LSTM-based approach for predicting periodic motions of an impacting system via transient dynamics

Kenneth Omokhagbo Afebu, Yang Liu*, Evangelos Papatheou, Bingyong Guo

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

Abstract

Dynamically impacting systems are characterised with inherent instability and complex non-linear phenomena which makes it practically difficult to predict the steady state response of the system at transient periods. This study investigates the ability of a data driven machine learning method using Long Short-Term Memory networks to learn the complex nonlinearity associated with co-existing impact responses from limited transient data. A one-degree-of-freedom impact oscillator has been used to represent the bit-rock interaction for percussive drilling. Simulated data results show velocity measurements to contribute most to predicting steady state responses from transient dynamics with most of the network models reaching an accuracy of over 95%. Limitations to practically measurable variables in dynamic systems warranted the development of a feature based network model for impact motion classification. Experimental data from a two-degrees-of-freedom impacting system representing percussive bit penetration has been used to demonstrate the effectiveness of this method. The study thus provides a precise and less computational means of detecting and avoiding underperforming impact modes in percussive drilling.

Keywords: Vibro-impact; Coexisting attractor; Long Short-Term Memory network; Basin prediction; Percussive drilling.

List of Abbreviations

ROP	Rate of Penetration
RED	Resonance Enhanced Drilling
LSTM	Long Short-Term Memory Network
RNN	Recurrent Neural Network
P-1-1	Period-one motion with one-impact
P-1-2	Period-one motion with two-impacts
BSN1	Basin 1
BSN2	Basin 2
BSN3	Basin 3

*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), `b.guo@exeter.ac.uk` (Bingyong Guo)

BSN4	Basin 4
BSN5	Basin 5
BiLSTM	Bi-directional Long Short-Term Memory Network
adam	Adaptive Moment Estimation
sgdm	Stochastic Gradient Descent with Momentum
BiLSTM-x	Displacement measurements based network
BiLSTM-v	Velocity measurements based network
BiLSTM-a	Acceleration measurements based network
BiLSTM-xv	Displacement and velocity measurements based network
BiLSTM-ap	Acceleration measurements and periodic boundary based network
BiLSTM-xvep	Displacement measurements, velocity measurements, impact boundary and periodic boundary based network
BiLSTM-axep	Acceleration measurements, displacement measurements, impact boundary and periodic boundary based network
BiLSTM-nrm	Normalised displacement and velocity measurements based network
BiLSTM-nsy	Noise effected displacement and velocity measurements based network
BiLSTM-a_{ft}	Acceleration features based network
OVA	Overall Accuracy
GUI	Graphic User Interface
LVDT	Linear Variable Differential Transformer
ACC	Accelerometer
CMD	Command
PWD	Pulse Width Modulation

1. Introduction

In attempts to optimise well boring processes by increasing the rate of penetration (ROP), tool lifespan, wellbore quality and minimising energy wastage during oil drilling, various drilling technologies have been developed over the past few years. A typical example of such technology is the rotary-percussive drilling which seeks to combine both rotational and percussive drill-bit movements in cutting through the rock layers [1–5]. As illustrated in Fig. 1(a), the percussive impact of the bit fractures the rock while its rotation cuts and shears away the fractured weak zone initiated by the percussive impact, thus exposing a fresh rock surface for the next percussive impact. It is also noticed that the introduction of intermittent impact actions instead of the continuous grating action between the drill-bit and the drilled formation accustomed to conventional rotational drilling helps to reduce the contact time between the drill-bit and the drilled formation, thus reducing bit wearing. The technique thus tries to combine the

advantages of both rotary and percussive drilling to increase the ROP and the life span of the drill-bit [6, 7]. Recent developments which are similar to the rotary-percussive drilling technique use high frequency vibro-impacts generated very close to the drill-bit in the place of the low frequency percussive impacts from the drill head. The impacts are generated on the bit by introducing axial oscillations in the form of a force or displacement into the drilling assemblages to instigate a resonance effect. Examples of these developments include the Self-excited Percussive-rotary Drilling [2], the Resonance Enhanced Drilling (RED) [4], also referred to as the high frequency vibro-impact drilling [8], the resonance hammer drilling [3, 7, 9] and the vibration assisted drilling [5]. RED technology uses controllable high frequency oscillations alongside drill-string rotation to generate high amplitude axial vibrations on the drill-bit. Both the rotary and oscillatory loading are controlled to ensure the resonance between the drill-bit and the drilled formation. However, due to the constantly changing downhole conditions and formation, the rotational and oscillatory loading need to be constantly adjusted to meet the drilled formation resonance condition. Therefore, a prediction system that can estimate the natural frequency of the drilled formation is needed and the frequency of oscillation and rotation of the bit should be chosen close to the predicted natural frequency to enable bit-rock resonance. For this present study, the focus will be on bit-rock interaction prediction for percussive drilling using transient dynamics. Fig. 1(b) shows the impact oscillator as a physical model for studying bit-rock interaction during percussive drilling. The use of the impact oscillatory system in mimicking the bit-rock impacting system of percussive drilling has not only provided an inexpensive means of understanding the mathematics behind the impact responses, but has also allowed the qualitative investigation of the long-time behaviour of the system at different operating conditions, e.g. see [8–17]. So the impact oscillator will be theoretically studied in this present work.
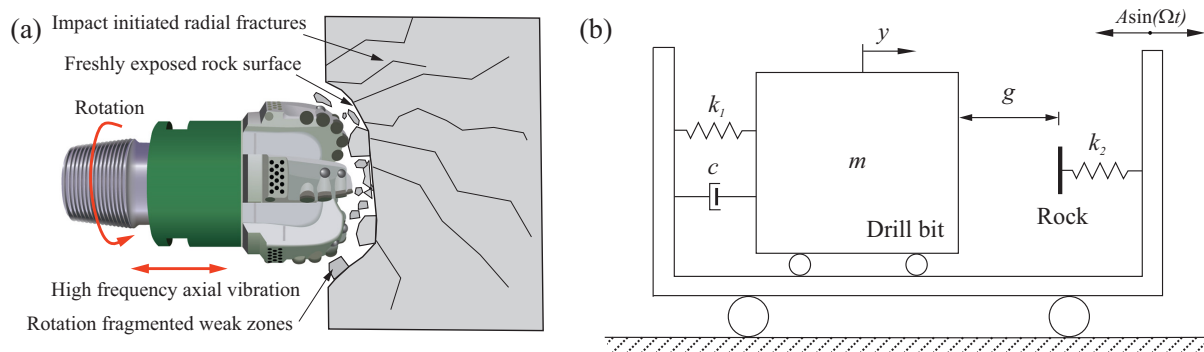

Figure 1: (a) Rock fragmentation mechanism of a rotary-percussive drilling system and (b) Physical model of the bit-rock interaction during percussion.

When studying vibro-impact drilling and its optimal strategies, Franca and Weber [3] investigated optimal parameter choice using a drifting oscillator and concluded that ROP significantly increases with increasing forcing amplitude under a period-one response with one impact per period of excitation. Pavlovskaia et al. [8] also confirm that a higher ROP of a vibro-impact drifting system can be achieved when the impact responses are kept in the period-one motion with one impact per period of excitation. Single parameter bifurcation analysis revealed three conditions for which optimal ROP can be achieved [18]. These include that the impact response must be periodic, the frequency of the vibrational responses should be the same as the frequency at which the system is excited. Liao et al. [10] carried out a bifurcation analysis for a bit-rock interaction model indicating the average variation of ROP with respect

to different vibro-impact responses. The results showed that the period-one responses with one impact and the period-two responses with two impacts produced better ROP compared to other responses. The implication of the above findings is that, despite fulfilling the resonance condition required for percussive drilling, the category of the resulting vibro-impact responses is crucial to the performance of the system and the high performance categories should be maintained during the entire drilling operation. Studies involving single and double parameter bifurcation analysis [8, 10] have revealed the influence of system parameters and initial conditions on resulting impact responses. As a consequence, due to the inhomogeneity of the underlying rock, the vibro-impact responses of the percussive drilling system will tend to change from one category to another over the entire period of drilling. Resulting impact response can either be desirable, which will need to be maintained or undesirable, which will need to be avoided before encountered. Drilling with chaotic or undesirable impacts will not only lead to reduced ROP but also a wastage of energy in the system. This thus brings about the need to continuously characterise impact responses during the drilling process, and this should be done as early as possible for the purpose of early optimisation of the system.

As a strategy for optimising the percussive drilling system, the present study focuses on investigating a data driven machine learning method for distinguishing between prevailing vibro-impact motions especially when the system parameters and initial conditions result into two or more co-existing impact responses. For this purpose, transient measurable system variables in the form of time series data will be used. The transient measurements are presented to a neural network as a set of labelled single or high dimension time series representing the early transient state dynamics of the system and are used in predicting the final steady state impact response of the system. This is quite different from learning to forecast the system's state variables per time-step but rather, learning the complex nonlinear pattern associated with the time–asymptotic trajectories traced out by co-existing impact responses as they progress through transiency to stability. The learnt nonlinear relationship is used in predicting the approached or final steady state response of the system from early measurable variables. A foreseen major concern as to the use of this approach in predicting steady state impact responses from their transient dynamics lies in:

1. Determining which of the measurable dynamical variables best differentiate co-existing impact responses in their transient state;

2. The periods of early dynamical variables that is enough to sufficiently train the network to a significant performance;

3. The hyperparameter combinations that will yield the best network configuration; and

4. The ability of the developed network to generalise for similar impact responses generated from different system conditions.

The present study aims to investigate and proffer solutions to these concerns and to our best knowledge, this is the first time an attempt will be made on characterising dynamical impacts like those of the bit-rock impacting system using readily measurable system variables and a non-computationally intensive technique with no much expert input. The rest of this paper is organised as follows. The modelling of dynamical systems using deep learning and Long Short-Term Memory (LSTM) network as a Deep Recurrent Neural Network is briefly studied in Section 2. In Section 3 the mathematical model of a soft impacting system as a representation of bit-rock impact interaction is presented. The proposed method

for steady state prediction from transient data is studied in Section 4 where its effectiveness and capability to generalise in other scenarios is also evaluated. Experimental validation and proposed improvement to impact motion classification problem is presented in section 5 while the conclusions and future work are given in Section 6.

