



First-order Layer in Artificial Pain Pathway

Oghuz Bektash¹ · Anders la Cour-Harbo²

Accepted: 7 May 2022 / Published online: 6 June 2022
© The Author(s) 2022

Abstract

The neural mechanisms involved in pain perception consist of a pathway which carry signals from the periphery to the cerebral cortex. First-order pain neurons transduce the potentially damaging stimuli detected by the sensorial extremes into long-ranging electrical signals that are transmitted to higher order neurons where the organisation is more heterarchical, especially in the cerebral cortex. However, the first order neurones, as their name states, have a degree of branching which clearly identifies them as hierarchical elements in the arrangement of pain pathway. This research aims to develop an artificial neural pain pathway that mimics this biological process, in particular the first order neurones. First, the research proposes the periodogram method on the condition monitoring data with a minor malfunction and operational damage. As the pain is associated with actual or potential tissue damage, using such data from a machinery system can provide insights which can be used to improve the computational effectiveness. Then, a one-dimensional convolutional neural network model is introduced to represent the second and third orders of the pain pathway. The research findings found clear support for studying the similarities between the major components of biological information processing of tissue damage and statistical signal processing for damage estimation.

Keyword Artificial pain pathway · First-order pain neurons · Damage prognostics and diagnostics · Convolutional neural networks · Periodogram method

1 Introduction

Research on artificial neural networks (ANN) has become one of the most significant development in the machine learning field of modern digital age. Structure of an ANN is often reported to be somewhat analogous to the biological neural networks that constitute a multi-layer mammalian brain. Admittedly, the analogy is vague and biological systems are substantially

✉ Oghuz Bektash
oguz.bektas@medeniyet.edu.tr

Anders la Cour-Harbo
alc@es.aau.dk

¹ Faculty of Engineering and Natural Sciences, Istanbul Medeniyet University, Istanbul, Turkey

² Automation and Control, Department of Electronic Systems, Aalborg University, Aalborg, Denmark

more complex than or current understanding. However, the transferring overall contents of neurons to computational process has provided a number of distinguished models -with record-shattering success- from recurrent neural network variants [1–5], to convolutional neural network-based deep learning classes [6–10], and the alternative physical energy based model translations for use in cognitive science [11–15], along with the unsupervised auto-encoders [16–20].

Another rapidly emerging field with a similar trend to ANN is the system health management by use of historical data, current conditions and future states [21]. For most contemporary complex systems embedded with peripheral sensors, deploying a machine learning model is certainly a stringent task. This might be even more challenging in systems where the authorities impose meticulous regulations for the sake of safe and secure practices [22–24].

Bio-inspired computing algorithms processing condition monitoring data can overcome this issue with a new ANN design that is comparable in certain respects to the evolution of the brain and its robust systems to handle the damage in the body. In this respect, it is important to comprehend the damage and its nature as well as how to adapt it in computing algorithms.

According to the International Association for the Study of Pain (IASP) terminology, the pain is *"an unpleasant sensory and emotional experience associated with, or resembling that associated with, actual or potential tissue damage."* [27, 28]. Based on this partial similarity, one can hypothesise that the damage (or operational anomaly) in the systems is analogous to pain phenomenon in the neural system. As the knowledge underlying these choices is mostly absent, this research aims to alter available neural network classes by introducing a novel first-order transformation layer to represent the pain pathways. The goal is to enable the systems to have specific cognitive abilities and coordination mechanisms of pain processing in a brain-inspired manner, and finally achieve a successful damage regulation level that is analogous to biological pain. Because the existence of hierarchy on "the degree of branching" on first order neurons and heterarchy across the following complex brain parts, this research chose to examine whether the conventional network structures based on a heterarchical collection of connected nodes transmitting a signal to each other, can be fed by a feature computing method rather than forming probability-weighted associations between different units. So that, such a method can convert a signal from its original domain (such as time) to a desired feature representation. Accordingly, this study proposes a one dimensional convolutional neural network model to represent the higher orders of the pain pathway and the Welch's method (also known as the periodogram method) to transform a sequence of time samples of a signal by estimating the power spectral density of the same signal.

The rest of the paper is structured as follows: First, the study reviews the extant scientific background and related works on the pain pathways, convolutional neural networks and the periodogram method. Then, a definition of the methods and techniques used in the article is given. A case study follows the methodology section and conducts an in-depth exploration of the success of the first order transformation layer within the operational damage context. The results of the case study are then discussed. In the final part, the article concludes with a discussion of implications and potentials for further research.

2 Background and Related Work

2.1 General Pathways of Pain Sensation

Pain is an unpleasant somatic sensation that is associated with tissue damage and it is a physiologically critical function for survival and evolutionary fitness in sustainable habitats.

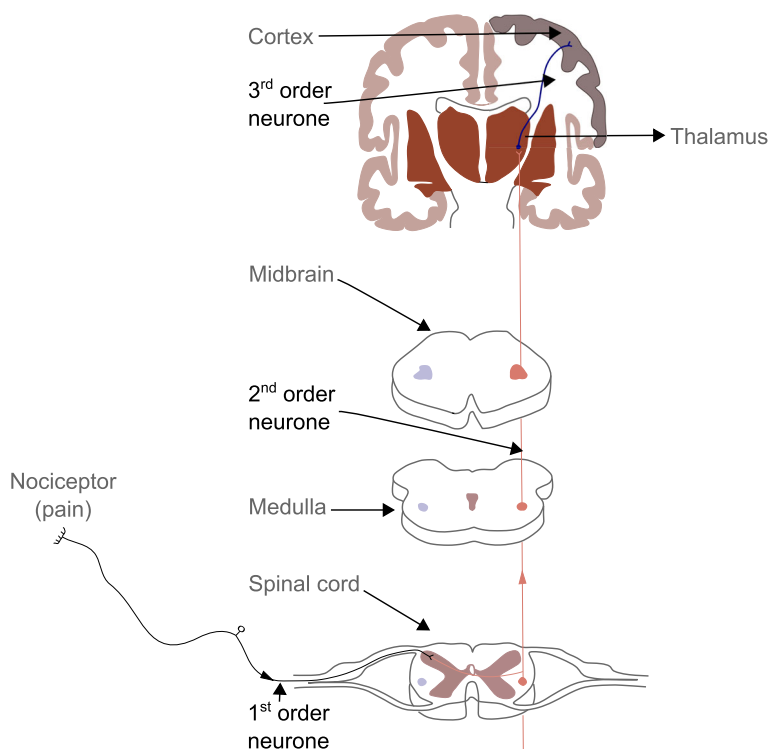


Fig. 1 Simplified depiction of three sensory neurones carrying pain information through ascending signaling pathway (adapted from [25, 26])

Within the general somatosensory warning mechanism, there are major known components of the ascending pain pathway designed to alarm against real or upcoming hazardous stimuli with the potential for tissue damage [25, 26, 29–34]. First-order pain afferents enter the spinal cord via the dorsal root [29, 34]. While terminating superficially at this location, the first order neuron synapses with the second-order neuron after which the may ascend signal to higher centres that relay the information to cortical centres [29, 30, 34]. Figure 1 gives a underlying diagram of the ascending pain pathway, with the first-order neurons from periphery to dorsal horn and second-order neurons from dorsal horn crossing over in the spinal cord into the thalamus and finally the third-order neurons synapse to thalamus and project to frontal cortex [25, 26, 30].

2.2 Modelling the Pain Sensation

By the very nature and architecture of this pain pathway depiction, the model in this paper simulates the way the brain processes the damage signals in the depicted neural orders. The analogy here is more like a shared abstraction and the corresponding elements do not share only a loosely relation, but also an inspiration. Displaying a resemblance to the activity of first order neurons which respond to noxious stimulation and send incoming signals up the spinal cord [29], the framework will initiate with an approach using Fast Fourier Transform (FFT) for the power spectra estimation, introduced by [35]. The spectral density characterises

the frequency of a signal, and it can represent and convert the properties and features of the same signal. A series of recent studies has indicated that spectral analysis can play a vital role for a better understanding of damage characterisation [36–38].

When the condition monitoring case is related with vibration or any rotating equipment (which will be in the following case study section), it would be difficult to picture the analysis without the FFT, the use of which can be found in almost all spectrum methods such as the Welch's method [36]. That being the case, the research will use the Fast Fourier Transform for power spectra estimation to represent the transformation of pain signal from nociceptors to the spinal cord. Second order neurons resembles the layer by layer structures of the convolutional and sub-sampling (also known as pooling) layers, the introduction of which propelled the field of Deep Learning. Therefore, the research chose to further advance the information from first order layer with a Convolutional Neural Network (CNN) and examine whether these frequency features can provide a strategic advantage over the understanding of damage.

