib. Control*, vol. 9, pp. 265–279, 2003. doi:https://doi.org/10.1177/107754603030752.

[8] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, “Deep learning and its applications to machine health monitoring,” *Mech. Syst. Signal Process.*, vol. 115, pp. 213–237, 2019. doi: https://doi.org/10.1016/j.ymssp.2018.05.050.

[9] R. Yan and R. X. Gao, “Hilbert–Huang transform-based vibration signal analysis for machine health monitoring,” *IEEE Trans. Instrum. Meas.*, vol. 55, no. 6, pp. 2320–2329, 2006. doi:https://doi.org/10.1109/TIM.2006.887042.

[10] 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 Netw.*, vol. 140, pp. 49–64, 2021. doi:https://doi.org/10.1016/j.neunet.2021.02.027.

[11] M. Wiercigroch, “Resonance enhanced drilling: method and apparatus,” Jan. 15 2013. US Patent 8,353,368.

[12] E. Pavlovskaia, D. C. Hendry, and M. Wiercigroch, “Modelling of high frequency vibro-impact drilling,” *Int. J. Mech. Sci.*, vol. 91, pp. 110–119, 2015. doi:https://doi.org/10.1016/j.ijmecsci.2013.08.009.

[13] E. Pavlovskaia, M. Wiercigroch, and C. Grebogi, “Modeling of an impact system with a drift,” *Phys. Rev. E.*, vol. 64, no. 5, p. 056224, 2001. doi:https://doi.org/10.1103/PhysRevE.64.056224.

[14] 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. doi:https://doi.org/10.1016/j.chaos.2003.12.064.

[15] M. Liao, J. Ing, J. Páez Chávez, and M. Wiercigroch, “Bifurcation techniques for stiffness identification of an impact oscillator,” *Commun. Nonlinear Sci. Numer. Simul.*, vol. 41, pp. 19–31, 2016. doi:https://doi.org/10.1016/j.cnsns.2015.11.027.

[16] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques.* Elsevier, 2011. doi:https://doi.org/10.1016/C2009-0-61819-5.

[17] E. Alpaydin, *Introduction to Machine Learning.* MIT Press, 2020.

[18] J. Tang, S. Alelyani, and H. Liu, “Feature selection for classification: a review,” *Data classification: Algorithms and Applications*, p. 37, 2014. doi:https://doi.org/10.1201/b17320.

[19] H. Packard, “The fundamentals of signal analysis,” *HP Application Note*, vol. 243, 1985.

[20] V. Srinivasan, C. Eswaran, Sriraam, and N, “Artificial neural network based epileptic detection using time-domain and frequency-domain features,” *J. Med. Syst.*, vol. 29, no. 6, pp. 647–660, 2005, doi:https://doi.org/10.1007/s10916-005-6133-1.

[21] Z. Pápai and T. Pap, “Analysis of peak asymmetry in chromatography,” *J. Chromatogr. A*, vol. 953, no. 1-2, pp. 31–38, 2002. doi:https://doi.org/10.1016/S0021-9673(02)00121-8.

[22] C. Yang, Z. He, and W. Yu, “Comparison of public peak detection algorithms for maldi mass spectrometry data analysis,” *BMC Bioinform.*, vol. 10, no. 1, p. 4, 2009. doi:https://doi.org/10.1186/1471-2105-10-4.

[23] L. Maršánová, M. Ronzhina, R. Smíšek, M. Vítek, A. Němcová, L. Smital, and M. Nováková, “Ecg features and methods for automatic classification of ventricular premature and ischemic heartbeats: A comprehensive experimental study,” *Sci. Rep.*, vol. 7, no. 1, pp. 1–11, 2017. doi:https://doi.org/10.1038/s41598-017-10942-6.

[24] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, “A review of classification algorithms for EEG-based brain–computer interfaces,” *J. Neural Eng.*, vol. 4, no. 2, p. R1, 2007. doi:https://doi.org/10.1088/1741-2552/aab2f2.

[25] R. H. Parkinson and J. R. Gray, "Neural conduction, visual motion detection, and insect flight behaviour are disrupted by low doses of imidacloprid and its metabolites," *Neurotoxicology*, vol. 72, pp. 107–113, 2019. doi:https://doi.org/10.1016/j.neuro.2019.02.012.

[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. doi:https://doi.org/10.1109/IJCNN.2019.8852096.