2. Deep Learning and Dynamical System Modelling

Deep learning, initially referred to as hierarchical learning was first introduced in 2006 as a new technique for machine learning [19]. It involves the use of stacked multiple hidden layers of artificial neural networks to self-extract relevant high-level complex abstractions from complicated data sets. Due to the multiple hidden layers and nodes between their inputs and outputs layers, deep neural network models have more complex architectures compared to standard neural networks. Their ability to yield improved results and optimise the processing times for both linear and complex nonlinear relationships has enabled their utilisation in several real life applications. Typical examples of deep neural networks include the Deep Belief Networks, Convolutional Neural Networks, Recurrent Neural Networks and Recursive Neural Networks. Recurrent Neural Networks (RNNs) are feedforward neural networks with recurrent hidden states which are usually activated by the previous states of the network at some time backwards. This gives them the ability to dynamically model contextual information in time sequence data of arbitrary lengths. For example, they have been used in approximating and modelling multiple attractor dynamics [20–23], dynamic system phase space learning [24] and dynamical systems characterisation and identification [25, 26]. However, their adaptation to modelling multiple attractor dynamics using standard gradient descent training method has been reportedly difficult [27]. Doya and Yoshizawa [28] developed an adaptive neural oscillator network which was able to memorise and regenerate time-periodic patterns contained in a simulated waveform of two distinct limit cycles. On the other hand, Tsung, [29] demonstrated that doing so becomes very difficult when the waveforms exceed more than three limit circles due to the conflict of connection weights. The conflicting connection weights cause the gradient of the RNNs to either explode or vanish during their backpropagation process when modelling long time sequence tasks[30]. Vanishing gradient may cause the network to only capture short-term dependent information while information from earlier time steps continue to decay. Exploding gradient on the other hand may also cause the backpropagated error to grow drastically at each time step which could affect the convergence of the network [30, 31]. Several approaches have been put forward to address these problems when learning multiple attractor dynamics from several time series, such is the mixture-of-expert-type architecture [32, 33] and the parametric bias method [34, 35] which are both RNN based. The first method uses RNNs as expert and a hierarchical gating mechanism to learn the local representation of the the attractor dynamics but does not generalise about them. The parametric bias RNNs individualises each sequence data using information derived from the parametric bias values and are able to learn greater time series compared to those without parametric bias. However, these methods do not totally avoid the conflict arising from learning multiple attractors from long time series and the use of LSTM networks has previously been explored [36, 37]. This study will focus on using LSTM networks as a type of RNNs for learning multiple attractor dynamics with little or no data pre-processing for the purpose of deciphering approached periodic impact motions during a vibro-impact drilling operation.

2.1. Long Short-Term Memory networks

LSTM networks use multiple internal layers of nonlinear processing units to conduct supervised or unsupervised deep learning, and unlike others they have memory cells and gates in replace of self-connected

5

hidden units. These components help the LSTM networks to minimise the gradient descent problem effect. During learning, the gates either permit the addition or removal of information from the memory cells, thus enabling the networks to map long term dependencies from distant time series while minimising the problem of vanishing or exploding gradient. Another advantage and a good reason for using LSTM networks is that as deep networks they allow the mapping of concurrent multivariate input sequences into a single output.
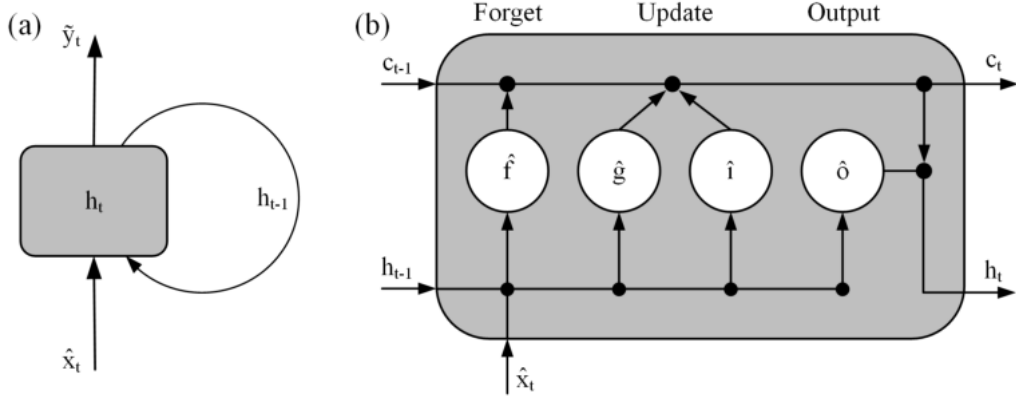

Figure 2: Components and flow of data in (a) a normal RNN and (b) a LSTM network unit layer.

Figs. 2 (a) and (b) illustrate the components and the flow of data at time step t through a normal RNN and a LSTM network unit layer. The LSTM layer state is defined by the cell state c and the output or hidden state h. The hidden state at time t contains the information learned by the network at that time step while the cell state contains information learned from the previous time steps. As seen in Fig. 2(b), a single LSTM unit consists of the input gate $\hat{i}$ controlling the level at the cell state is updated with new information from the present time-step, the forget gate $\hat{f}$ controlling the level at learned information are retained in the cell or discarded and the output gate $\hat{o}$ controlling the amounts of the information in the cell state used in computing the output state layer. The memory cell $\hat{g}$ helps keep track of the dependencies between the input sequence vectors. At time step t, the LSTM layer takes the initial states of the network, c_{t-1} and h_{t-1}; and also the current time step sequences $\hat{x}_t$ to compute the new learned information h_t and an updated cell state, c_t. At each time step during training, the layer either removes information using the forget gate $\hat{f}$ or add information using the input gate $\hat{i}$ and memory cell $\hat{g}$. The layer uses sigmoid activation function (Eq. (1)) to scale the input between 0 and 1 in the gate cells, and uses an element-wise multiplication function (hadamard product $\odot$) and a hyperbolic tangent function (Eq. (2)) whose outputs are between -1 and 1 in calculating the updated states of the layer per time step (Eqs. (4) and (5)).

$$\sigma(u) = (1 - \exp^{-u})^{-1} \tag{1}$$

$$\tanh u = \frac{1 - \exp^{2u}}{1 + \exp^{2u}} \tag{2}$$

The learnable objects of an LSTM layer include the input weights (W), the recurrent weights (R) and the bias (b). The values of these objects as put out from each LSTM unit component are concatenated into matrices W, R and b as shown by Eq. (3).

6

$$W = \begin{bmatrix} W_i \\ W_f \\ W_g \\ W_o \end{bmatrix}, \qquad R = \begin{bmatrix} R_i \\ R_f \\ R_g \\ R_o \end{bmatrix}, \qquad b = \begin{bmatrix} b_i \\ b_f \\ b_g \\ b_o \end{bmatrix} \tag{3}$$

Table 1: The formula for each component state vectors at time t

Component	Formula
Input gate ($\hat{i}_t$)	$\sigma_g(W_i \hat{x}_t + R_i h_{t-1} + b_i)$
Forget gate ($\hat{f}_t$)	$\sigma_g(W_f \hat{x}_t + R_f h_{t-1} + b_f)$
Memory cell candidate ($\hat{g}_t$)	$\sigma_g(W_g \hat{x}_t + R_g h_{t-1} + b_g)$
Output gate ($\hat{o}_t$)	$\sigma_g(W_o \hat{x}_t + R_o h_{t-1} + b_o)$

Table 1 shows the formula for computing each gate's component vectors at time t, while the cell state and the hidden state are computed as

$$c_t = \hat{f}_t \odot c_{t-1} + \hat{i}_t \odot \hat{g}_t \tag{4}$$

$$h_t = \hat{o}_t \odot \tanh(c_t) \tag{5}$$

where $\hat{i}_t$, $\hat{f}_t$, $\hat{g}_t$ and $\hat{o}_t$ are the computed state vectors with the values between $[0, 1]$, and are of the same size as the cell output vector h_t.

3. Mathematical Model of Bit-Rock Interaction

The impact oscillator shown in Fig. 1(b) is a piecewise smooth system resembling the intermittent impact action between the drill-bit and the rock. It consists of a mass m representing the drill-bit and it's connected to a rigid frame via a linear spring with stiffness k_1 and a damping coefficient c. The secondary linear spring with the stiffnes k_2 acting as the rock media is attached to the opposite side of the frame, and its stiffness may be varied to represent different types of rocks. As for vibro-impact drilling, the rigid frame is subjected to an external harmonic excitation with the amplitude A and the frequency ω. Mass displacement y is closely monitored in order to compare with the initial gap g between the mass and the secondary spring. If $y > g$, an impact between the mass and the secondary spring occurs. Therefore, the equation of motion of the impact oscillator can be written in a nondimensional form as

$$\begin{cases} x' = v, \\ v' = a\omega^2 \sin(\omega\tau) - 2\zeta v - x - \beta(x - e)H(x - e), \end{cases} \tag{6}$$

where x' and v' are respectively the displacement and velocity differentiates with respect to the dimensionless time τ and $H(\cdot)$ stands for the Heaviside step function. The variables and the parameters in Eq. (6) have been nondimensionalised as

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

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.

Fig. 3 shows the displacement and the acceleration time histories of the impact oscillator obtained using Eq. (6) for (a) a period-one response with one impact per period of excitation (P-1-1) and (b) a co-existing period-one response with two impacts per period of excitation (P-1-2). The trajectories traced out by the two co-existing motions before and after reaching their stability are displayed on the phase plane (x, v) in the additional windows of the figure. Fig. 4 presents the Poincaré sections of these two motions as they progress from transiency to stability. The solutions to the equations of motion were programmed and run on MATLAB using the iterative fourth order Runge-Kutta method of solving ordinary differential equations at a fix step-size per iteration. As expected of dynamically impacting system, the impact oscillator shows complex non-linearity between the system parameters and the resulting impact responses, while its early periods iteration is characterised with dynamic instability. These effects have made it difficult to establish a functional "if and only if" condition or a universal rule for characterising impact responses from operating system parameters (i.e. ω, a and β) or from the earlier periods of measurable system variables, such as displacement x, velocity v and acceleration (v'). However, in this study an analytical tool with the potential of approximating unknown relationships from observable time histories, such as Fig. 3, and the convergent pattern as illustrated in Fig. 4 is investigated by using the LSTM networks.

3.1. Basins of attraction

During the operation of a dynamical system, the early periods are characterised with intense instability and transiency but after some periods, it is expected that the dissipation and driving force will become stable, annulling the initial transients found within the system and then settle it to its typical dynamic behaviour, as shown in Figs. 3 and 4. A closure of initial conditions whose long-time behaviour converges to a fixed position, namely attractor, is called the basin of attraction. In some cases, a closure of initial conditions can result into more than one point of attraction, which is referred to the basins of co-existing attractors. During percussive drilling operation, the drilling parameters and the inhomogenous rock media can cause the drilling system to exhibit multiple co-existing attractors, which may degrade drilling efficiency. In the present work, five basins of attraction presenting two classes of co-existing impact motions (P-1-1 and P-1-2) as shown in Fig. 5 were used to study the LSTM-based approach. For each figure, there are 1681 initial values calculated by using Eq.(6), and their system parameters and the statistics of the resulting motions are presented in Table 2.