CNN, proposed by [6], have repeatedly proven its success in various deep learning applications, including the damage related topics of fault detection, diagnostics and prognostics along with the systems health management [39]. The work of [40] presented a vibration-based structural damage detection system using 1D CNN. The model could automatically extract damage-sensitive features from the acceleration signals measured under random excitation, and output a binary classification label (damaged and undamaged). Likewise, vibration and acoustic data have a major role in damage for rotating machinery and has become focal point in the field of prognostics and health management (PHM) [39]. Current strategies have been used in various domains.

Ince et al. [41] implemented an adaptive 1-D CNN into an early motor fault-detection system with an inherent adaptive design fusing the feature extraction and classification phase into a single learning body. To autonomously extract useful features for bearing fault detection, [42] used a CNN based fault detection system as an end-to-end machine learning model. One important part of the damage degradation is to construct an effective health indicator that can expose quantifiable characteristics of system deterioration. [43] proposed a convolution feature learning based model to construct such indicators in bearing systems. [44] extended the use of CNN to wind turbine monitoring applications by processing complex vibration signal inputs from both rotor and gear box bearings. Their results demonstrate that the deep learning systems can outperform both the so-called shallow-learning models and human experts.

With an alternative design, [45] processed raw vibration signals through Deep CNN with wide first-layer kernels. These raw data are processed by the wide kernels in the first convolutional layer which extract the features and suppress the high frequency noise. In a similar manner, [46] proposed a hybrid model based on the continuous wavelet transform and CNN. The wavelets constructed the time-frequency gray scale images to be used in as the input of CNN. The trained function could extract the features from the fault signals and detect the faulty feeder at the same time. In an another hybrid model, a fusion layer was annexed to integrate a multi-layer perceptron with CNN [47]. The features were extracted from time-domain statistical features by using the annexed feed-forward ANN and combined with the local vibration signal feature from the CNN so that they can further identify different faulty states.

Extending the fusion concept to parallel computing of deep learning models, [48] employed a long short-term memory neural network (LSTM) for temporal feature extraction, and also simultaneously use CNN for spatial features. Then, a fusion centre combines these two paths to estimate the remaining useful life time of aero-propulsion systems. Overall, these studies demonstrates the significant performance of CNN on the damage detection,

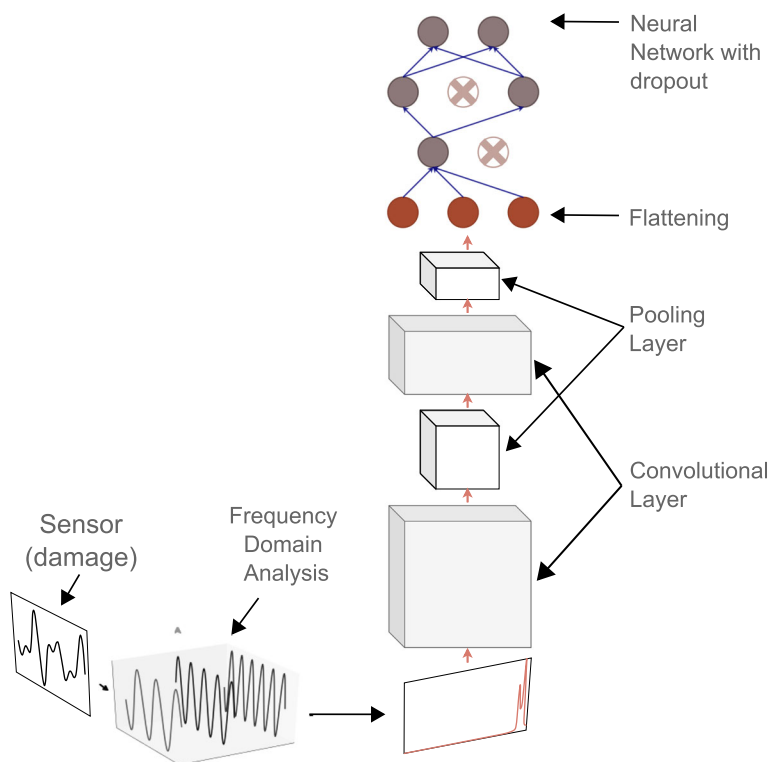


Fig. 2 Artificial Pain Pathway represented by three orders of layers, namely: the periodogram method, convolutional layers and fully connected layer

particularly when fused with an accompanying approach. In accordance with the success of the hybrid approaches, this research will fuse the periodogram method with the convolutional layers that will represent the second order neurons. the outputs will be converted (or flattened) into a 1-dimensional array for inputting them to the last layer. The cell bodies of third order neurons lie within a region on the primary cortex. Here is where convolutional layers and a classical feed forward network will collide as the latter is included in the form of a “Fully Connected” layer. The specific region will be represented as a dropout layer. The term refers to randomly “dropping out” units, both hidden and visible ones along with their connections, from the neural network during training process [49, 50] so that over-fitting in training will be avoided. This theoretical foundation is a simplistic modelling of an artificial pain pathway (APP), and it is shown in Fig. 2.

3 Methodology

This section provides information on the proposed periodogram method and CNN procedures for the application of damage diagnosis. The framework is a series of three layers which helps extracting the anomaly related features and respond to the final feed forward network for classification. The details of these layers are defined in the following parts.

3.1 Welch's Method

The power spectral, or power spectrum, estimation of a given signal is one of the most well-known practices in digital signal processing [51]. It is the measure of a signal's power content falling within given frequency bins, and represented by $P_{xx}(f)$.

A periodogram is depicted by such a power spectrum of a finite time series as:

$$x_0, \dots, x_{N-1} \quad (1)$$

by using the finite time series, the periodogram can be composed by the method of [52], with the following formulation [53]:

$$P_{xx}(f) = \frac{1}{N} |X(f)|^2 \quad (2)$$

where, the Fourier transform of a sample sequence, $X(f)$, is defined as:

$$X(f) = \sum_{n=0}^{N-1} x(n) e^{-i\frac{2\pi}{N}fn}, \quad f = 0, \dots, N-1 \quad (3)$$

With reference to this, the time series can be transformed into a spectrum of frequencies.

$$P_{xx}(f) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x(n) e^{-i\frac{2\pi}{N}fn} \right|^2 \quad (4)$$

Welch's method is regarded an adaptation of this conventional periodogram method [35, 51] and, accordingly, often called as the periodogram method. The method is implemented by allocating the time series into sequential divisions (segments), using the periodogram formula for each segment, and calculating the average of them [54].

Let the segments' length be L and the starting points of each be D steps apart [35], first of the segments is given by

$$x_1(j) = x(j), \quad j \in \{0, 1, \dots, L-1\} \quad (5)$$

hereby, second of the segments is

$$x_2(j) = x(j + D), \quad j \in \{0, 1, \dots, L-1\} \quad (6)$$

so that the total number of data is

$$N = L + D(K-1) \quad (7)$$

where K corresponds to the total number of segments and the k -th segment is defined as

$$\begin{aligned} x_k(j) &= x(j + D(k-1)), \quad j = 0, \dots, L-1 \\ k &= 1, \dots, K. \end{aligned} \quad (8)$$

Figure 3 shows this segmentation of time series and illustrates the above mentioned parameters in the coordinate system.