[27] 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. Trans. Biomed. Eng.*, vol. 53, no. 12, pp. 2507–2515, 2006. doi:https://doi.org/10.1109/TBME.2006.880879.

[28] 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. doi:https://doi.org/10.1109/ICASSP.1998.679665.

[29] 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.doi: https://doi.org/10.1109/I2MTC.2012.6229708.

[30] B. Esmael, A. Arnaout, R. K. Fruhwirth, and G. Thonhauser, "A statistical feature-based approach for operations recognition in drilling time series," *Int. J. Comput. Inf. Sys & Ind. Manag. Appl.*, vol. 5, pp. 454–61, 2015.

[31] I. M. Pires, G. Marques, N. M. Garcia, F. Flórez-Revuelta, M. Canavarro Teixeira, E. Zdravevski, S. Spinsante, and M. Coimbra, "Pattern recognition techniques for the identification of activities of daily living using a mobile device accelerometer," *Electronics*, vol. 9, no. 3, p. 509, 2020. doi:https://doi.org/10.7287/peerj.preprints.27225v2.

[32] B. Schowe, "Feature selection for high-dimensional data with rapidminer," in *Proceedings of the 2nd RapidMiner Community Meeting And Conference (RCOMM 2011), Aachen*, 2011. doi:http://dx.doi.org/10.17877/DE290R-14289.

[33] Y. N. Jeng and Y.-C. Cheng, "A short time fourier transform for time series data," *J. Aeronaut. Astronaut. Aviat. Ser.*, vol. 39, no. 2, pp. 117–127, 2007. doi:http://dx.doi.org/10.6125/JoAAA.200706_39(2).05.

[34] A. T. Poyil, K. Nasimudeen, and S. Aljahdali, "Significance of cohen's class for time frequency analysis of signals," *Int. J. Comput. Appl.*, vol. 72, no. 12, 2013. doi:https://doi.org/10.5120/12543-8960.

[35] W. J. Staszewski, K. Worden, and G. R. Tomlinson, "Time–frequency analysis in gearbox fault detection using the wigner–ville distribution and pattern recognition," *Mech. Syst. Signal Process.*, vol. 11, no. 5, pp. 673–692, 1997. doi:https://doi.org/10.1006/mssp.1997.0102.

[36] H.-I. Choi and W. J. Williams, "Improved time-frequency representation of multicomponent signals using exponential kernels," *IEEE Trans. Signal Process.*, vol. 37, no. 6, pp. 862–871, 1989. doi:https://doi.org/10.1109/ASSP.1989.28057.

[37] D. P. Mandic, N. ur Rehman, Z. Wu, and N. E. Huang, "Empirical mode decomposition-based time-frequency analysis of multivariate signals: The power of adaptive data analysis," *IEEE Signal Process. Mag.*, vol. 30, no. 6, pp. 74–86, 2013. doi:https://doi.org/10.1109/MSP.2013.2267931.

[38] L. Cohen, *Time-frequency analysis*, vol. 778. Prentice hall, 1995.

[39] O. Faust, U. R. Acharya, H. Adeli, and A. Adeli, "Wavelet-based EEG processing for computer-aided seizure detection and epilepsy diagnosis," *Seizure*, vol. 26, pp. 56–64, 2015. doi:https://doi.org/10.1016/j.seizure.2015.01.012.

[40] Y. Mallet, D. Coomans, J. Kautsky, and O. De Vel, "Classification using adaptive wavelets for feature extraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 10, pp. 1058–1066, 1997.doi: https://doi.org/10.1109/34.625106.

[41] W. Zhou, S. Gao, L. Zhang, and X. Lou, "Histogram of oriented gradients feature extraction from raw bayer pattern images," *IEEE Trans. Circuits Syst. II Express Briefs*, 2020. doi:https://doi.org/10.1109/TCSII.2020.2980557.

[42] J. Ma, F. Wu, J. Zhu, D. Xu, and D. Kong, "A pre-trained convolutional neural network based method for thyroid nodule diagnosis," *Ultrasonics*, vol. 73, pp. 221–230, 2017. doi:https://doi.org/10.1016/j.ultras.2016.09.011.

[43] K. Tao, Y. He, and C. Chen, "Design of face recognition system based on convolutional neural network," in *2019 Chinese Automation Congress (CAC)*, pp. 5403–5406, IEEE, 2019. doi:https://doi.org/10.48084/etasr.3490.