Table 2: System parameters and the numbers of initial values for the co-existing motions.

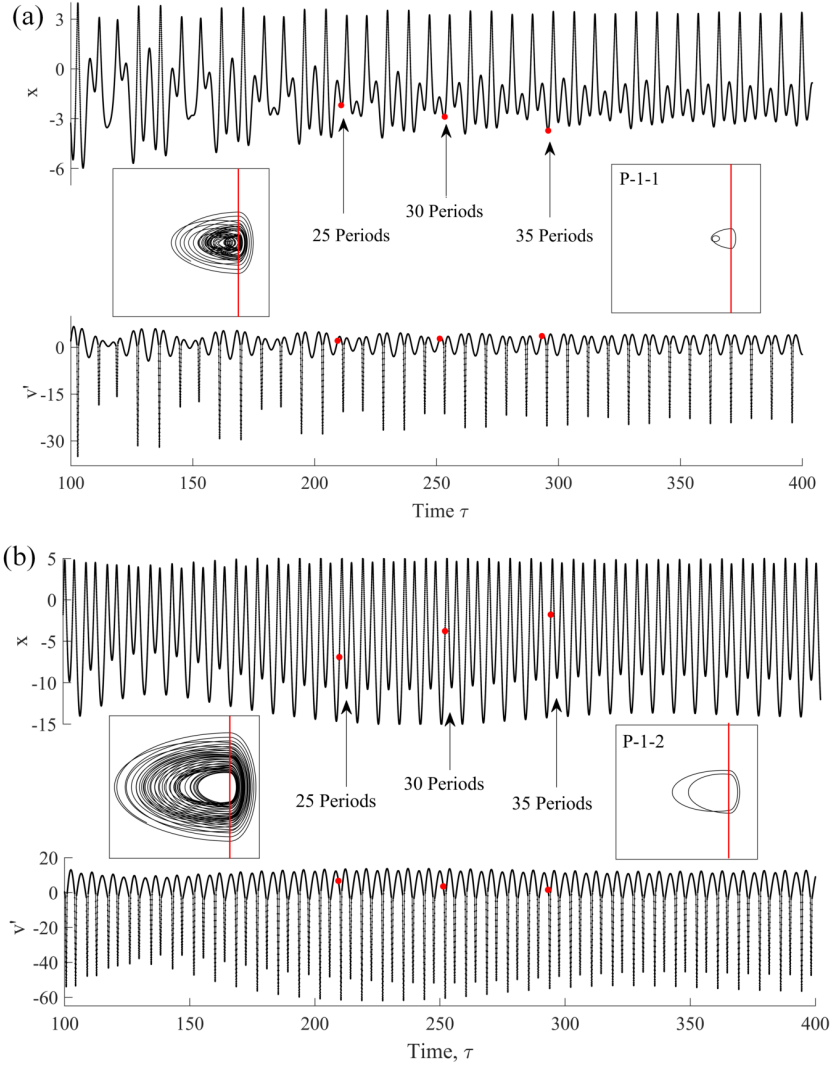| Basins	a	ω	e	ζ	ϕ	β	P-1-1	P-1-2
BSN1	5.6	0.75	2.1	0.01	0	18.27	795	886
BSN2	5.6	0.75	2.1	0.01	0	29.0	402	1279
BSN3	4.8	0.75	2.1	0.01	0	18.27	1386	295
BSN4	5.6	0.73	2.1	0.01	0	18.27	432	1249
BSN5	4.8	0.73	2.1	0.01	0	29.0	485	1196

Figure 3: Time histories of displacement (x) and acceleration (v') of the impact oscillator for (a) a period-one with one impact (P-1-1) and (b) a coexisting period-one with two impacts per period of external excitation (P-1-2) computed for $a = 5.6$, $\omega = 0.75$, $\zeta = 0.01$, $\beta = 18.27$, and $e = 2.1$ by using the initial values $(1, 1)$ and $(0, 0)$, respectively. Left and right panels show the transient and the steady-state responses of the system on the phase plane (x, v), respectively. Red line indicates the impact boundary between the mass and the secondary spring.

4. Supervised Steady State Prediction

4.1. Network development and optimisation

As discussed earlier, aside the complex nonlinearity that characterises the impact oscillator, it is difficult to predict a system's attractor using its transient response measurements. This is due to the inherent instability in system's impacting force at transient periods, but with time the impacting force becomes stable and suppresses the initial transients found within the system as presented in Figs. 3 and 4. For the purpose of characterising vibro-impacts from dynamically transient variables, the ability of the LSTM network as a deep recurrent neural network to learn and differentiate between the nonlinear transient trajectory (shown in Fig. 4) traced out by the co-existing impact motions from raw and unprocessed system variables is explored in this section. In the present work, different models of multi-layered LSTM
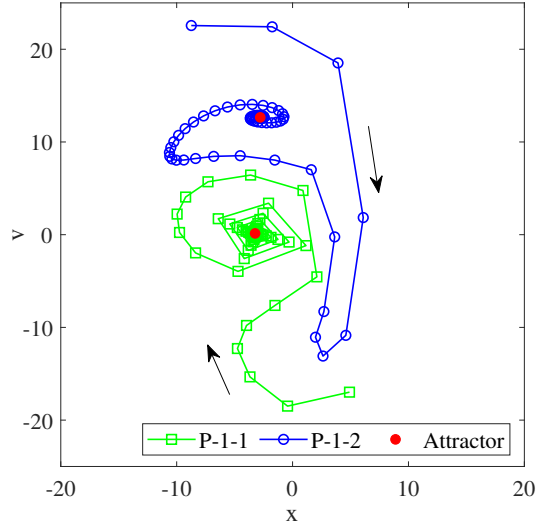
9

Figure 4: Poincaré sections of P-1-1 and P-1-2 on the phase plane (x, v) from transiency to stability. Arrows indicate the evolution of the Poincaré sections approaching to their attractors.

networks were developed in MATLAB environment to learn the dynamics of the prevailing impact motions from their transient displacement x, velocity v, acceleration v', impact boundary x_e, and periodic boundary x_p. As a time series, the impact boundary, x_e, is the value of e representing the maximum allowable displacement between the impacting mass and the impacted constraint. The periodic boundary, x_p, marks the end of a complete periodic response. These data were presented to the network either as a single series of measured variable

$$X = \begin{bmatrix} X_1 & X_2 & X_3 & ... & X_{n-2} & X_{n-1} & X_n \end{bmatrix}, \tag{7}$$

or as a high dimensional series of concurrently measured variables

$$\begin{bmatrix} x \\ v \\ x_e \\ x_p \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & ... & x_{n-2} & x_{n-1} & x_n \\ v_1 & v_2 & v_3 & ... & v_{n-2} & v_{n-1} & v_n \\ x_{e,1} & x_{e,2} & x_{e,3} & ... & x_{e,n-2} & x_{e,n-1} & x_{e,n} \\ x_{p,1} & x_{p,2} & x_{p,3} & ... & x_{p,n-2} & x_{p,n-1} & x_{p,n} \end{bmatrix}, \tag{8}$$

representing the system's condition and position in state space through time, where n is the total number of time steps. The notation and dimensions of the developed network models based on the input variables are summarised in Table 3.

By using the standard means of time series learning, the network translates the impact mode identification into a pattern recognition problem and learns temporal information, such as the consistently repeating patterns which are identical to the different impact modes rather than learning their signal values. The network learns to predict the approached attractor by learning to differentiate between sequences of repeating patterns. Therefore, the network not only considers the quantitative variation of the time series vectors but also their pattern in time space as the system progresses through its transient state.

As illustrated in Fig. 6, each of the developed networks consists of a sequence input layer, a bi-
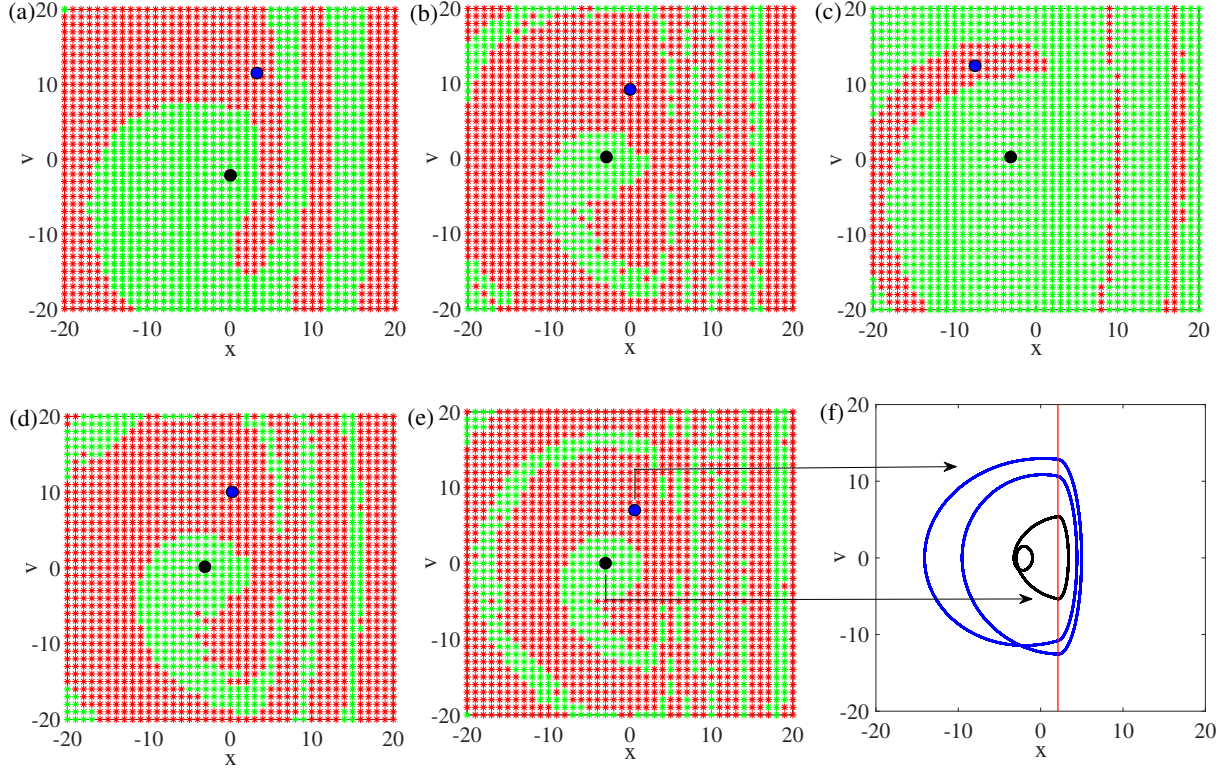
Figure 5: Evolution of basins of co-existing impact motions for (a) BSN1, (b) BSN2, (c) BSN3, (d) BSN4, and (e) BSN5. Black dots with green basins denote the P-1-1 motion, and blue dots with red basins indicate the P-1-2 motion. Blue and black lines in (f) represent the phase trajectories of the P-1-1 and the P-1-2 motions, respectively. Red line indicates the impact boundary between the mass and the secondary spring.

Table 3: Details of the developed LSTM networks, where 'nrm' and 'nsy' denote the normalised and the noise effected data, respectively.

Network Models	Input Data	Input Dimension
BiLSTM-x	x	1×16110
BiLSTM-v	v	1×16110
BiLSTM-a	v'	1×16110
BiLSTM-xv	$[x, v]^T$	2×16110
BiLSTM-ap	$[v', x_p]^T$	2×16110
BiLSTM-xvep	$[x, v, x_e, x_p]^T$	4×16110
BiLSTM-axep	$[v', x, x_e, x_p]^T$	4×16110
BiLSTM-vaep	$[v, v', x_e, x_p]^T$	4×16110
BiLSTM-nrm	$[x, v, x_e, x_p]^T_{\mathrm{nrm}}$	4×16110
BiLSTM-nsy	$[x, v, x_e, x_p]^T_{\mathrm{nsy}}$	4×16110

directional LSTM (BiLSTM) layer defined with a *"last"* output mode and a fully connected layer. Being a classification problem, a softmax layer and a classification output layer were added after the fully connected layer. Unlike the standard LSTM networks, BiLSTM networks can learn bidirectional long-term dependencies between preceding and succeeding sequential data using forward and backward hidden layers. The data are presented to the network as $[j \times n]$ matrix within a $\{M \times 1\}$ cell arrays, where j is the number of features per time step representing the network input size, and M is the number
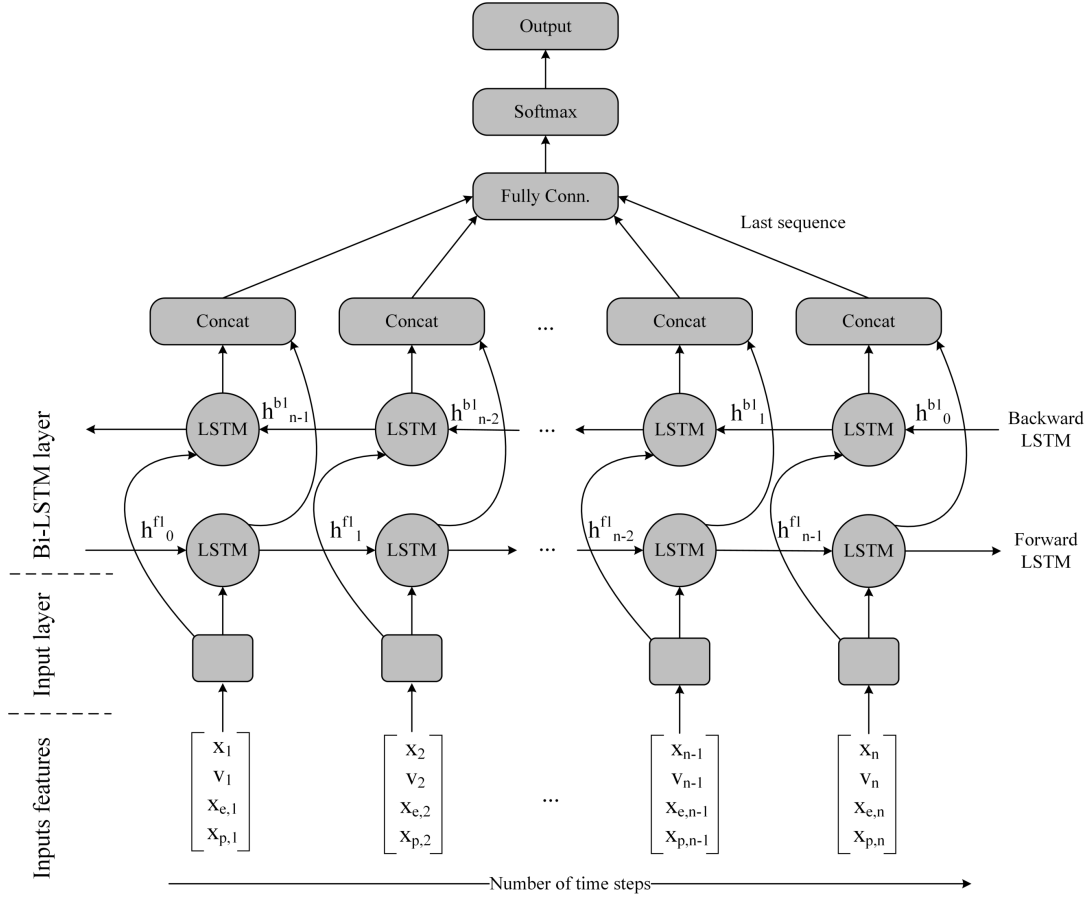
11

Figure 6: Architecture of the developed bi-directional LSTM network.

of cells of samples resulting from each basin, which in this case is 1,681 based on the total numbers of initial conditions. The targeted impacting motions were defined as categorical arrays with *"P-1-1"* and *"P-1-2"*. To ensure that the best network was developed, different models of LSTMs using (1) univariate, (2) multivariate, (3) noisy multivariate, and (4) normalised multivariate sequences of transient dynamical responses were tested. As it is necessary to deduce the minimum periods of transient data and the set of hyperparameters that are sufficient to optimise the network, a number of hyperparameter tuning was carried out as presented in Fig. 7 to obtain the best network configuration. In the training options, both the stochastic gradient descent with momentum "sgdm" and the adaptive moment estimation "adam" were alternated and compared during the network restructuring.

It can be seen from Fig. 7 that both optimisers performed better at lower mini-batches of data and increasing numbers of hidden units, but the "sgdm" optimiser at some points suffered a setback in its performance once the hidden units were increased above 30. Such a reduction in the performance of the "sgdm" optimiser may be attributed to overfitting effect resulting from an overtrained network. Instead of learning generalised information, the network learns patterns that are unique to the inputted data set resulting in a large gap between the training and the testing errors. Therefore, after optimisation, the final network consisting of a five-layer RNN-BiLSTM network with 60 hidden neurons in each layer was obtained as presented in Fig. 6. Each training was carried out using "adam" optimiser with the mini-batch number and the training epochs set to 10 and 35, respectively, while applying a dropout of
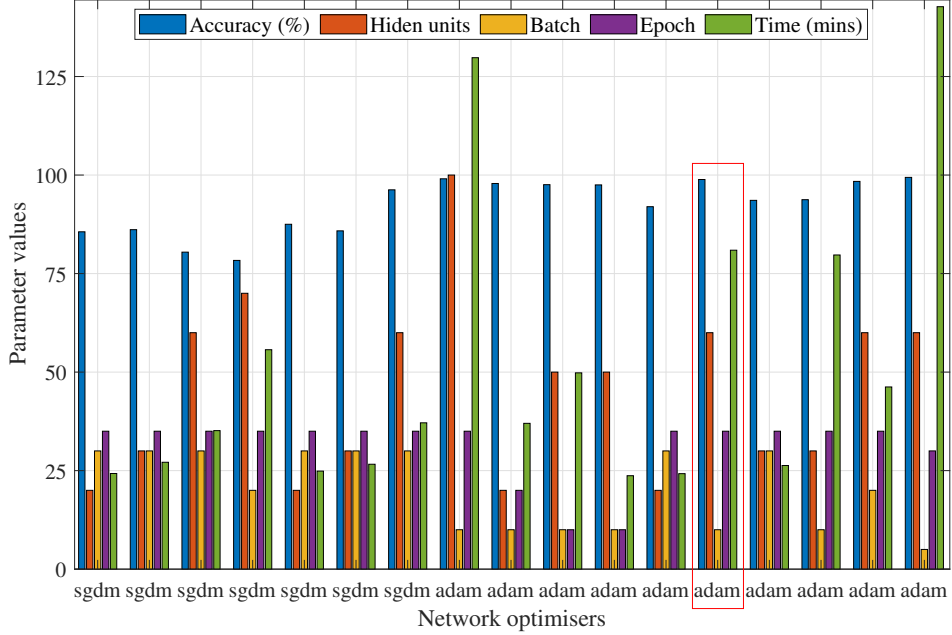
Figure 7: Hyperparameter tuning using the "sgdm" and "adam" optimiser. Red rectangular indicates the final selected network after optimisation.

0.02 after each layer. The final output layer is activated with a sigmoid function that uses mean square error as its loss function and each training took about 35 minutes on a 16GB RAM intel core i7-7700 CPU. The probability that an input sequence X belongs to a class C_k is given as

$$p(C_k \| X) = \frac{\exp(\hat{o}_m)}{\sum_{k=1}^{k} \exp(\hat{o}_k)}, \quad m = 1, 2, ..., k. \tag{9}$$

where k is the number of classes, $\hat{o}$ is the output vector from the output gate and the input vector into the softmax layer.

4.2. Developed network models and performance

As listed in Table 2, BSN1 comprises 795 initial values for the P-1-1 motion and 886 for the P-1-2 motion. A subset of this basin as indicated in Fig. 8, consisting of 201 pairs of initial values that result into P-1-1 motion and 219 points that lead to P-1-2 motion, was used for generating the time history data used in training all the various types of network models listed in Table 3. The remaining part of the basin (BSN1) and the other basins (BSN2-5) were used for testing the networks. The training data (i.e. a total of 420 initial values of BSN1) thus sum up to be approximately 5% of the total simulated data considering that five basins of 1681 data points each (i.e. 8405 data), were simulated for this study, while the remaining 95% were used as unknown data for testing the developed networks.

The details and the performances of the developed network models during training and testing on BSN1 are presented in Table 4. All the network models showed good performance both in training and in testing, although both the training and testing data sets have come from the same basin. To further evaluate the performance of these network models, each of them was tested on BSN5, which is an entirely different basin but with the same P-1-1 and P-1-2 motions. BSN5 consists of 485 and 1196 points leading to P-1-1 and P-1-2 motions, respectively. The results of using BSN1 developed network models
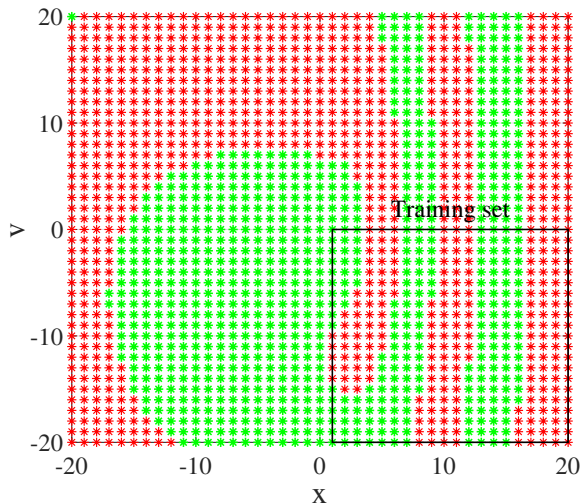
13

Figure 8: BSN1 shows the subset of data used for developing the network.

on BSN5 are presented in Table 5. Some of the network models are seen to have performed very poorly, especially on the P-1-2 motions going as low as 0%. This indicates the low capacity of such networks and their defining variables to capture the nonlinear relationship that distinguishes between the two impact motions.

Table 4: Performance of the developed network models on BSN1. (OVA=Overall Accuracy)

Network models	Training accuracy (%)			Testing accuracy (%)			Training time (mins)
	P-1-1	P-1-2	OVA	P-1-1	P-1-2	OVA	
BiLSTM-x	94.0	100	97.1	98	100	99	35.05
BiLSTM-v	100	99.5	99.8	100	100	100	36.83
BiLSTM-a	99.0	99.5	99.5	99.5	100	99.8	35.03
BiLSTM-xv	100	100	100	100	100	100	36.12
BiLSTM-ap	97.0	100	98.6	98.8	100	99.4	35.04
BiLSTM-xvep	100	100	100	100	100	100	35.52
BiLSTM-axep	99.0	100	99.5	99.3	100	99.7	35.15
BiLSTM-vaep	100	100	100	100	100	100	35.35
BiLSTM-nrm	91.0	79	84.8	78.6	93.7	86.6	35.32
BiLSTM-nsy	100	100	100	100	100	100	35.5

BiLSTM-nsy network was developed and tested using noisy data sets, while BiLSTM-nrm was developed and tested using normalised data sets. This was done to investigate the effectiveness of the network in the presence of noise as should be expected in real life situation, and to study the contribution of data normalisation to the networks' learning ability. As shown in Fig. 9, an additive white Gaussian noise of signal-noise ratio 10 was added to the simulated time data, and the following formula was used to normalise the data sets,

$$X_{\text{norm}} = \frac{X - \min(X)}{\max(X) - \min(X)}. \tag{10}$$

Results from using the noisy data sets in Tables 4 and 5 show that the networks can significantly learn impact motion dependent features from the noisy data, as well as use the learning to generalise on

14

Table 5: Performance of the developed network models on BSN5. OVA=(Overall Accuracy)

Network models	Testing accuracy (%)		
	P-1-1	P-1-2	OVA
BiLSTM-x	99.4	4.3	31.8
BiLSTM-v	100	96.4	97.4
BiLSTM-a	98.6	3.1	30.6
BiLSTM-xv	100	94.7	96.3
BiLSTM-ap	100	0	28.9
BiLSTM-xvep	100	95.6	96.8
BiLSTM-axep	100	19.8	43.0
BiLSTM-vaep	100	94.1	95.8
BiLSTM-nrm	89.7	84.4	85.9
BiLSTM-nsy	100	94.1	95.8
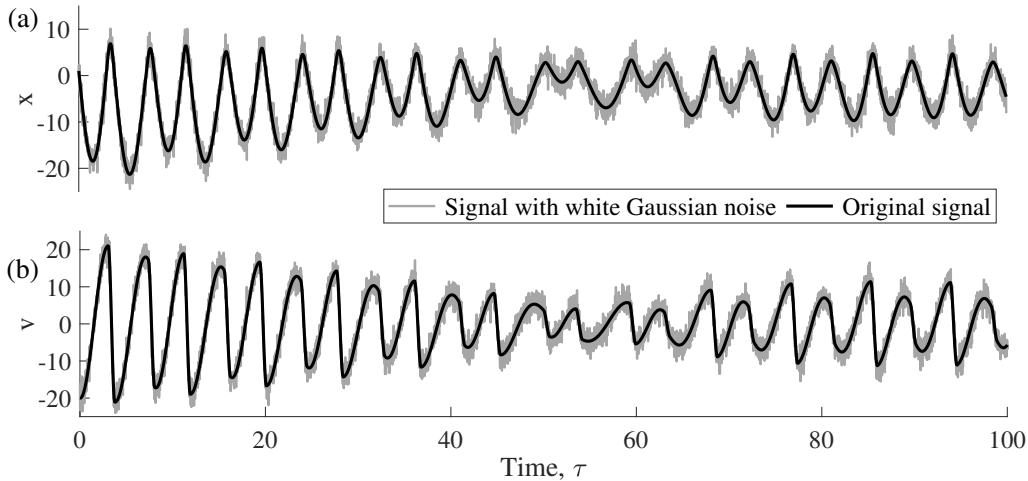
Figure 9: Displacement and velocity of the impact oscillator with additive white Gaussian noise (signal-noise ratio at 10) for testing BiLSTM-nsy network. Black and grey lines represent the original and the noisy signals, respectively.

unknown noisy data. On the other hand, Table 6 shows the results of using BiLSTM-xvep developed from noiseless data (BSN1) on noisy data from BSN1-5. It can be observed that the network performance shows no difference when compared with its performance on noiseless data of these basins. For the normalised data network, BiLSTM-nrm, the accuracy was found to be a little lesser compared to the non-normalised data network, BiLSTM-xvep. This was noticed for both the training and the testing data sets in Tables 4 and 5. It may be stated that the data normalisation probably caused the network to overfit. However, it is worth mentioning that the training of the normalised data network did converge 12 seconds faster compared to the non-normalised data network, but such a trade-off is not worth the reduced performance. Here, reducing the number of hidden units in the network when training with normalised data may help to counteract the overfitting effect.

According to Tables 4 and 5, six networks, BiLSTM-v, BiLSTM-xv, BiLSTM-xvep, BiLSTM-vaep, BiLSTM-nrm, and BiLSTM-nsy, have performed well in predicting the final steady state response of the system by learning to distinguish between transient data of co-existing impact motions. Further validation by testing some of the networks on new data set from BSN2, BSN3, and BSN4 showed BiLSTM-v, BiLSTM-xv, and BiLSTM-xvep to be consistent in their performance compared to BiLSTM-a. BiLSTM-

Table 6: Testing BiLSTM-xvep on noisy data. (OVA=Overall Accuracy)

Basins	Testing data		Testing accuracy (%)		OVA (%)
	P-1-1	P-1-2	P-1-1	P-1-2	
BSN1	594	667	100	100	100
BSN2	402	1297	100	98.3	98.7
BSN3	1386	295	86.6	100	88.9
BSN4	432	1249	100	100	100
BSN5	485	1196	100	95.6	96.8

v showed an accuracy of 98.2%, 88.7% and 100% for BSN2, BSN3 and BSN4, respectively, while both BiLSTM-xv and BiLSTM-xvep showed an accuracy of 98.7%, 88.9% and 100% for the same basins, respectively (see Table 7).

Table 7: Performance of BiLSTM-a, BiLSTM-v, BiLSTM-xv, BiLSTM-vaep and BiLSTM-xvep on prediction of BSN2-4 (OVA=Overall Accuracy)

Network models	Input dImension	BSN2			BSN3			BSN4		
		P-1-1	P-1-2	OVA	P-1-1	P-1-2	OVA	P-1-1	P-1-2	OVA
BiLSTM-a	1 x 16110	58	44.6	47.8	56.6	34.2	52.7	64	66.2	65.7
BiLSTM-v	1 x 16110	100	97.6	98.2	86.3	100	88.7	100	100	100
BiLSTM-xv	1 x 16110	100	98.3	98.7	86.5	100	88.9	100	100	100
BiLSTM-vaep	1 x 16110	97.5	3.1	25.7	89.4	0.0	73.7	100	96.8	97.6
BiLSTM-xvep	1 x 16110	100	98.3	98.7	86.6	100	88.9	100	100	100

Table 8: Overall accuracy of BiLSTM-xvep on different basins with various periods of training data.

Network models	Input dimension	BSN1	BSN2	BSN3	BSN4	BSN5
25 Periods	4×13425	100%	90.4%	85.9% %	93.9%	85.8%
30 Periods	4×16110	100%	98.7%	88.9% %	100 %	96.8%
35 Periods	4×18795	100%	98.7%	88.9% %	100 %	96.8%

The performance of networks developed from different periods of data indicated in Fig. 3 was also explored. Three BiLSTM-xvep networks were developed and tested using 25, 30 and 35 periods of transient data, respectively. Their performances were seen to increase with increasing periods of data as presented in Table 8. In general, Tables 5 and 7 show inconsistent performances in the ability of the BSN1 developed networks to classify the P-1-1 and P-1-2 motions across the out-of-sample basins (i.e. BSN2-BSN5). To understand the reason for the variation, a period to stability plot was developed for the basins with respect to their defining initial conditions. Fig. 10 is a plot of the solutions of the system at the end of each period along the velocity plane for (a) P-1-1 and (b) P-1-2 motions. Given from the 50th period to the 130th period, the periodic solutions are seen to oscillate at initial periods before leveling out to the final steady state solution at later periods. As illustrated in the figure, given that Z is the solution to the system at its final steady state (i.e attractor's position) along one of its measurable variable, the periods to stability was calculated as the number of periods expended before the solution began to periodically oscillate around Z_{2D}, where Z_{2D} is Z approximated to two decimal places. The average numbers of periods before stability calculated for each of the basins are presented in Fig. 11.

Reconstructed basins using the BiLSTM-xvep network are presented in Fig. 12 alongside their reconstruction confusion matrices and periods to stability plots, where the incorrectly predicted points are marked by blue circles. Comparing the reconstructed basins with their stability plots, it can be
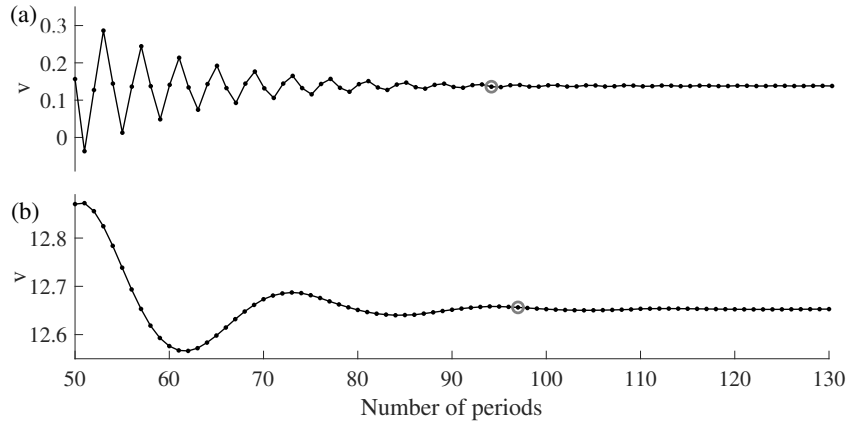
Figure 10: Variation in system velocity solutions from the 50th period to the 130th period for (a) P-1-1 and (b) P-1-2, where Z_{2D} is marked by a grey circle.
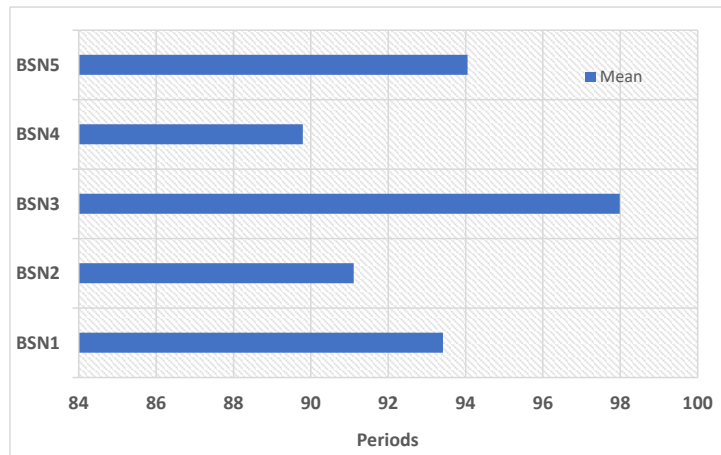

Figure 11: Average periods to stability across the basins .

observed that the regions where the network struggled to make accurate prediction matches the regions with higher instability rather than the basin boundaries or motion type. Such regions will require higher periods of data for training and testing in order to make accurate predictions as noted from Table 8. Among the basins, BSN3 shows the highest form of instability, ranging between 35.0 and 171.5. This explains why the networks overall performance for the basin never exceeded a certain range compared to their performances in other basins as seen in Tables 6-8.

Network development and performance presented in Table 4 for BSN1 shows the overall performance during testing to be slightly higher than the overall performance during training. This does not suggest an overfitting effect, as it would have if it was the other way round, but can be an indication of the effect of the uneven instability that characterises the basin as most of the training data (black box, Fig. 8) are from the unstable part of the basin. This simply means that a network trained on data sets from a region of higher instability will show better testing performance compared to the training performance when tested on data set from a more stable region. For each of the basins, the periodic motion that dominates the unstable region will have the highest tendency to be wrongfully predicted. This explains
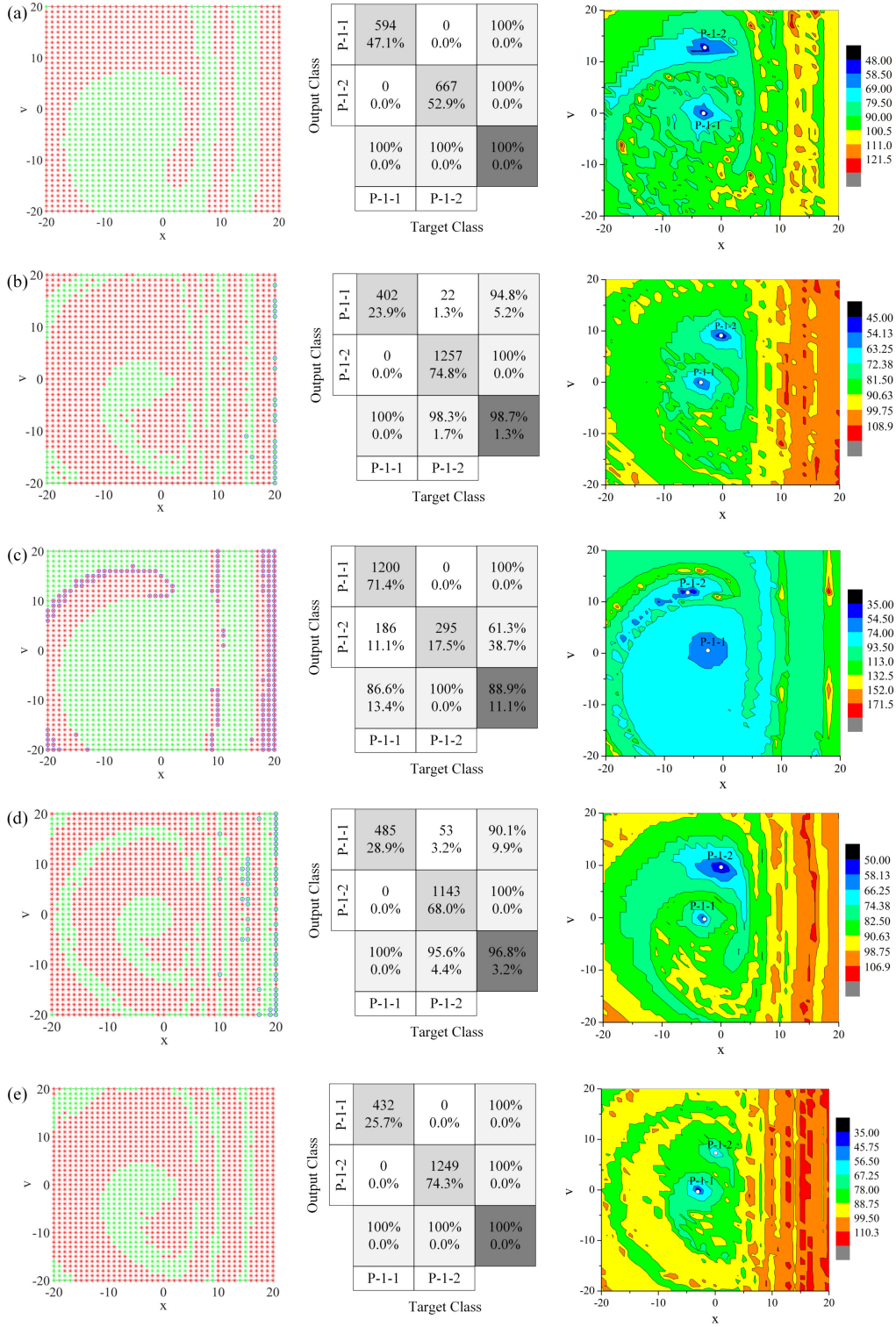
17

Figure 12: Reconstructed basins, confusion matrices and periods to stability maps for (a) BSN1, (b) BSN2, (c) BSN3, (d) BSN4, and (e) BSN5 by using the BiLSTM-xvep network, where the incorrect predicted points are marked by blue circles, and the color bars indicate the nondimensional time from the initial point to its attractor.

the reason why BSN3 (Fig. 12c) whose unstable region is dominated by P-1-1 motions unlike the other basins in figure 12, showed lesser performance in predicting the P-1-1 motions compared to the other basins which have the P-1-2 motions dominating their unstable regions. In general, the performances of the consistently performing networks, BiLSTM-v, BiLSTM-xv and BiLSTM-xvep, across the basins is seen to vary in order of their stability.

5. Experimental Validation

5.1. Experimental apparatus

To evaluate the practicability of the proposed approach for impact motion classification, experimental data based on a universal experimental rig developed at the Applied Dynamics and Control Laboratory at the University of Exeter were used in this study. The rig can be used for experimental testing of a number of piecewise-smooth dynamical systems, such as the impact oscillator studied in the present work, the drifting oscillator [18], and the capsule system [38]. Since the proposed LSTM-based method is a generic means of classifying periodic motions via their transient dynamics, it is applicable in various engineering systems. Thus it provides the rationale to use this rig to test the proposed method.

The photograph and schematic of the experimental rig are shown in Fig. 13, where a shaft was used as the hammer of percussive drilling and driven by the electrical coil. It was connected to the bit via a helical spring and its motion was limited to translation by a linear bearing. The hammer moves forward when the coil is switched on, compressing the helical spring. When the relative displacement between the hammer and the bit exceeds the front gap, the hammer hits the front constraint. When the coil is switched off, the hammer is pulled back by the restoring force of the compressed helical spring. If the relative displacement exceeds the back gap, the hammer impacts the back constraint. The bit either penetrates forward or drifts backward when the interactive force between the hammer and the bit overcomes the environmental friction.
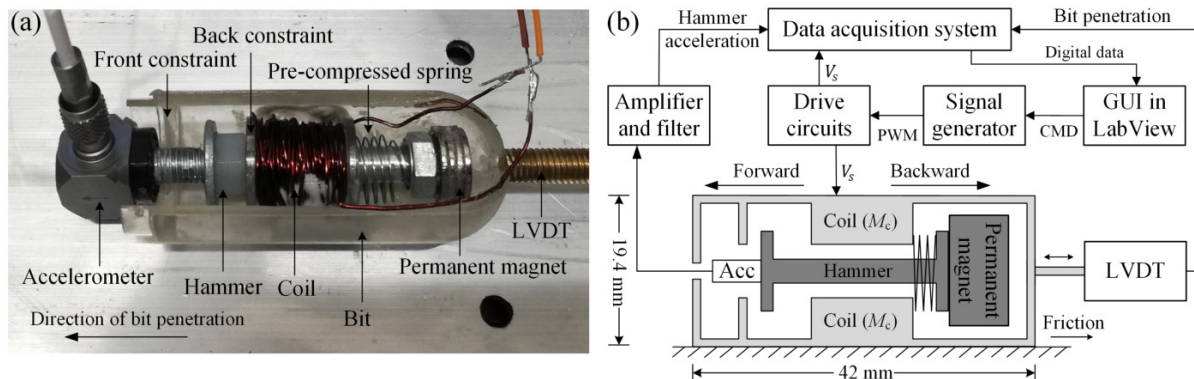

Figure 13: (a) Photograph and (b) schematic of the experimental rig. A solenoid is mounted inside the rig with the coil fixed to the inner surface of the rig and the shaft acting as a vibrating hammer excited by an on-off rectangular waveform signal. The hammer is connected with the coil via a pre-compressed helical spring at one end, and a nylon nut and an iron washer are fixed on the other end of the hammer. Impact will occur when the washer hits either the front or the back constraint. A linear variable differential transformer (LVDT) is attached to the rig, and an accelerometer is fixed to the hammer, measuring bit penetration and the acceleration of the hammer, respectively. Both hammer acceleration and bit displacement are collected by a data acquisition card through a graphic user interface (GUI) in LabView with the sampling frequency of 1 kHz. The GUI also sends command (CMD) to a signal generator to control the solenoid drive circuit by using the pulse-width modulation (PWM) signal [39].

Like percussive drilling, the direction and the average ROP of the experimental rig is dependent on the type of impact motions performed by the hammer within the bit. Experimental testing of this rig [38] has reported the overall rate of progression to be higher for motions with forward impacts, compared to those without impacts or with backward impacts. This implies that the optimisation of this system can largely benefit from the present study on impact motion characterisation, especially for scenarios when system parameters lead to multiple co-existing impact motions.

A major concern of vibration experiment is the easiness of data measurement. Time series data including displacement (x), velocity (v) and acceleration (v') describing vibration systems are easily calculated from the mathematical models representing such systems but this is practically difficult in experiments. This is in part due to the high cost of some of the measuring equipment and also due to the difficulty of integrating all the measuring equipment into a single experimental set-up, such as the one shown in Fig. 13. Over the time, acceleration (v') measurements using accelerometer has been found to be handy in studying vibration motions [40, 41]. Hence, only the acceleration (v') data of the experimental rig were used in this study. It should be noted that these experimental acceleration (v'_{exp}) data were for the impacting hammer and not for the bit itself. The impact actions of the hammer are similar to the ones considered in the impact oscillator, and they include a period–one motion with one impact (P-1-1) and a period-one motion with two impacts (P-1-2). As shown in Fig. 14, forward impacts are projected on the negative peaks of acceleration, while backward impacts are projected on the positive peaks. The motions have been derived by operating the system on different parameters, which is similar to having oscillatory impact motions from different basins of attraction.
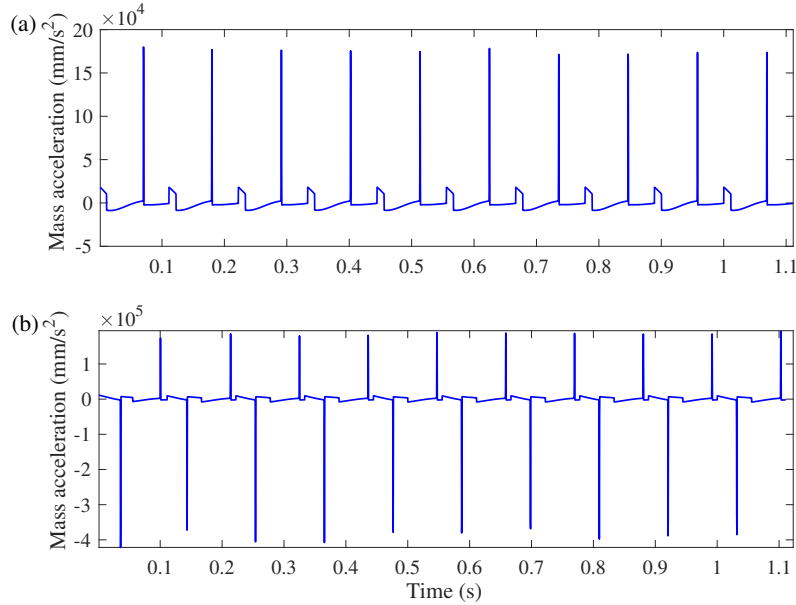

Figure 14: Time histories of acceleration of the hammer: (a) period-one motion with one impact (P-1-1) and (b) period-one motion with two impacts (P-1-2). The forward impacts are projected as the negative peaks, while the backward impacts are projected as the positive peaks.

The acceleration based LSTM network (BiLSTM-a) developed from the simulated accelerations of the impact oscillator (i.e. v') was used for classifying the experimental acceleration data (i.e. v'_{exp}). As part of the pre-processing procedures, v'_{exp}, was firstly scaled to the range of v'. This was done by normalising

v'_{exp} with the minimum and maximum values of v'. Secondly, the input experimental data were structured to have the same dimension (30 periods each) as the input data used in training BiLSTM-a. The confusion matrices resulting from using BiLSTM-a on structured and unstructured v'_{exp} are presented in Fig. 15. As expected, the performance of BiLSTM-a on the structured v'_{exp} was better than the unstructured, however, both results still portray the incapability of BiLSTM-a to effectively distinguishing between impact motions as earlier displayed by the simulated results listed in Tables 4, 5 and 7.

Figure 15: Results of BiLSTM-a on (a) unstructured and (b) structured experimental data, v'_{exp}.

5.2. Improvement to impact motion classification

As a way of improving the use of acceleration measurements for impact motion classification, a feature extraction procedure was carried out on the acceleration time series. In each of the time series, peak points depicting points where impacts occurred along the time series were selected and represented as ones. Points marking the end of each cycle of oscillation were also selected and represented as zeros along the time series. Other data points which are neither points of impacts or end of periods were disposed from the time series, resulting into a reduced sample of data made up of ones and zeros. To do this, the acceleration time series was first differentiated and the peak function available in MATLAB was used to highlight impact points along the series. The end of each period of oscillation was defined and annotated based on the number of samples per period. Fig. 16 is an example of differentiated v' and v'_{exp} showing impacting points (red circles) and ends of periodic cycles (green asterisks). Differentiating the time history data helped project the negative-sided forward impacts into the positive side as well for the peak analysis and during the analysis, the peak function parameters must be selected to avoid picking any noise and bounce-backs.

Similar to the procedure carried out for the raw data, features extracted from v' were used in developing a feature based network (BiLSTM-a_{ft}) which was later used in classifying the features extracted from v'_{exp}. Table 9 and Fig. 17 shows the reconstruction performance of BiLSTM-a_{ft} across the simulated basins using different periods of data. Unlike others, the 30 periods data features showed consistent performance across the basins showing an overall accuracy of 98.1%, 97.3%,78.2%, 99.9% and 94.6% for BSN1, BSN2, BSN3, BSN4 and BSN5, respectively. This implies that even in their feature format, sufficient periods of data is still required to develop a consistent and effective network. In addition, the performance of the network on the out-of-sample basins followed the order of their stability with BSN4 > BSN2 > BSN5 > BSN3. The 30 periods BiLSTM-a_{ft} was used for classifying the experimental data of hammer's acceleration, v'_{exp}, and the results are presented in Fig. 18. The network showed 100% accuracy for both the P-1-1 and P-1-2 experimental motions. Aside from the experimental data being fewer, this
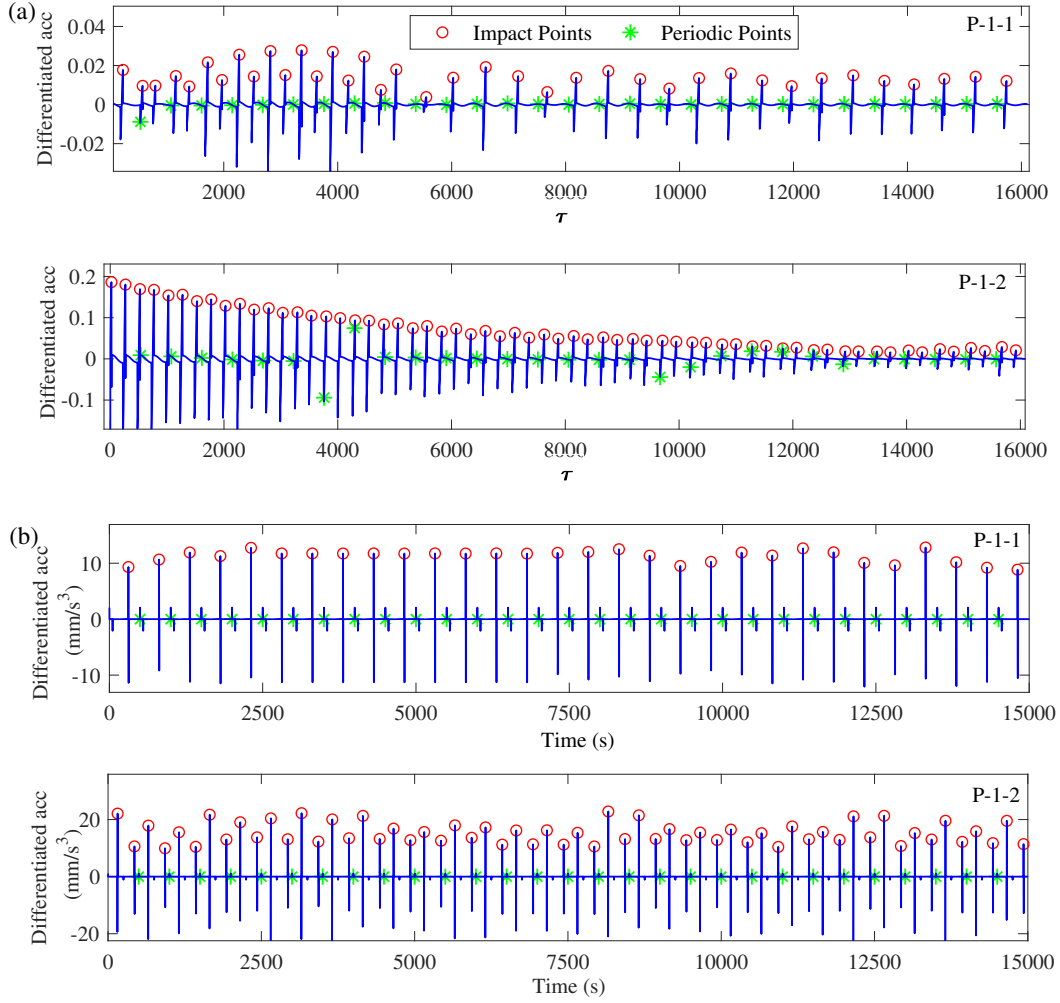
Figure 16: Plots of differentiated (a) v' and (b) v'_{exp} showing impacting points (red circles) and periodic ends ((green asterisks).

exceptional performance may be due to the fact that the dynamics of the rig are more stable compared to those of the impact oscillator (see Fig. 3 and Fig. 14) and the networks have repeatedly showed better performance on more stable systems. Another factor of consideration relating to the accuracy of a feature-based network is the expertise of the researcher in defining and picking useful features throughout the data samples as any deficiency in these respects will lower the performance of the network.

Table 9: Overall performance of BiLSTM-a_{ft} on different basins using various periods of the simulated acceleration.

Basins	15 Periods	20 Periods	25 Periods	30 Periods
BSN1	88.3%	95.2%	96.7%	98.1%
BSN2	91.0%	87.0%	83.9%	97.3%
BSN3	53.2%	65.3%	73.0%	78.2%
BSN4	84.9%	75.2%	80.0%	99.9%
BSN5	84.9%	75.9%	84.5%	94.6%

22
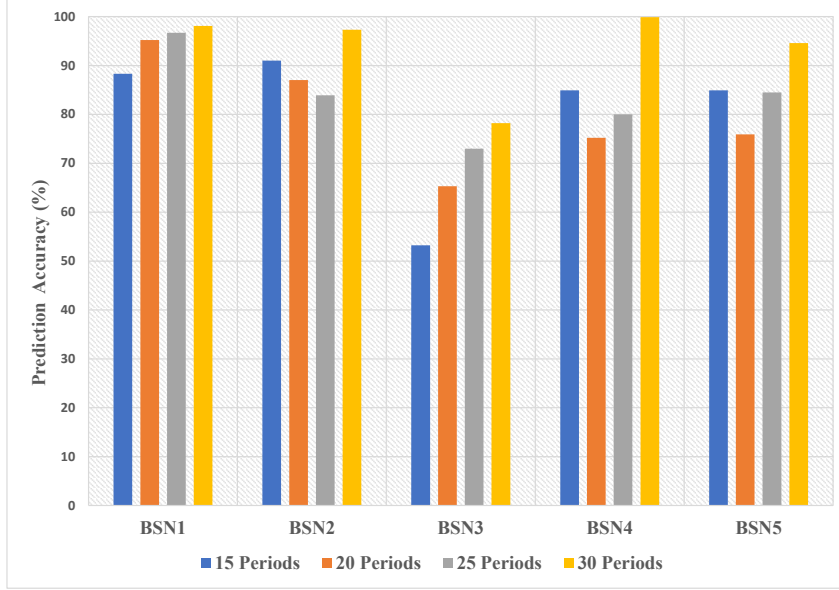
Figure 17: Feature based prediction accuracy using different periods of data samples.

Figure 18: Performance of the feature based network, BiLSTM-a_{ft}, based on the experimental data, v'_{exp}.

6. Conclusions and future work

For a continuously impacting system, such as the percussive drilling, where two or more impact modes can co-exist for the same set of parameters, real-time impact characterisation is very important for keeping the systems out of low performing and undesirable impact modes. In this study,the ability of bi-directional Long Short-Term Memory (BiLSTM) networks to self-learn impact dependent features from selected single or concurrent variables has been shown. Due to the limitation of measuring dynamical variables in real-life systems, a feature extraction approach has been suggested and utilised on readily measurable acceleration time histories.

Five simulated basins of co-existing impact oscillator motions and experimental data from a proof-of-concept experimental rig for percussive drilling have been used to demonstrate the effectiveness of this method. Results from using the simulated data showed that the BiLSTM-v, BiLSTM-xv and BiLSTM-xvep models are of exceptional performance compared to others even in the presence of noise-added data and no pre-processing. The slight difference between the numerical results obtained for using BiLSTM-v

and BiLSTM-xv show that extracting more than one useful variable can contribute to the performance of the networks. The results have also shown that using more periods of data can contribute to the optimal characterisation of the impact motions, especially those with longer periods to stability.

The proposed feature extraction technique increased the performances of the acceleration based network from 47.8%, 52.7%, 65.7% and 30.6% (see Table 7) to 97.3%, 78.2%, 99.9% and 94.6% (see Table 9), for the simulated data from BSN2, BSN3, BSN4 and BSN5, respectively. Its performance on experimental data increased from 61.9% to 100%.

The results so far obtained in this study show the potential of this method to distinguish between two co-existing impact motions from their transient dynamical variables with minimum expert interaction, thus making it deployable as part of an online control system for the autonomous percussive drilling system. To the best of the authors' knowledge, this is the first-time deep learning classification of vibro-impact motions is being carried out as the first step to the optimisation of percussive drilling.

Future work will involve using the proposed methods on multi-sourced and multifarious impact motions using readily measurable acceleration data. More feature extractions will be carried out to enhance accurate complex nonlinear pattern recognition from impact motion data, while also exploring the potential of shallow standard neural networks and automatic feature extractions which are devoid of human errors.

Acknowledgements

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

[2] A. Batako, V. Babitsky, and N. Halliwell, "A self-excited system for percussive-rotary drilling," *Journal of Sound and Vibration*, vol. 259, no. 1, pp. 97–118, 2003.

[3] 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.

[4] M. Wiercigroch, "Resonance enhanced drilling: method and apparatus," June 4 2013. US Patent 8,453,761.

[5] S. Butt, B. Gillis, and P. S. Rana, "Vibration assisted rotary drilling (vard) tool," Feb. 26 2019. US Patent 10,214,972.

[6] M. Gerlero, T. Soroldoni, D. Barreto, *et al.*, "San francisco basin tight reservoir play: Defining benefits of hammer bit/percussion drilling, onshore brazil," in *IADC/SPE Drilling Conference and Exhibition*, Society of Petroleum Engineers, 2014.

[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] 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.

[9] R. Aguiar and H. Weber, "Impact force magnitude analysis of an impact pendulum suspended in a vibrating structure," *Shock and Vibration*, vol. 19, no. 6, pp. 1359–1372, 2012.

[10] M. Liao, J. Ing, J. P. Chávez, and M. Wiercigroch, "Bifurcation techniques for stiffness identification of an impact oscillator," *Communications in Nonlinear Science and Numerical Simulation*, vol. 41, pp. 19–31, 2016.

[11] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv:1211.3711*, 2012.

[12] Y. Liu and M. Wiercigroch, "Rock type identification by impact action," in *Proceedings Nonlinear dynamics and control of deep drilling systems (ICNDCDS 2012): 2nd international colloquium, Eindhoven, May 15-16, 2012*, p. 87, 2012.

[13] N. Hinrichs, M. Oestreich, and K. Popp, "Dynamics of oscillators with impact and friction," *Chaos, Solitons & Fractals*, vol. 8, no. 4, pp. 535–558, 1997.

[14] A. B. Nordmark, "Non-periodic motion caused by grazing incidence in an impact oscillator," *Journal of Sound and Vibration*, vol. 145, no. 2, pp. 279–297, 1991.

[15] L. Virgin and C. Begley, "Grazing bifurcations and basins of attraction in an impact-friction oscillator," *Physica D: Nonlinear Phenomena*, vol. 130, no. 1-2, pp. 43–57, 1999.

[16] 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.

[17] S. Banerjee, J. Ing, E. Pavlovskaia, M. Wiercigroch, and R. K. Reddy, "Invisible grazings and dangerous bifurcations in impacting systems: the problem of narrow-band chaos," *Physical Review E*, vol. 79, no. 3, p. 037201, 2009.

[18] 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.

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

[20] K. Funahashi and Y. Nakamura, "Approximation of dynamical systems by continuous time recurrent neural networks," *Neural networks*, vol. 6, no. 6, pp. 801–806, 1993.

[21] J. Namikawa and J. Tani, "Building recurrent neural networks to implement multiple attractor dynamics using the gradient descent method," *Advances in Artificial Neural Systems*, vol. 2009, pp. 1–11.

[22] X. Jin, J. Shao, X. Zhang, W. An, and R. Malekian, "Modeling of nonlinear system based on deep learning framework," *Nonlinear Dynamics*, vol. 84, no. 3, pp. 1327–1340, 2016.

[23] C. A. Bailer-Jones, D. J. MacKay, and P. J. Withers, "A recurrent neural network for modelling dynamical systems," *network: computation in neural systems*, vol. 9, no. 4, pp. 531–547, 1998.

[24] F. Tsung and G. W. Cottrell, "Phase-space learning," in *Advances in Neural Information Processing Systems*, pp. 481–488, 1995.

[25] Z. Wang, D. Xiao, F. Fang, R. Govindan, C. C. Pain, and Y. Guo, "Model identification of reduced order fluid dynamics systems using deep learning," *International Journal for Numerical Methods in Fluids*, vol. 86, no. 4, pp. 255–268, 2018.

[26] O. Ogunmolu, X. Gu, S. Jiang, and N. Gans, "Nonlinear systems identification using deep dynamic neural networks," *arXiv preprint arXiv:1610.01439*, 2016.

[27] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[28] K. Doya and S. Yoshizawa, "Memorizing oscillatory patterns in the analog neuron network," in *International Joint Conference on Neural Networks, Washington DC*, 1989.

[29] F. Tsung, *Modeling dynamical systems with recurrent neural networks*. PhD thesis, Citeseer, 1994.

[30] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[31] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[32] D. M. Wolpert and M. Kawato, "Multiple paired forward and inverse models for motor control," *Neural networks*, vol. 11, no. 7-8, pp. 1317–1329, 1998.

[33] J. Tani and S. Nolfi, "Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems," *Neural Networks*, vol. 12, no. 7-8, pp. 1131–1141, 1999.

[34] J. Tani and M. Ito, "Self-organization of behavioral primitives as multiple attractor dynamics: A robot experiment," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 33, no. 4, pp. 481–488, 2003.

[35] J. Tani, "Learning to generate articulated behavior through the bottom-up and the top-down interaction processes," *Neural networks*, vol. 16, no. 1, pp. 11–23, 2003.

[36] K. O. Afebu, Y. Liu, and E. Papatheou, "LSTM-based approach for predicting periodic motions of an impacting system via transient dynamics," in *the First International Nonlinear Dynamic Conference (NODYCON 2019)*, (Rome, Italy.).

[37] Y. Xue, J. Jiang, and L. Hong, "A LSTM based prediction model for nonlinear dynamical systems with chaotic itinerancy," *International Journal of Dynamics and Control*, 2020.

[38] B. Guo, Y. Liu, R. Birler, and S. Prasad, "Self-propelled capsule endoscopy for small-bowel examination: Proof-of-concept and model verification," *International Journal of Mechanical Sciences*, vol. 174, p. 105506, 2020.

[39] Y. Liu, J. Páez Chávez, B. Guo, and R. Birler, "Bifurcation analysis of a vibro-impact experimental rig with two-sided constraint," *Meccanica*, 2020.

[40] J. Yang, Z. Li, and S. Liu, "Control of hysteretic system using velocity and acceleration feedbacks," *Journal of engineering mechanics*, vol. 118, no. 11, pp. 2227–2245, 1992.

[41] A. Preumont, *Vibration control of active structures*, vol. 2. Springer, 1997.