Second modification of the Welch's method is to weight the segments by a window [35], formulated as

$$w(j), \quad j = 0, 1, \dots, L-1 \quad (9)$$

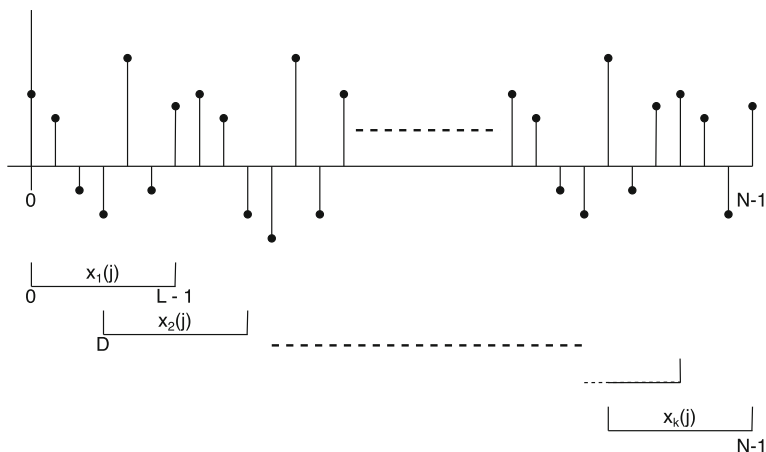


Fig. 3 Record segmentation in the Welch's method

Segments before computing the periodogram form the sequences of

$$\begin{aligned} x_k(j)w(j), \quad j = 0, 1, \dots, L-1 \\ k = 1, \dots, K. \end{aligned} \quad (10)$$

When these sequences are applied into the periodogram formula, the result then becomes a “modified” periodogram, $\tilde{P}_{xx}(f)$.

$$\tilde{P}_{xx}^{(k)}(f) = \frac{1}{LU} \left| \sum_{j=0}^{L-1} x_k(j)w(j)e^{-i\frac{2\pi}{N}fn} \right|^2 \quad (11)$$

where U stands for a normalisation factor for the power in the window function [53] and it is defined as

$$U = \frac{1}{L} \sum_{j=0}^{L-1} w^2(j) \quad (12)$$

In the final step of the Welch estimate, $P_{xx}^W(f)$, the modified periodograms, $\tilde{P}_{xx}(f)$, are averaged.

$$P_{xx}^W(f) = \frac{1}{K} \sum_{k=1}^K \tilde{P}_{xx}^{(k)}(f) \quad (13)$$

After applying these spectral density estimation steps, the data are transformed into a new form that will be used for the first layer of CNN. This is also known as the input layer and formed by artificial input neurons that bring these transformed data into the network structure for the subsequent layers.

3.2 Convolution

A convolution operation is the first building block in a CNN architecture. The feature detectors here serves as the filters. Proposed framework employs a 1D layer that forms a convolution

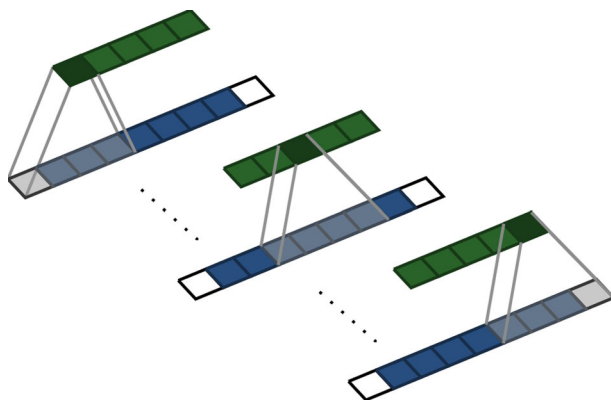


Fig. 4 Visual demonstration of convolution operation: blue grid is the input map, kernel is the glared area and green grid is the convoluted output

kernel which will be convolved with the corresponding input layer over a single dimension (see Fig. 4).

This is a slide over a spatial (or sometimes temporal) dimension to produce a tensor of outputs. Sliding allows a element wise multiplication with the corresponding input section, and then summing up the matrix into a single production (Fig. 4, dark green square) which will be placed in an another 1d feature vector (the green grid).

For mathematical definition, let's call the input vector -the one arrived from the periodogram method- f and the kernel g . The convolution of f and g is defined as

$$(f * g)(i) = \sum_j g(j)f(i - j) \quad (14)$$

where the result vector index is given by i .

3.3 Non Linearity - ReLU Layer

An additional activation function Rectified Linear Unit is applied an to the convolutional output. Rectified Linear Unit (or shortly ReLU) is an essential unit of CNN with an outperforming performance comparing to other activation functions and a wide recognition in the literature [55]. It is a non-linear operation and its output is given as

$$f(x) = x^+ = \max(0, x), \quad (15)$$

This is applied piece-wise by replacing all negative ones in the feature map by zero.

3.4 Sub-sampling - Pooling

The following part of CNN is max-pooling, a form of down-sampling (also called sub-sampling). It partitions the convoluted data into a set of non-overlapping sub-regions and outputs the maximum from each. So it reduces the dimensionality of representation while retaining the most useful information. Down-sampling decrease the number of features and network computation, and therefore it can reduce the optimisation complexity and control

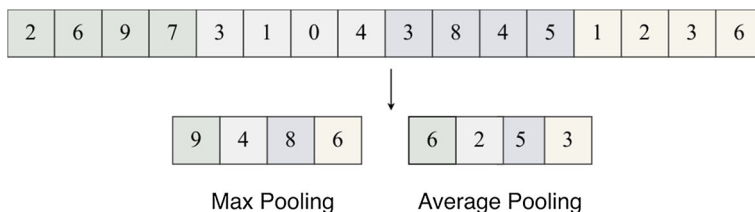


Fig. 5 Max and average pooling with a 1D filter

over-fitting. Figure 5 illustrates the Max-Pooling (and also Average pooling) with a 1×4 window. Each operation is over 4 numbers and discards 75% of the content.

After including the max-pooling layer, the artificial pain pathway takes the following form becomes ready for third order neurons.

$$\text{First-order} \Rightarrow \text{Conv} \Rightarrow \text{Act} \Rightarrow \text{Pool} \quad (16)$$

Simulating the primary thalamus function of relaying signals to the cerebral cortex, the output of the preceding pooling layer is flattened into a single column to be inserted into an ANN classification layer in the following step.

3.5 Fully Connected Layer- ANN Classification

After previous layers and flattening, the high-level classification in the model is done via fully connected layers in which the neurons connect to all outputs of the prior layer and also to following ones on the next layer, as seen in regular (non-convolutional) ANNs.

A basic single-input neuron is defined by input (x), weight (w), bias (b) [56]. To find its output, the arriving input signals are adjusted by the connection weights, then summed and and a bias term is added. The output is passed through a transfer function (f) to produce results. These are expressed as

$$f(\text{net}), \quad \text{where } \text{net} = \sum_{i=1}^n w_i x_i + b \quad (17)$$

In a fully connected neural network layer, there are multiple of these units organised in multiple layers, that can simply associate input elements to an output, while in aggregate do complex computations (see Fig. 6).

3.5.1 Dropout

Over-fitting is a critical problem in ANN training due to either large number of parameters or complex co-adaptations unable to generalise the data. An over-fitted model corresponds too closely to input data, learns the detail and noise, and accordingly fail to fit additional samples accurately. Applying a dropout layer to the input of previous one is an efficient way of addressing this problem [49]. The term refers to randomly "dropping out" neurons - along with their connections—during training process [49, 50]. In Fig. 7, there is an illustration of how the neurons, and their incoming and outgoing links are temporarily removed from the training process. Here, an architecture randomly sets the neuron input units to 0 with a frequency of rate at each step while the rest are scaled up by $1/(1 - \text{rate})$.

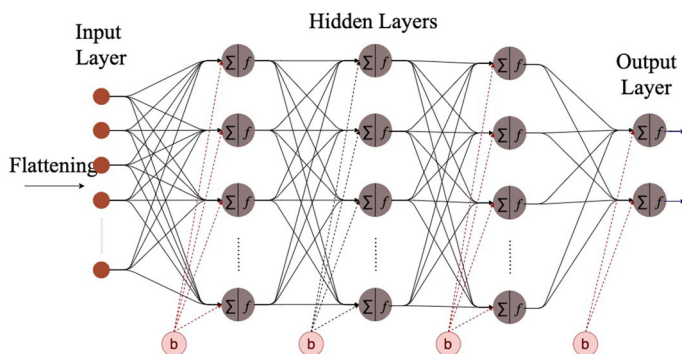


Fig. 6 Multi-layer fully-connected artificial neural nets

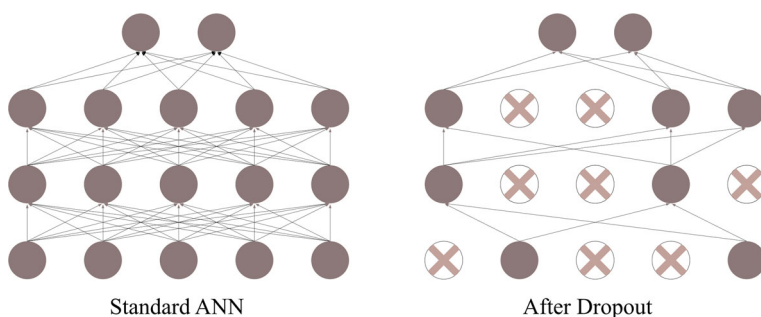


Fig. 7 A sample of dropout

3.6 Sigmoid Function

Sigmoid activation function is applied in the output layer. Having a characteristic S-shaped sigmoid curve, it always provides a return value between 0 and 1, and defined by the formula of

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}}. \quad (18)$$

The final structure of the framework is visualised in Fig. 8 where the methodology steps are carried out according to the order. This representation follows the proposed *Artificial Pain Pathway* workflow: start of the processing by segmenting the vibration data into smaller segments, input layer formation from the segmented data, first-order transformation layer, feature learning at convolutional layers, and fully-connected layer for classification.

4 Testing and Results

4.1 Flight Data

The methodology is tested on vibration data monitored by sensors mounted on a T50 rotorcraft from Aveox—a turbine-powered unmanned aircraft with dual rotors. The data set was recorded on September 17, 2019, at a small airfield in Denmark. There were four flights

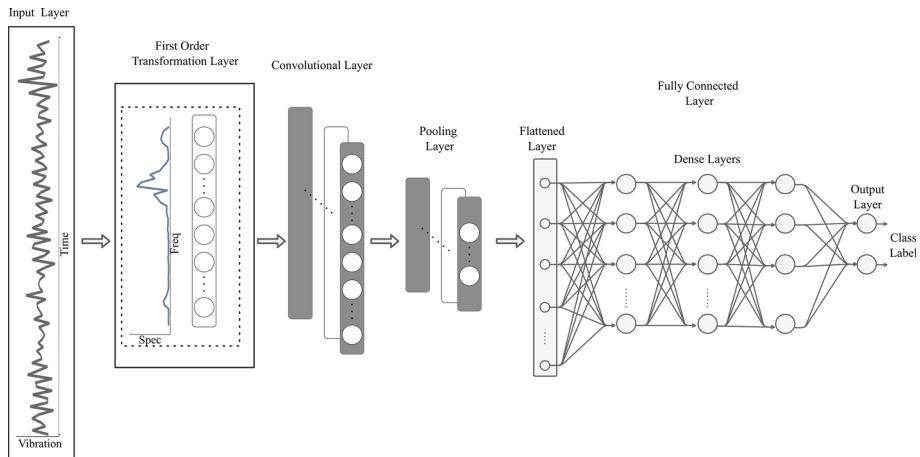


Fig. 8 Artificial pain pathway showing key sites for damage processing

conducted over about 2 hours time, each lasting around 10 minutes and standing mostly in hover. The overall goal of this testing was to measure the vibration level in various operational conditions such as flight with no payload or with full payload, as well as in strong wind. Testing was conducted in varying wind speed between 5 and 20 m/s over the flight time and also with altitude.

From previous operations, the flight team observed that, when fully loaded, the aircraft had severe vibrations—beyond the acceptable level. With this anomaly in mind, one of the flights was presented with high payload to understand how severe this issue was. After consulting the manufacturer, the problem was mostly remedied by a change to the rotor blade and rotor hub configuration. That being the case, the vibration data from this flight indicates a minor malfunction (or damage) that might have let into a more serious issue of structural degradation and perhaps ultimate disintegration.

All test flights started with the same take-off position and reached to an altitude around 25 meters, with the exception of the last flight which was in higher altitude to increase the wind speed exerted on the aircraft. Detailed between between the flights are as follows:

1. Initial flight conducted to establish a baseline. No payload and altitude around 25 m.
2. High payload. 35 kg of steel plates were fixed to the bottom of the fuselage. Altitude around 25 m.
3. High payload removed—a hook installed to attach a 5 kg payload in a 4 meter long metal chain. Altitude around 25 m.
4. 5 kg payload unhooked—no other payload installed. This flight was hover at different altitudes of 25 m, 50 m, and 75 m. Both nose and right side faced into the wind.

To record the vibration data during these flights, a series of identical IMU sensors—SparkFun 9DoF Razor IMU—were fixed to the aircraft in locations as shown in Fig. 9. They were located in random orientations, since only the magnitude was of interest. Sensor recordings include angular velocity, translational acceleration and the magnetic field at a rate of 100 Hz.

During the test flights, the constantly changing wind speed and direction had an obvious effects on the aircraft which might have caused aerodynamic effects on the rotor and changed the vibration pattern. Flight videos can be accessed at the [UAS-ability YouTube channel](#) [57].

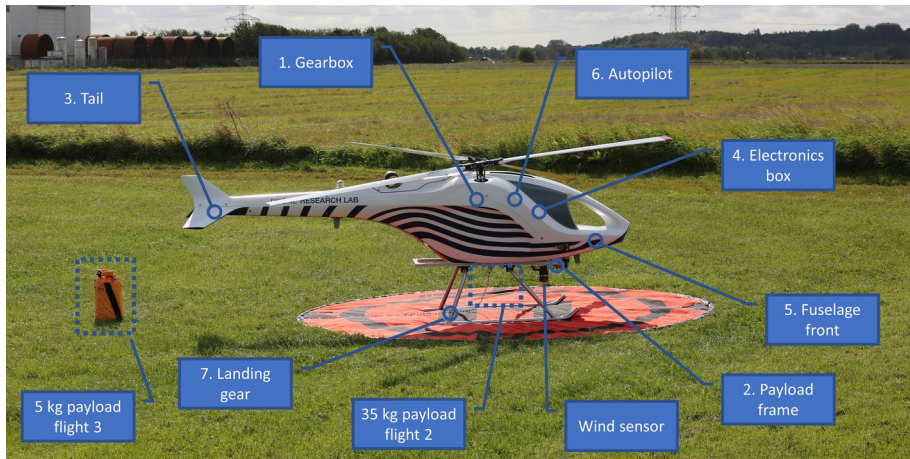


Fig. 9 The photo shows the T50 aircraft with the location of the seven IMU sensors, the wind sensors, and the payloads for flight 2 and 3. The payload for flight 2 is no mounted, but the location is shown

They are named with the date - i.e. "19.09.17" - and - FL00x - referring to the flight number. There were four camera views available; an overview camera, a manual tracking camera, an on-board camera, and a ground station camera (called MGCS). During the third flight, there were also an additional video of the aircraft taken by another drone.

4.2 Signal Pre-processing

Flight data were uploaded to a common repository and timestamped. These raw sensor data contain instrumental errors - some extreme measurements beyond the expected range and unlike the other readings. Once captured, the primary data need to be cleaned from these outliers for a better analysis and useful observations best illustrating the case.

To find the outlier points that differ significantly from the distribution, the data are normalised based on the mean and standard deviation. Any standardised point significantly higher than the mean is an outlier and replaced by using linear interpolation between two of its neighbour points.

Figure 10 provides raw data representation after dropping the outliers. Of the 7 IMUs on the drone, the one mounted directly onto the autopilot—called VECTOR in this aircraft - is the most representative sensor for what would be available on a normal unmanned aircraft. This is mainly because it is located in the ideal position where it can accurately detect anomalies or changes in the drone environment and provide the related information. On account of this, the data from this IMU is used in the model testing.

To apply the proposed framework, the data on Fig. 10 are segmented by dividing the recorded time series up and grouping sequentially into segments so that the APP can classify each segment into a proper label - "regular" or faulty operation. These segments are represented in Fig. 11 with a windows size of 100 for each. Both early and late predictions are removed to avoid the take off and landing data which include relatively less vibration than the rest of operation, and therefore can confuse the model. After the proposed periodogram method applied into each of them, their size are reduced and they could provide a distinguishing damage related features. Figure 12 gives an indication of how the power spectral

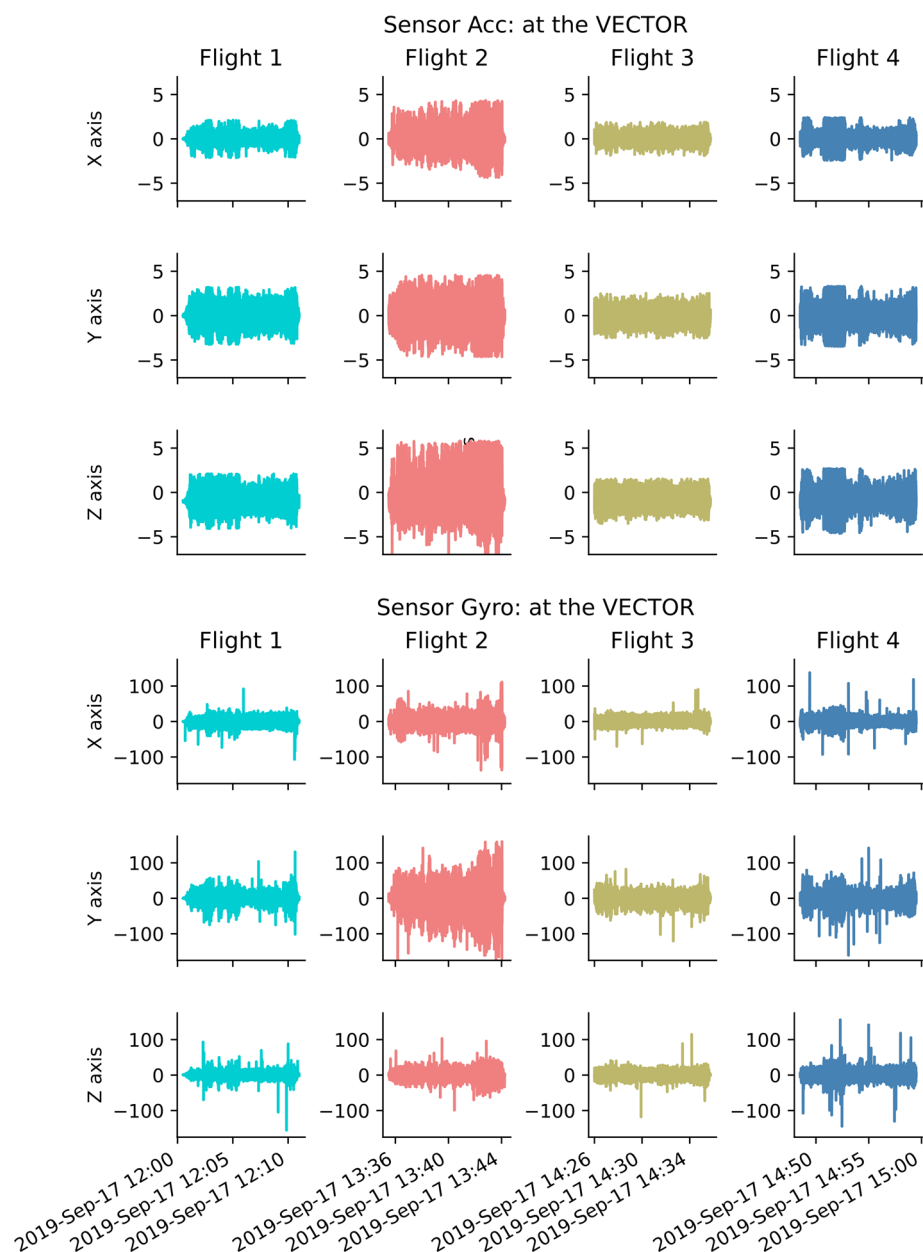


Fig. 10 Raw Flight Data (outliers removed)

density for each segment varies with frequency. Each spectral density estimation is estimated by dividing the cycles into overlapping parts, calculating a modified periodogram for each part and averaging them.

Even at first glance, it is obvious that the second flight comprise higher power spectrum values than the other three regular flights. The extracted features from these periodograms are potential indicators of the flight regularity or anomaly.

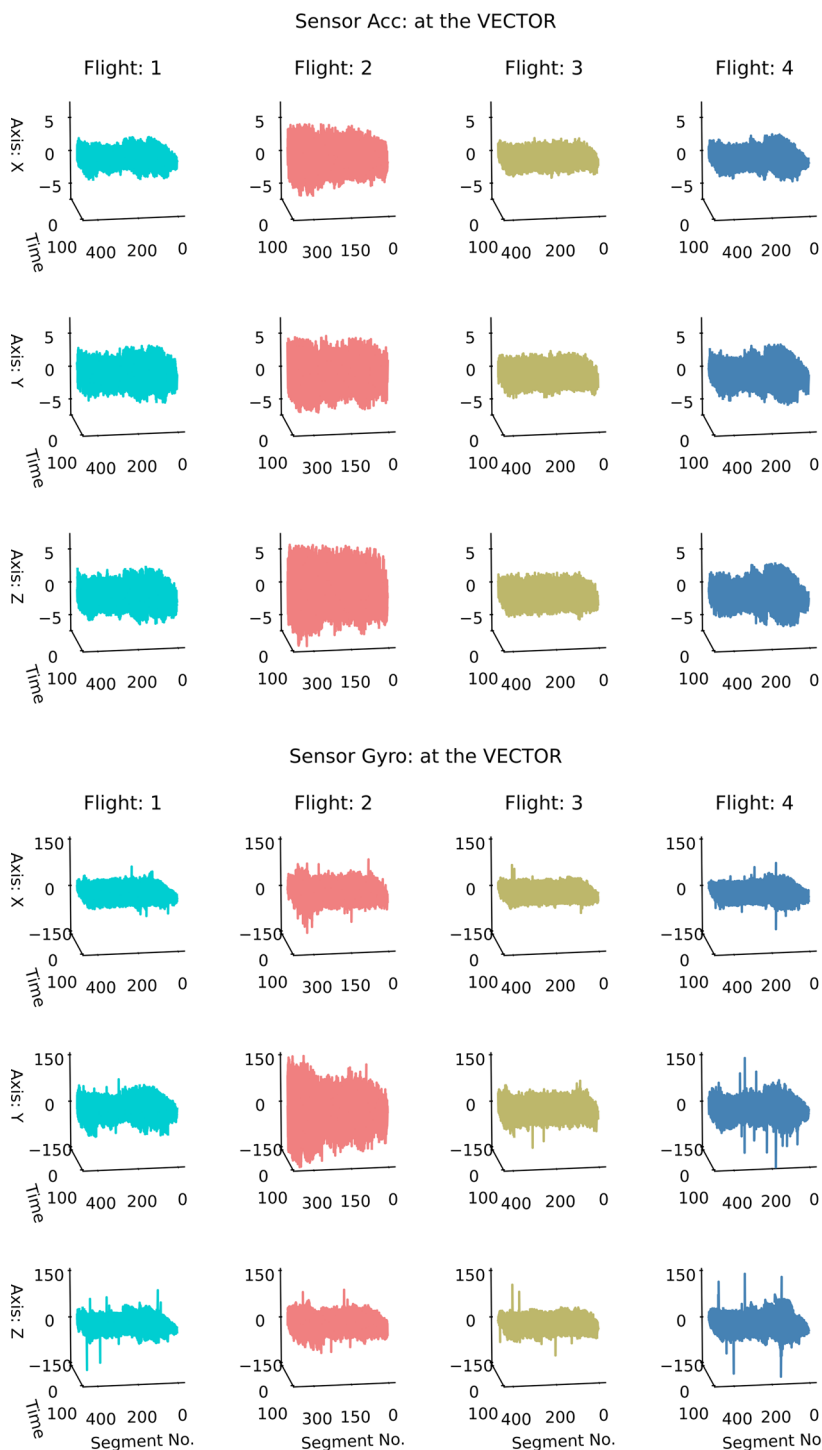


Fig. 11 Segmented vibration data

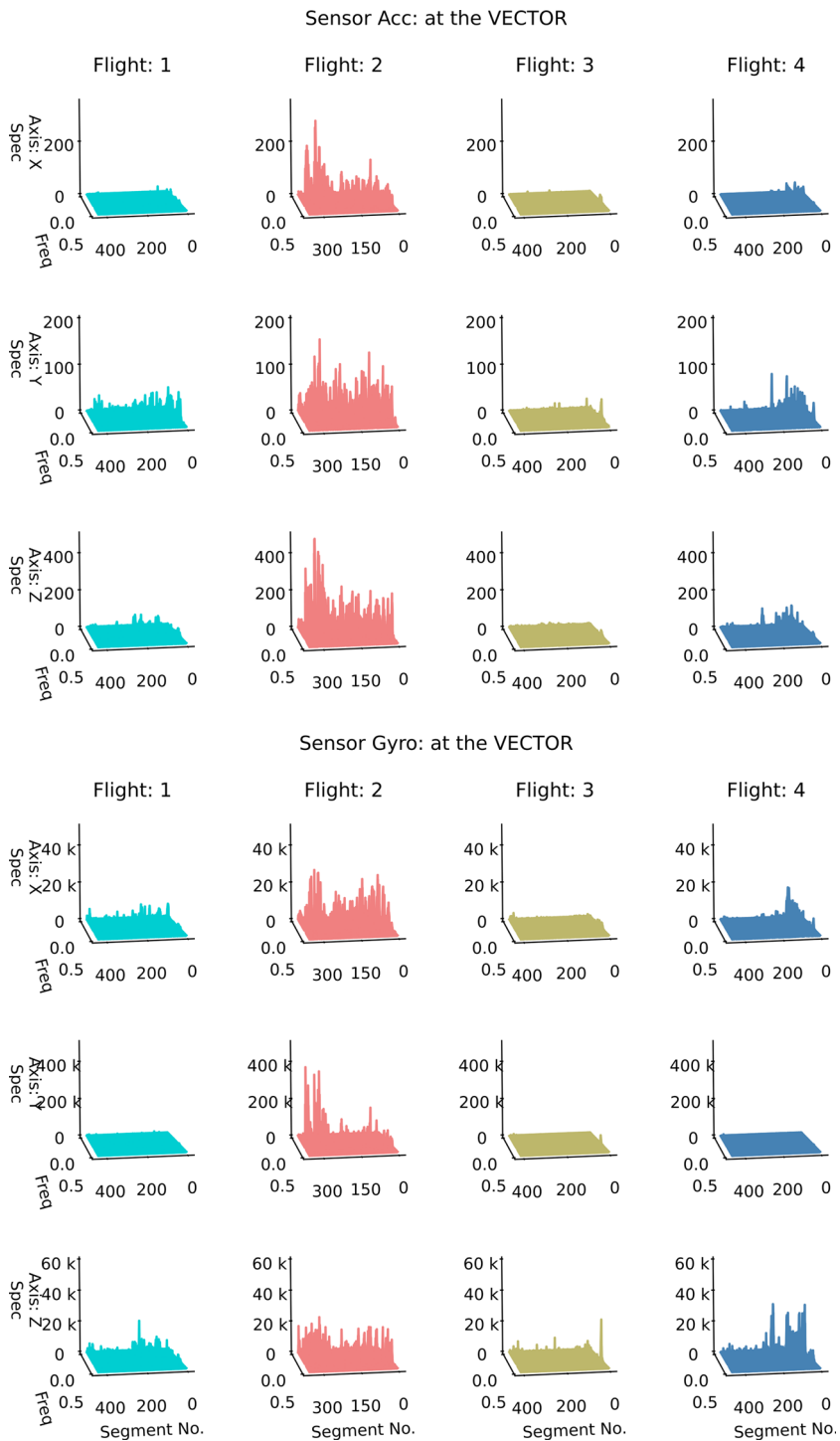


Fig. 12 Power spectra of the segmented data

When the data are transformed as like in Fig. 12, the results become the output of the first order transformation layer in the APP and will then be associated with a 1D CNN. The results indicate a relatively small difference in both the accelerometer and gyro values—except for the second flight which displays an anomaly in the operation as expected due to the extra payload. From this, it is possible to argue that Flight 2 has an appreciably more disorder than the others conducted by the same drone.

The findings may also hint a threshold level indicating boundaries of the acceptance region for each sensor reading. When there is a significant number of critical values surpassing the threshold level, it can be regarded that a flight anomaly have occurred.

However, some sensor readings during Flight 4 may surpass the threshold value—probably due to higher altitude in operation. A threshold model may not distinguish these from an flight anomaly - suggesting that some measurements in a certain altitude or a direction might fail to demonstrate overall performance. On the other hand, the other directions of the same sensor on Fig. 12 perform as expected. To have an efficient classification, CNN employs a process of taking all 6 sensor vectors as an input and outputting a probability that the segments fall in particular labels of "regular" and "anomaly" operation.

Table 1 gives details of the configuration, or architecture, which specifies what layers the framework contain, and in what order they are connected. There is the input layer in which 6 different vibration data are segmented and processed through to the transformation layer. Then, it outputs (51,6) values to the second order layer where the convolutional layer creates a kernel with a single point window length. To produce a tensor of outputs, the kernel is convoluted with the first order layer output over a single spatial dimension. There are 32 output filters in the convolution and 6-length input vectors with 51 time-steps. The following pooling layer down-samples the incoming output representation to (25,6) by taking the maximum value over the window of 2. Then, these data are flattened and processed by 3 regular densely-connected neural network layers. Finally, an output layer with the sigmoid function is responsible for producing the final classification.

With these steps, the model is compiled and fitted to the segmented vibration data. The number of epochs to train the model is set to 25 – after this number, the epochs do not affect directly the result of the training step. An epoch is a full iteration over samples, the entire input and output data provided. During the training process, the iterations are required to update the network's parameters so that the algorithm can reach an optimal point where the classification is accurate enough. In compiling, an optimiser is implemented with the Adam algorithm to optimise the parameters. This is a stochastic gradient-based optimisation method, based on adaptive estimation of first-order and second-order moments [58]. Adam optimisation allows straightforward implementation with little memory requirement and computational efficiency, and the method is also compatible with the complex cases that are large with regard to data and parameters [58]. With the purpose of computing the loss quantity that ANN seek to minimise during training, the built-in loss function computes "the mean squared error" between the known output labels and the model estimations. The metric function used to judge the model performance is "accuracy" which calculates how often the model estimations with actual output labels.

5 Results

The experimental findings in this section come from the model testing that evaluates the framework in the form of descriptive statistics. The section provides graphs and plots as well

Table 1 APP layer formation

APP order	Layer type	Size	
1st	Vibration - Input	Input:	–
		Output:	(100, 6)
	Transformation Layer (Welch's Method)	Input:	(100, 6)
		Output:	(51, 6)
2nd	Conv1D (Activation ReLU)	Input:	(51, 6)
		Output:	(51, 32)
	MaxPooling1D	Input:	(51, 32)
		Output:	(25, 32)
3rd	Flatten	Input:	(25, 32)
		Output:	(800)
	Dense_1 (Activation: ReLU)	Input:	(800)
		Output:	(100)
	Dropout_1	Input:	(100)
		Output:	(100)
	Dense_2 (Activation: ReLU)	Input:	(100)
		Output:	(100)
	Dropout_2	Input:	(100)
		Output:	(100)
	Dense_3 (Activation: ReLU)	Input:	(100)
		Output:	(100)
	Dropout_3	Input:	(100)
		Output:	(100)
	Dense_4 (Output Layer)	Input:	(100)
		Output:	(2)
	Activation: Sigmoid	Input:	(2)
		Output:	(2)

as the results of the model analysis. In light of the estimations and model performance, the section shows the model's ability to accurately perform classification task not only just with training data but also with validation data; so that the model can be actually deployed in real-time with run-time data. The outcomes are reported in detail so that one can justify the proposed configurations. The output labels takes the value [1, 0] for regular operations and [0, 1] for anomaly to convert the categorical classification into indicator variables.

The full data set is divided into two subsets: a set to train the model and set to test the trained model. The partition of full data set allows an unbiased evaluation of the final model fit on the training data. On the assumption that the test set meets the preceding training results, it can be concluded that the model generalises well to new data and the test set serves as a proxy for evaluating the loss and metrics at the end of each epoch.

In Fig. 13, the model accuracy for both subsets are plotted. An accuracy of $> 95\%$ is achieved for both and, that is to say, the model does not over-fit the training data. In both cases, the model could fit the parameters and produce improving results by iterations without corresponding too closely to the training date or making an overly complex focus on the



Fig. 13 Metrics of training and validation accuracy over epochs



Fig. 14 Loss functions. Results of the mean squared error between labels and classification estimations

idiosyncrasies. Using the dropout layers as presented in Table 1 is a major reason why the model could fit the test data efficiently.

There is also a similar trend in the training and validation loss over epochs, see Fig. 14. The minimal difference between the two lines and the same decreasing results proves that

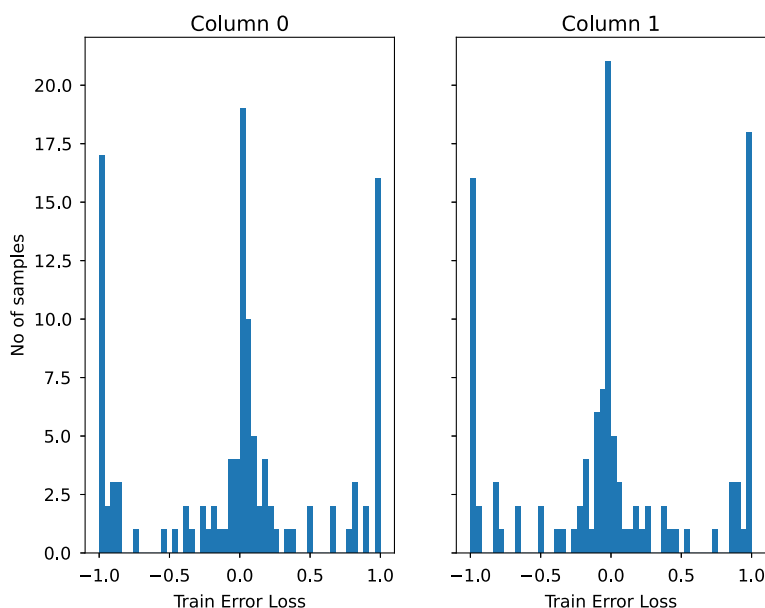


Fig. 15 Histogram of train error loss for both labels

Table 2 Classification report and confusion matrix

Classification report	Precision	Recall	f1-score	support
0.0 (Regular)	0.98	0.98	0.98	1320
1.0 (Faulty)	0.92	0.93	0.93	355
accuracy			0.97	1675
macro avg	0.95	0.95	0.95	1675
weighted avg	0.97	0.97	0.97	1675
Confusion matrix	Pred. 0.0	Pred. 1.0	Sum	
Actual 0.0 (Regular)	1293	27	1320	
Actual 1.0 (Faulty)	25	330	355	
Sum	1317	358		

the models is computationally inexpensive and has an effective regularisation process along with an improved generalisation. The low loss metrics, along with the high accuracy, are promising findings hinting that the proposed models can perform well at deciding on regular or anomaly conditions.

Figure 15 displays an approximate representation of the distribution of error ($|e| > 0.01$) for both label columns. In the histogram, the bar groups show that much of train error falls into the acceptable range between -0.5 and 0.5 . By returning the indices of the maximum values along each row of the estimated targets as returned by a classifier, the ones in the acceptable range are regarded as successful estimations. Comparing these returned indices with the corresponding second column of the ground truth (correct) output labels, a report showing the main classification metrics is built and the results are shown on Table 2.

Table 3 APP layer formation with the windows size of 200

APP order	Layer type	Size	
1st	Vibration-Input	Input:	–
		Output:	(200, 6)
	Transformation Layer (Welch's Method)	Input:	(200, 6)
		Output:	(101, 6)
2nd	Conv1D (Activation ReLU)	Input:	(101)
		Output:	(101, 32)
	MaxPooling1D	Input:	(101, 32)
		Output:	(50, 32)
3rd	Flatten	Input:	(50, 32)
		Output:	(1600)
	Dense_1 (Activation: ReLU)	Input:	(1600)
		Output:	(100)
	Dropout_1	Input:	(100)
		Output:	(100)
	Dense_2 (Activation: ReLU)	Input:	(100)
		Output:	(100)
	Dropout_2	Input:	(100)
		Output:	(100)
	Dense_3 (Activation: ReLU)	Input:	(100)
		Output:	(100)
	Dropout_3	Input:	(100)
		Output:	(100)
	Dense_4 (Output Layer)	Input:	(100)
		Output:	(2)
	Activation: Sigmoid	Input:	(2)
		Output:	(2)

rt includes precision, recall, F1 score for both labels and also the overall accuracy of the model. The reported averages include macro avg - averaging the unweighted mean per prediction class— and weighted avg -averaging the support-weighted mean per prediction class. Table 2 also provides a confusion matrix to evaluate the classification accuracy. In the binary classification estimation, the count of true negatives (regular) is 1293, false negatives is 27, true positives (faulty) is 330 and false positives is 25.

Even though the results are desirable, the data scarcity is a major bottleneck and the model performance relied heavily on the windows size used to segment data. A segment with more data can provide better features to process but this also reduces the number of available samples. After the segments are updated with a new windows size of 200, the network was re-configured as in Table 3, re-run with the same setup and the results are refreshed.

The results of the later training has provided slightly better results. Figure 16 illustrates the performance of classification. Both validation and accuracy rates confirms that more data for each label output can provide better results on classification. Additionally, the results on Fig. 17 lead to similar conclusion where there are fewer errors beyond the acceptable range (between -0.5 and 0.5). Even though slightly superior results are achieved, it is worth

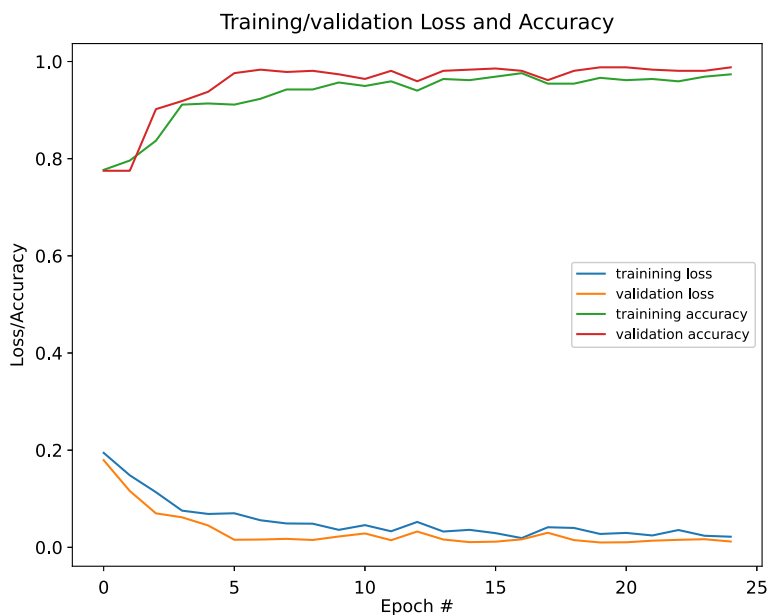


Fig. 16 Accuracy and loss metrics for the second training

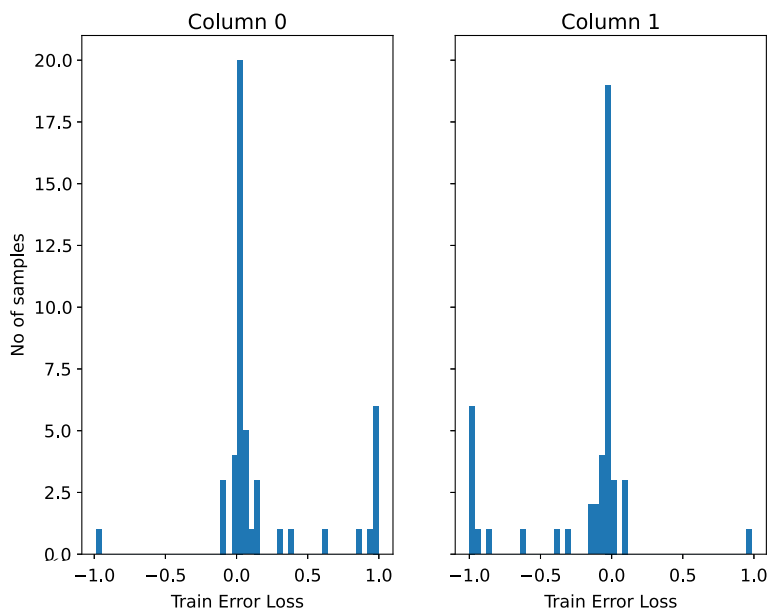


Fig. 17 Train error loss results of the second training

discussing that these interesting facts revealed by the results of higher windows size. These might even cast a new light on using more data from additional sensors or data types.

With a similar comparison between the indices of the maximum of the estimated targets and the ground truth label, a further classification report and a confusion matrix is given on

Table 4 Classification report and confusion matrix for the second training

Classification report	Precision	Recall	f1-score	support
0.0 (Regular)	0.99	1.00	0.99	658
1.0 (Faulty)	0.99	0.95	0.97	177
accuracy			0.99	835
macro avg	0.99	0.97	0.98	835
weighted avg	0.99	0.99	0.99	835
Confusion matrix	Pred. 0.0	Pred. 1.0	Sum	
Actual 0.0 (Regular)	657	1	658	
Actual 1.0 (Faulty)	9	168	177	
Sum	666	169		

on Table 4. Together, these results tie well with previous ones. When comparing the label accuracies, it must be pointed out that false positives are higher than false negatives even though there are more regular ones. This might be regarded a negligible bias or potential limitation but, in a real case scenario, it would be more important to correctly estimate and classify the regular operations without causing any undesired interruption.

6 Conclusion

This work proposed and demonstrated an artificial pain (damage) mechanism which process vibration signals from four different drone operations. The model employs the integration of the periodogram method and a 1D CNN with multiple dense layers. The trained model has the ability of extracting the fault related features, and also detecting the operational anomaly with high accuracy. The results clearly revealed that the artificial pain pathway (APP) was able to successfully classify shuffled data—the segmented windows from both "regular" and "anomaly" flights. Furthermore, in addition to initial setting, an increase in the windows size (more input data per classification label) reveals a slightly higher performance of the model. Based on the classification results demonstrating that the proposed framework is reliable and robust for vibration related damage, it would be an interesting direction to consider other data types and signal conversion methods to be used in the first order layer. So that, the use of APP can be extended to further prognostics and health management applications.

Acknowledgements This work was supported by the Innovation Fund Denmark (SafeEYE 0Project—no. 7049-00001A). We would like to thank Jesper Andersen (CEO & Founder at SenseAble) for his support and assistance. We would also like to extend our thanks to Simon Jensen (Assistant Engineer, Department of Electronic Systems, Aalborg University) for his help in drone operations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. *Proc Natl Acad Sci* 79(8):2554–2558
2. Goller C, Kuchler A (1996) Learning task-dependent distributed representations by backpropagation through structure. In: *Proceedings of International Conference on Neural Networks (ICNN'96)*, IEEE, 1: 347–352
3. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
4. Sak H, Senior AW, Beaufays F (2014) Long short-term memory recurrent neural network architectures for large scale acoustic modeling
5. Tai KS, Socher R, Manning CD (2015) Improved semantic representations from tree-structured long short-term memory networks, arXiv preprint [arXiv:1503.00075](https://arxiv.org/abs/1503.00075)
6. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86(11):2278–2324
7. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst* 25:1097–1105
8. Simonyan K, Zisserman (2014) A Very deep convolutional networks for large-scale image recognition, arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)
9. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9
10. Chollet F (2017) Xception: Deep learning with depthwise separable convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* 1251–1258
11. Ackley DH, Hinton GE, Sejnowski TJ (1985) A learning algorithm for boltzmann machines. *Cogn Sci* 9(1):147–169
12. Salakhutdinov R, Mnih A, Hinton G (2007) Restricted boltzmann machines for collaborative filtering. In: *Proceedings of the 24th international conference on Machine learning*, 791–798
13. Salakhutdinov R, Hinton G (2009) Deep boltzmann machines. In: *Artificial intelligence and statistics*, PMLR, pp. 448–455
14. Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In: *Icml*
15. Hinton GE (2012) A practical guide to training restricted boltzmann machines. In: *Neural networks: Tricks of the trade*, Springer 599–619
16. Hinton GE, Zemel RS (1994) Autoencoders, minimum description length, and helmholtz free energy. *Adv Neural Inf Process Syst* 6:3–10
17. Vincent P, Larochelle H, Bengio Y, Manzagol P-A (2008) Extracting and composing robust features with denoising autoencoders. In: *Proceedings of the 25th international conference on Machine learning* 1096–1103
18. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol P-A, Bottou L (2010) Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research* 11(12):3371–3408
19. Baldi P (2012) Autoencoders, unsupervised learning, and deep architectures. In: *Proceedings of ICML workshop on unsupervised and transfer learning, JMLR Workshop and Conference Proceedings* 37–49
20. Hinton GE, Krizhevsky A, Wang SD (2011) Transforming auto-encoders. In: *International conference on artificial neural networks*, Springer 44–51
21. Saha B, Goebel K, Poll S, Christophersen J (2008) Prognostics methods for battery health monitoring using a bayesian framework. *IEEE Trans Instrum Meas* 58(2):291–296
22. Commission E Commission implementing regulation (eu) 2019/947 of 24 may 2019 on the rules and procedures for the operation of unmanned aircraft., *Official Journal of the European Union*
23. Commission E Commission delegated regulation (eu) 2019/945 of 12 march 2019 on unmanned aircraft systems and on third-country operators of unmanned aircraft systems, *Official Journal of the European Union*
24. la Cour-Harbo A (2018) The value of step-by-step risk assessment for unmanned aircraft. In: *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE 149–157
25. Godfrey H (2005) Understanding pain, part 1: physiology of pain. *British journal of nursing* 14(16):846–852
26. Godfrey H (2005) Understanding pain, part 2: pain management. *British journal of nursing* 14(17):904–909
27. Iasp terminology, [website], <https://www.iasp-pain.org/resources/terminology/?ItemNumber=1698#Pain>, accessed: 2021-02-08

28. Merskey H (1994) Part iii pain terms, a current list with definitions and notes on usage, Classification of chronic pain-descriptions of chronic pain syndromes and definitions of pain terms 207–214
29. Watson J (1981) Pain mechanisms-a review: Ii, afferent pain pathways. *Australian Journal of Physiotherapy* 27(6):191–198
30. Cross SA (1994) Pathophysiology of pain. *Mayo Clinic Proceedings*, Elsevier 69:375–383
31. Willis WD, Al-Chaer ED, Quast MJ, Westlund KN (1999) A visceral pain pathway in the dorsal column of the spinal cord. *Proc Natl Acad Sci* 96(14):7675–7679
32. Almeida TF, Roizenblatt S, Tufik S (2004) Afferent pain pathways: a neuroanatomical review. *Brain Res* 1000(1–2):40–56
33. Gold JJ, Belmont KA, Thomas DA (2007) The neurobiology of virtual reality pain attenuation. *CyberPsychology & Behavior* 10(4):536–544
34. Cioffi CL (2017) Modulation of glycine-mediated spinal neurotransmission for the treatment of chronic pain. *J Med Chem* 61(7):2652–2679
35. Welch P (1967) The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Trans Audio Electroacoust* 15(2):70–73
36. Bechhoefer E, Van Hecke B, He D (2013) Processing for improved spectral analysis. In: *Annual Conference of the PHM Society* 5
37. Saidi L, Ali JB, Bechhoefer E, Benbouzid M (2017) Wind turbine high-speed shaft bearings health prognosis through a spectral kurtosis-derived indices and svr. *Appl Acoust* 120:1–8
38. Ali JB, Saidi L, Harrath S, Bechhoefer E, Benbouzid M (2018) Online automatic diagnosis of wind turbine bearings progressive degradations under real experimental conditions based on unsupervised machine learning. *Appl Acoust* 132:167–181
39. Zhang L, Lin J, Liu B, Zhang Z, Yan X, Wei M (2019) A review on deep learning applications in prognostics and health management. *Ieee*. Access 7:162415–162438
40. Abdeljaber O, Avci O, Kiranyaz S, Gabbouj M, Inman DJ (2017) Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *J Sound Vib* 388:154–170
41. Ince T, Kiranyaz S, Eren L, Askar M, Gabbouj M (2016) Real-time motor fault detection by 1-d convolutional neural networks. *IEEE Trans Industr Electron* 63(11):7067–7075
42. Janssens O, Slavkovikj V, Vervisch B, Stockman K, Loccufer M, Verstockt S, Van de Walle R, Van Hoecke S (2016) Convolutional neural network based fault detection for rotating machinery. *J Sound Vib* 377:331–345
43. Guo L, Lei Y, Li N, Xing S (2017) Deep convolution feature learning for health indicator construction of bearings. In: *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, IEEE 1–6
44. Bach-Andersen M, Rømer-Oggaard B, Winther O (2018) Deep learning for automated drivetrain fault detection. *Wind Energy* 21(1):29–41
45. Zhang W, Peng G, Li C, Chen Y, Zhang Z (2017) A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. *Sensors* 17(2):425
46. Guo M-F, Zeng X-D, Chen D-Y, Yang N-C (2017) Deep-learning-based earth fault detection using continuous wavelet transform and convolutional neural network in resonant grounding distribution systems. *IEEE Sens J* 18(3):1291–1300
47. Li H, Huang J, Ji S (2019) Bearing fault diagnosis with a feature fusion method based on an ensemble convolutional neural network and deep neural network. *Sensors* 19(9):2034
48. Al-Dulaimi A, Zabihi S, Asif A, Mohammadi A (2019) A multimodal and hybrid deep neural network model for remaining useful life estimation. *Comput Ind* 108:186–196
49. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15(1):1929–1958
50. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012) Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*
51. Barbé K, Pintelon R, Schoukens J (2009) Welch method revisited: nonparametric power spectrum estimation via circular overlap. *IEEE Trans Signal Process* 58(2):553–565
52. Schuster A (1898) On the investigation of hidden periodicities with application to a supposed 26 day period of meteorological phenomena. *Terr Magn* 3(1):13–41
53. Proakis JG, Manolakis DG, Proakis JG (1992) *Digital signal processing: principles, algorithms, and applications*. Macmillan, New York
54. Smith JO (2011) *Spectral audio signal processing*, W3K
55. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444
56. Lippmann R (1988) An introduction to computing with neural nets. *artificial neural networks. Theoretical Concepts* 209(1):36–54
57. Uas-ability.home, [youtube channel], <https://www.youtube.com/channel/UCwIUbrNZCwBuWZ4rRBUq3LA>, accessed: 2020-08-25

58. Kingma DP, Adam BJ (2015) A method for stochastic optimization, arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.