[44] M. H. Beale, M. T. Hagan, and H. B. Demuth, *Deep Learning Toolbox User's Guide.* Mathworks, 2020.

[45] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015. doi:https://doi.org/10.1007/s11263-015-0816-y.

[46] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986. doi:https://doi.org/10.1038/323533a0.

[47] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural Netw.*, vol. 2, no. 1, pp. 53–58, 1989. doi:https://doi.org/10.1016/0893-6080(89)90014-2.

[48] 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. doi:https://doi.org/10.1007/978-3-319-10605-2_1.

[49] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *International Conference on Artificial Neural Networks*, pp. 52–59, Springer, 2011. doi:https://doi.org/10.1007/978-3-642-21735-7_7.

[50] S. Savalia and V. Emamian, "Cardiac arrhythmia classification by multi-layer perceptron and convolution neural networks," *Bioengineering*, vol. 5, no. 2, p. 35, 2018. doi:https://dx.doi.org/10.3390/bioengineering5020035.

[51] M. Darbor, L. Faramarzi, and M. Sharifzadeh, "Performance assessment of rotary drilling using non-linear multiple regression analysis and multilayer perceptron neural network," *Bull. Eng. Geol. Environ.*, vol. 78, no. 3, pp. 1501–1513, 2019. doi:https://doi.org/10.1007/s10064-017-1192-3.

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

[53] A. Rocha and S. K. Goldenstein, "Multiclass from binary: Expanding one-versus-all, one-versus-one and ecoc-based approaches," *IEEE Trans. Neural Netw. Learn Syst.*, vol. 25, no. 2, pp. 289–302, 2013.doi:https://doi.org/10.1109/TNNLS.2013.2274735.

[54] F. Al-Shargie, T. B. Tang, N. Badruddin, and M. Kiguchi, "Towards multilevel mental stress assessment using svm with ecoc: an EEG approach," *Med. Biol. Eng. Comput.*, vol. 56, no. 1, pp. 125–136, 2018. doi:https://doi.org/10.1007/s11517-017-1733-8.

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

[56] E. L. Allwein, R. E. Schapire, and Y. Singer, "Reducing multiclass to binary: A unifying approach for margin classifiers," *J. Mach. Learn Res.*, vol. 1, no. Dec, pp. 113–141, 2000. doi:https://doi.org/10.1162/15324430152733133.

[57] Y.-C. Wu, Y.-S. Lee, and J.-C. Yang, "Robust and efficient multiclass svm models for phrase pattern recognition," *Pattern Recognit.*, vol. 41, no. 9, pp. 2874–2889, 2008. doi:https://doi.org/10.1016/j.patcog.2008.02.010.

[58] T. Kudo and Y. Matsumoto, "Chunking with support vector machines," in *Second Meeting of the North American Chapter of the Association for Computational Linguistics*, 2001, doi:https://doi.org/10.3115/1073336.1073361.

[59] Y.-C. Wu, T.-K. Fan, Y.-S. Lee, and S.-J. Yen, "Extracting named entities using support vector machines," in *International workshop on knowledge discovery in life science literature*, pp. 91–103, Springer, 2006. doi:https://doi.org/10.1007/11683568_8.

[60] J. Wu, P. Guo, Y. Cheng, H. Zhu, X.-B. Wang, and X. Shao, "Ensemble generalized multiclass support-vector-machine-based health evaluation of complex degradation systems," *IEEE ASME Trans. Mechatron*, vol. 25, no. 5, pp. 2230–2240, 2020. doi:https://doi.org/10.1109/TMECH.2020.3009449.

[61] Y. Guo, J. Meng, Y. Li, S. Ge, J. Xing, and H. Wu, "Incremental small sphere and large margin for online recognition of communication jamming," *Appl. Intell.*, vol. 50, pp. 3429–3440, 2020. doi:https://doi.org/10.1007/s10489-020-01717-0.

[62] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997. doi:https://doi.org/10.1162/neco.1997.9.8.1735.

Credit Author Statement

Kenneth Afebu: Methodology, Software, Validation, Formal analysis, Investigation, Writing - Original Draft, Visualization, Funding acquisition;

Yang Liu: Conceptualization, Methodology, Writing - Review & Editing, Visualization, Supervision, Project administration;

Evangelos Papatheou: Conceptualization, Methodology, Writing - Review & Editing, Supervision.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: