Noise modeling of offshore platform using progressive normalized distance from worst-case error for optimal neuron numbers in deep belief network

Cheng Siong Chin¹ · Ronghui Zhang²

Published online: 13 July 2020
© The Author(s) 2020

Abstract

Noise prediction is important for crew comfort in an offshore platform such as oil drilling rig. A deep neural network learning on the oil drilling rig is not widely studied. In this paper, a deep belief network (DBN) with the last layer initialized with trained DBN (named DBN-DNN) is used to model the sound pressure level (SPL) in the compartments of the oil drilling rig. The method finds an optimal number of the hidden neurons in restricted Boltzmann machine by using a normalized Euclidean distance from the worst possible error for each hidden layer progressively. The dataset used for experimental results is obtained via vibroacoustics simulation software such as VA-One and actual site measurements. The results show that output parameters such as spatial SPL, average spatial SPL, structure-borne SPL and airborne SPL improve the testing root mean square error to around 20% as compared to randomly assigning the number of neurons for each hidden layer. The testing RMSE in the output parameters has improved when compared with a multi-layer perceptron, sparse autoencoder, Softmax, self-taught learning and extreme learning machine.

Keywords Deep learning · Noise prediction · Marine and offshore industry · Sound pressure level · Structure-borne noise · Airborne noise

1 Introduction

Due to rapid technology advancement, the industries aim to reduce the manpower by incorporating artificial intelligence (AI) in a different domain. For example, adding intelligence in the sense of having conscious thought in the systems allows them to make decisions that impact performance in the maritime industry. There are many applications of AI in electric vehicle (Elassad et al. 2020), optimal path planning for research vessels (Liang and Wang 2019) and fault diagnosis of marine diesel engine (Hou et al. 2019). In addition,

a deep evolutionary modeling was applied to marine propulsion systems (Diez-Olivan et al. 2019). It helped to predict temperatures related to a marine propulsion system, detecting anomalies in operating conditions.

It was followed by using neural network for collision avoidance system (Praczyk 2020), motion planning and task assignment (Zadeh et al. 2018) in an autonomous underwater vehicle. A broader application of the fuzzy neural network for condition maintenance of marine electric propulsion system (Liang et al. 2014) was demonstrated. The results improved the operational efficiency and maintenance ability of ship. In addition, machine-learning-based approach was used in marine environment (Li et al. 2019). The approach combined cross-recurrence plot and statistical analysis. It gives visualization of marine time series and the degree of similarity between different marine factors.

The knowledge-based engineering methods (Yang et al. 2012) involved an intelligent method for ship deck design used to provide suitable suggestions to reduce design cycle time and improve work efficiency. Monte Carlo simulation (Cui and Wang 2013) was applied on container ship. Reliability analysis using the uncertainty in the design vari-

Communicated by V. Loia.

✉ Cheng Siong Chin
cheng.chin@ncl.ac.uk

Ronghui Zhang
zhangrh25@mail.sysu.edu.cn

¹ Faculty of Science, Agriculture and Engineering, Newcastle University in Singapore, Singapore 599493, Singapore

² Guangdong Key Laboratory of Intelligent Transportation System, School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou 510275, China

ables subjected to seawater corrosion was performed. The approach applied on a ship structural design problem had resulted in a more robust design approach. Another paper Abramowski (2013) used simulated annealing and genetic algorithm to model the cargo ship effective power that helps in the power prediction on-board the ship.

In addition, the neural network approach was used to predict marine traffic (Daranda 2016) under different disturbances and geographical structures. The successful prediction on the turning regions led to a future application to predict the marine traffic situation. The estimation of the main engine power of a container ship was performed using multiple linear regressions (Cepowski 2017). The approach was applied to the engine power data collected from 4414 container ships over 10 years. The results obtained were more accurate than using a simple linear regression.

Despite the successes of the neural network on marine engineering and naval architecture involving ship, the AI applications on offshore industry such as oil drilling rig are still quite limited. Recently, there is a growing interest in noise control on the oil rig platform where crews are staying on board for an extended period. The noise affects their well-being especially at night where they require rest hour in the crew accommodation block while drilling is still ongoing. A new SOLAS regulation II-1/3-12 (Resolution MSC.337(91)) requires new ships to reduce on-board noise and to protect personnel from noise.

The mandatory maximum noise level limits in different locations such as machinery spaces, accommodation and other spaces on board ships are imposed. For example, crew members without hearing protection should not expose to noise levels, exceeding 85 dB(A) for less than eight hours. In the event of more than eight hours in spaces with high noise, it should not exceed 80 dB(A). However, at least one-third of a day, they should subject to a noise level of below 75 dB(A).

In order to tackle the problem at the early design stage, a paper was first published using intelligent method (instead of expensive commercial software). A multiple generalized regression neural network model using fuzzy c-means with principal component analysis (PCA) (Chin et al. 2017) was used to reduce the size of the input parameters before predicting the sound pressure level (SPL) in different compartments on the oil rig platform. However, the method was quite computationally exhaustive as pre-processing using the fuzzy c-means and PCA was required. Hence, another paper suggested an adaptive online sequential extreme learning machine (ELM) (Chin and Ji 2018) for solving the same problem where the noise data were only available in batches. The ELM-based approach requires less training time as the parameters of the hidden layers are randomly generated without tuning. However, it exhibited a higher root mean square error over time than its counterpart such as multi-layer perceptron.

A deep neural network based on stacked autoencoder (Essien and Giannetti 2020) was also proposed. It consists of multiple layers of sparse autoencoders (AE) where outputs of each layer are connected to successive input layers. The first layer is usually meant to learn lower features, and it progresses to learn even higher-order features at a higher layer. In addition, a multilayer perceptron (MLP) (Fan et al. 2020) combined with other approaches can be used. However, it is still a class of feedforward neural network that consists of more than three layers where the input node consists of a nonlinear activation function. The updating of weight is still performed using the classical back-propagation (BP).

An extension from the single-layered neural network to multiple-layered neural networks to overcome the higher testing RMSE at the expense of reasonable training time was proposed. A deep architecture consists of several levels of nonlinear neural nets with various hidden layers (LeCun et al. 2015). Boltzmann machine (BM) is a log-linear Markov random field (MRF) (Fischer and Igel 2012). A restricted Boltzmann machine (RBM) is BM without connections between hidden–hidden and visible–visible. However, the RBM is the basic block used in the deep architectures in deep belief network (DBN) (Hinton et al. 2006) and deep Boltzmann machine (DBM). The DBN trains numerous layers of RBM via a greedy algorithm. After training the stack of RBMs, it can initialize a multi-layer neural network. On the other hand, the deep neural network (DNN) is stacked by including the last layer where it is initialized with trained DBN. The proposed network is sometimes called DBN-DNN (Tanaka and Okutomi 2014). The DBN-DNN is fine-tuned in a supervised manner using the classical back-propagation (BP) algorithm (Rumelhart et al. 1986).

In this paper, the DBN-DNN of 5-layer is used and compared with other networks such as MLP and AE to predict the four outputs, namely spatial sound pressure level (SPL), spatial average SPL, structure-borne noise and airborne noise at different octave frequencies (from 125 to 8000 Hz) using 13 input variables obtained from experiments and numerical results from the commercial software. However, issue related to the network structure such as finding the number of optimal neurons used in each hidden layer is still quite challenging in the AI research. Thus, a method to determine an optimal number of the hidden neurons in DBN-DNN is proposed. It uses a normalized Euclidean distance from the worst possible error across each hidden layer to determine the optimal number of neurons. The optimization moves progressively from first hidden layer to last hidden layer. The results using the optimized neuron number exhibit improvement in the testing root mean square error (RMSE) as compared to the DBN-DNN without the optimal search for the number of neurons or using the random number of neurons.

Most of the noise prediction uses sound spectrum and log-mel (Su et al. 2019) for subsequent CNN training. As

the log-mel inputs are color images, the computation time and resources can be quite intensive to perform the CNN for different locations on the oil drilling rig. Furthermore, they are mainly for classification tasks. In this paper, the sound pressure level (SPL) at different frequencies and spatial locations as inputs to model the sound pressure level is proposed. This paper contributes to the area of deep learning for noise prediction of spatial SPL, spatial average SPL, structure-borne noise and airborne noise at different octave frequencies on the offshore platform during the initial design stage where design data or information such as sound spectrum is limited. In addition, the paper also contributes to the finding of optimal number of neurons for each hidden layer that is still an active research areas, particularly for the frequency- and spatial-dependent data. Lastly, the overall approach eliminates the use of expensive commercial noise simulation software during the preliminary design stage of offshore rig design. Similar approach can be applied to other applications of interest.

This paper consists of the following sections. Section 2 describes the frequency-dependent noise dataset from the oil drilling rig for training; Sect. 3 reviews DBN fundamental that is essential to understand the topic. Section 4 describes the proposed neurons optimization algorithm for the hidden layers. Section 5 demonstrates the deep neural networks and comparisons with other approaches. Section 6 concludes the paper.

2 Frequency-dependent noise dataset

The frequency-dependent noise data used in this paper were obtained from the previous works (Chin et al. 2017; Chin and Ji 2018). The compartments or rooms in the oil rig can be affected by airborne, structure-borne and transmission noise. The main sound pressure level (SPL) parameters in dB(A) are mainly: output spatial SPL (or localized SPL measurement), structure-borne SPL (due to structural vibration), average spatial SPL (average of different SPL measurements within the compartment) and airborne (transmitted through the air from the noise source) SPL which can be predicted using the following input variables in Table 1. The data and related information can be downloaded from the following link: https://github.com/mcschin1/deeplearning_noise_offshoreplatform

1. total sound power level (SWL) or sound source
2. room type (NORSOK S-002 (Resolution MSC.337 (91)) administrated by Norwegian Technology Standards Institution for Norwegian offshore sector) for eight different room types based on the permitted noise levels on the board of the oil-rig

- (a) Type 1- unmanned machinery room (maximum allowable 110dBA)
 - (b) Type 2- unmanned machinery room (maximum allowable 90 dBA)
 - (c) Type 3- manned machinery room (maximum allowable 85dBA)
 - (d) Type 4- unmanned instrument room (maximum allowable 75 dBA)
 - (e) Type 5- store, workshop and instrument room (maximum allowable 70 dBA)
 - (f) Type 6- living quarter public area, change room, corridor and toilets (maximum allowable 65 dBA)
 - (g) Type 7- living quarter public area, laboratory, local control room, galley, mess room, office, gymnasium, lobby (maximum allowable 60 dBA)
 - (h) Type 8- cabin, hospital, central control room (maximum allowable 45dBA)
3. room surface area
 4. room volume
 5. first nearest source sound power levels
 6. source–receiver distance from the first source
 7. second nearest source sound power levels
 8. source–receiver distance from the second source
 9. room mean absorption coefficient
 10. the maximum sound power level of adjacent rooms
 11. panel or insulation thickness
 12. room type (refers to item 2) of the adjacent room
 13. number of decks to the main deck

The hull dimensions of the jack-up rig involved in the study can be shown in Fig. 1. The proposed approach uses dataset (Chin and Ji 2018) to train the DBN-DNN model. There are not more than 215 observations in the thirteen input variables (Table 1) across the octave frequencies (i.e., 125 Hz, 250 Hz, 500 Hz, 1000 Hz, 2000 Hz, 4000 Hz and 8000 Hz) for predicting the SPL of the oil drilling rig. Data for each input variables were obtained from different rooms on the offshore platform (Table 1) using both the commercial VA-One software and site measurements. There are around 19565 data for all seven octave frequencies where 50% of it will be used for training and remaining for testing. As shown in Table 1, all data are continuous variables except for the data used for the room types which is a discrete variable. All data are normalized to the range from 0 to 1 before the machine learning.

3 Deep belief network

A graphical representation of an RBM used to form stacked RBM is shown in Fig. 2. The RBM is pre-trained using the training data via contrastive divergence (CD) (Hinton

Fig. 1 Typical compartment (top) and model of oil drilling rig (bottom) (Chin and Ji 2018)

2002) training algorithm of the RBMs. Then, the states of the hidden binary units of the trained RBM are utilized as the training data for the next layer of the RBM. This process is repeated for other layers. The deep belief network (DBN) is constructed by stacking the trained RBMs. Adding a final decision layer to the DBN gives the DBN-DNN. The supervised training of the DBN-DNN by the back-propagation algorithm is called fine-tuning. The details of RBM to final DBN-DNN construction can be found in the following reference (Tanaka and Okutomi 2014).

The joint probabilities of visible units $\mathbf{v}$ and hidden units $\mathbf{h}$ of RBM (Fig. 2) can be denoted by

$$p(\mathbf{v}, \mathbf{h}; \mathbf{W}, \mathbf{a}, \mathbf{b}) = \exp^{E(\mathbf{v}, \mathbf{h}; \mathbf{W}, \mathbf{a}, \mathbf{b})} / \mathbf{Z} \quad (1)$$

where $\mathbf{E}$ is the energy function, $\mathbf{Z}$ is the partition function, $\mathbf{W}$ represents the weights connecting hidden and visible units and $\mathbf{a}$ and $\mathbf{b}$ are the biases of the visible and hidden layers,

respectively. For example, $\mathbf{v}=1$ (or $\mathbf{h}=1$) when $\mathbf{b}$ (or $\mathbf{c}$) will be positive or vice versa. The partition function $\mathbf{Z}$ is defined as

$$\mathbf{Z} = \sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp^{E(\mathbf{v}, \mathbf{h}; \mathbf{W}, \mathbf{a}, \mathbf{b})} \quad (2)$$

The energy function $\mathbf{E}$ in (2) can be expressed as

$$E(\mathbf{v}, \mathbf{h}; \mathbf{W}, \mathbf{a}, \mathbf{b}) = -\mathbf{h}^T \mathbf{W} \mathbf{v} - \mathbf{b}^T \mathbf{h} - \mathbf{a}^T \mathbf{v} \quad (3)$$

where $\mathbf{a}$ and $\mathbf{b}$ are the biases of the visible ($\mathbf{v}$) and hidden ($\mathbf{h}$) layers, respectively, and $\mathbf{W}$ represents the weights connecting $\mathbf{v}$ and $\mathbf{h}$.

The conditional probabilities of the hidden (visible) state given the visible (hidden) state are written as Eq. (4) and (5), respectively. The given visible and hidden states are both random binary variable. Hence,

$$P(\mathbf{h} = 1 | \mathbf{v}) = \sigma(\mathbf{W}^T \mathbf{v} + \mathbf{b}) \quad (4)$$

where σ is the sigmoid function defined as $\sigma(\mathbf{W}^T \mathbf{v} + \mathbf{b}) = \frac{1}{1 + \exp[-(\mathbf{W}^T \mathbf{v} + \mathbf{b})]}$, $\mathbf{b}$ is the bias and $\mathbf{W}$ represents the weights connecting $\mathbf{v}$ and $\mathbf{h}$.

Similarly, the conditional probabilities of the visible state given the hidden state are

$$P(\mathbf{v} = 1 | \mathbf{h}) = \sigma(\mathbf{W} \mathbf{h} + \mathbf{a}) \quad (5)$$

where σ is the sigmoid function defined as $\sigma(\mathbf{W} \mathbf{h} + \mathbf{a}) = \frac{1}{1 + \exp[-(\mathbf{W} \mathbf{h} + \mathbf{a})]}$, $\mathbf{a}$ is the bias and $\mathbf{W}$ represents the weights connecting $\mathbf{v}$ and $\mathbf{h}$.

The set of visible and hidden units are conditionally independent in RBMs. The block Gibbs sampling by simultaneously sampling visible (or hidden) units with fixed values of hidden (visible) units can be performed. A single step of Markov chain can be expressed as shown in Eqs. (6) and (7).

$$\mathbf{h}^{n+1} = \sigma(\mathbf{W}^T \mathbf{v}^n + \mathbf{b}) \quad (6)$$

$$\mathbf{v}^{n+1} = \sigma(\mathbf{W} \mathbf{h}^{n+1} + \mathbf{a}) \quad (7)$$

where $\mathbf{h}^n$ is the set of all hidden units at n -th step of the Markov chain, $\mathbf{a}$ and $\mathbf{b}$ are the biases of the visible ($\mathbf{v}$) and hidden ($\mathbf{h}$) layers, respectively, and σ is the sigmoid function.

However, the update of each parameter in the process needs one such chain to convergence. It can be quite computationally exhaustive. The principle of CD can reduce the sampling process to just a single-step CD (known as CD-1). As seen in Fig. 3, it consists of the positive phase where the input sample went into the input layer $\mathbf{v}_1$ and propagated to the hidden layer to give $\mathbf{h}_1$ via Eq. (4). The result of the hidden layer $\mathbf{h}_1$ propagates down to a new or reconstructed

Table 1 Types of input and output variables (and limits) (Chin et al. 2017)

No.	Input variables and outputs	125 Hz		250 Hz		500 Hz		1000 Hz		2000 Hz		4000 Hz		8000 Hz	
		Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.
Inputs															
1	Sound power level (dBA)	104.6	0.0	115.2	0.0	122.0	0.0	128.0	0.0	123.0	0.0	2052.0	39.2	114.0	0.0
2	Room Type	8.0	1.0	8.0	1.0	8.0	1.0	8.0	1.0	8.0	1.0	2160.0	16.2	8.0	1.0
3	Room surface area (m2)	2052.0	39.2	2052.0	39.2	2052.0	39.2	2052.0	39.2	2052.0	39.2	119.0	0.0	2052.0	39.2
4	Room volume, V (m3)	2160.0	16.2	2160.0	16.2	2160.0	16.2	2160.0	16.2	2160.0	16.2	20.0	0.0	2160.0	16.2
5	Nearest source 1 SWL(dBA)	101.0	0.0	112.0	0.0	119.0	0.0	125.0	0.0	120.0	0.0	119.0	0.0	111.0	0.0
6	Dist. to source 1 (m)	20.0	0.0	20.0	0.0	20.0	0.0	20.0	0.0	20.0	0.0	20.2	0.0	20.0	0.0
7	Nearest source 2 SWL (dBA)	101.0	0.0	112.0	0.0	119.0	0.0	125.0	0.0	120.0	0.0	0.5	0.0	111.0	0.0
8	Dist. To nearest source 2 (m)	20.2	0.0	20.2	0.0	20.2	0.0	20.2	0.0	20.2	0.0	122.0	0.0	20.2	0.0
9	Mean absorption coefficient	0.3	0.0	0.6	0.0	0.7	0.0	0.6	0.0	0.6	0.0	75.0	1.0	0.5	0.0
10	Max sound power level of adjacent room (dBA)	104.6	0.0	115.2	0.0	122.0	0.0	128.0	0.0	123.0	0.0	75.0	0.0	114.0	0.0
11	Room type of adjacent room	8.0	1.0	8.0	1.0	8.0	1.0	8.0	1.0	8.0	1.0	6.0	-2.0	8.0	1.0
12	Panel/insulation thickness (mm)	75.0	0.0	75.0	0.0	75.0	0.0	75.0	0.0	75.0	0.0	28.6	-25.5	75.0	0.0
13	Number of decks to main deck	6.0	-2.0	6.0	-2.0	6.0	-2.0	6.0	-2.0	6.0	-2.0	94.1	8.8	6.0	-2.0
Outputs															
14	Spatial SPL (dBA)	90.5	20.4	97.2	21.0	103.3	16.2	109.4	12.9	104.5	9.9	5.0	1.0	95.9	0.0
15	Spatial average SPL (dBA)	89.8	20.4	96.5	21.0	101.6	16.2	108.0	12.9	103.0	9.9	19.4	0.0	94.6	0.0
16	Structure-borne SPL (dBA)	45.8	0.0	90.8	0.0	93.3	0.0	98.2	0.0	93.2	0.0	89.8	0.0	73.7	0.0
17	Airborne SPL (dBA)	58.2	0.0	104.4	0.0	100.6	0.0	104.6	0.0	105.7	0.0	106.4	0.0	96.5	0.0

Fig. 2 Graphical representation of stacked RBM, DBN and DBN-DNN

Fig. 3 Input data and reconstruction during CD training

visible layer $\mathbf{v}_2$ via Eq. (5) Then to $\mathbf{h}_2$ again. The change of the parameters in the CD training can be expressed as

$$\Delta w_{ij} = \epsilon (\langle v_i^1 h_j^1 \rangle - \langle v_i^2 h_j^2 \rangle) \quad (8)$$

where ϵ is the training rate, the superscripts of 1 and 2 of h_j and v_i are used to denote the step in the sampling process and the angle bracket refers to the average training data.

The inference Tanaka and Okutomi (2014) of the RBM by taking expectation after the activation function is defined as

$$\mathbf{P}(\zeta_{l+1} = 1) = E_{\mathbf{P}(\zeta_l)}[\sigma(\mathbf{W}_l^T \zeta_l + \mathbf{b}_l)] \quad (9)$$

where σ is the sigmoid function, E is the expectation of the probability $\mathbf{P}(\cdot)$, ζ_l is the input, $\mathbf{b}_l$ is the bias and $\mathbf{W}_l^T$ represents the weights.

The conditional probabilities of the $(l+1)$ -th node given all possible combinations of binary states of the l -th node are computed. It is followed by evaluating the expectation of conditional probabilities. The eventual closed-form approximation (Tanaka and Okutomi 2014) of (9) can be expressed as

$$\mathbf{P}(\zeta_{l+1}^j = 1) = E[\sigma(w_l^j \zeta_l + b_l^j)] \simeq \sigma\left(\frac{\mu}{\sqrt{1 + \pi \rho^2 / 8}}\right) \quad (10)$$

where E is the expectation, w_l^j is the j -th column vector of the matrix $\mathbf{W}_l$, b_l^j is the j -th column vector of the matrix $\mathbf{b}_l$, ζ_l is the input, σ is the sigmoid function, ρ^2 is the variance and μ is the mean.

Here, the mean μ and variance ρ^2 Tanaka and Okutomi (2014) in (10) can be defined as follows.

$$\begin{aligned} \mu &= E[\zeta_{l+1}^j] = E[w_l^j \zeta_l + b_l^j] \\ &= \sum_i w_l^{ij} \mathbf{P}(\zeta_l^i = 1) + b_l^j \end{aligned} \quad (11)$$

$$\begin{aligned} \rho^2 &= V[\zeta_{l+1}^j] = V[w_l^j \zeta_l + b_l^j] \\ &= \sum_i w_l^{ij2} \mathbf{P}(\zeta_l^i = 1) \{1 - \mathbf{P}(\zeta_l^i = 1)\} \end{aligned} \quad (12)$$

where $\mathbf{P}(\cdot)$ is the probability, w_l^{ij} is the (i,j) -th element of the matrix $\mathbf{W}_l$, b_l^{ij} is the ij -th column vector of the matrix $\mathbf{b}_l$, ζ_l is the input and $V[\zeta_{l+1}^j]$ represents the variance of ζ_{l+1}^j .

The forward inference of the DBN-DNN with the inference is easy. For the fine-tuning of the DBN-DNN, back-propagation two sets of partial derivatives (Wang and Manning 2013) are required. The back-propagation process can be obtained similarly using classical BP. The details can be found in Tanaka and Okutomi (2014), and its potential application can be seen in Zhang et al. (2017), Sun et al. (2018) and Xiong et al. (2018).

Fig. 4 Overall flow of optimizing number of neurons from first to last hidden layer in DBN-DNN

4 Neuron optimization algorithm for hidden layers

Problems related to the network structure such as the number of neurons for each hidden layer are still a challenging task within the AI research. In this paper, the optimal number of neurons used in the three hidden layers is based on the idea of Euclidean Distance. The method finds an optimal number of the hidden neurons in DBN-DNN by using a normalized Euclidean distance between the current RMSE and the worst possible error. The approach applies on the first hidden layer

and propagates to the last hidden layer progressively as seen in Fig. 4 before validating the final set of hidden neurons.

In the process of finding the optimal number of hidden neurons in one layer, the remaining hidden layer is set to one. A final validation using the optimal set of hidden neuron is performed to ensure the RMSE is indeed smaller than the worst-case value.

In summary, the procedure of finding the optimal neurons for one hidden layer is as follows.

1. The root mean square error (RMSE) of the test sample (E_{ij}) consisting of four actual SPL outputs (j), namely spatial SPL, average spatial SPL, structure-borne SPL and airborne SPL across different sets of neuron numbers (i), is computed. For clarity, only ten different numbers of neurons starting from 2 to 11 are used. The RMSE matrix can be tabulated in Table 2
2. The best-case (equal to 0) and worst-case (equal to 80dBA as it should not exceed this value under SOLAS regulation II-1/3-12 [14]) RMSE value for each SPL output are also provided in the RMSE matrix as seen in Table 2.
3. The RMSE matrix is then normalized by the root mean square (RMS) value for each SPL output in Eq. (13). The normalized values can be seen in Table 3.
4. The Euclidean distance between the current and the best-case value of RMSE for the SPL outputs is given in Eq. (14). The values for the Euclidean distance are tabulated

$$R_{ij} = \frac{E_{ij}}{\sqrt{\sum_{j=1}^m E_{ij}^2}}, \forall i = 1, \dots, n; j = 1, \dots, m \quad (13)$$

where $\text{RMS} \sqrt{\sum_{j=1}^m E_{ij}^2}$ is computed as [85.773 86.510 88.174 87.937]

Table 2 Testing RMSE of SPL outputs at different numbers of neurons

No. of Neurons	Spatial SPL(dBA)	Average spatial SPL(dBA)	Structure-borne SPL(dBA)	Airborne SPL (dBA)
2	9.0090	11.117	9.0475	15.127
3	11.510	11.802	13.985	12.965
4	9.9609	10.222	12.575	10.594
5	9.5580	10.822	10.360	9.6611
6	9.1339	9.6388	12.282	11.120
7	9.5831	9.8609	12.266	11.007
8	10.146	10.525	11.480	12.200
9	9.7829	10.020	12.759	10.958
10	9.7910	10.111	11.388	9.7678
11	9.1148	9.7975	10.303	11.015
R^-	0	0	0	0
R^+	80	80	80	80

Table 3 Normalized testing RMSE of SPL outputs at different numbers of neurons

No. of neurons	Spatial SPL (dBA)	Average spatial SPL (dBA)	Structure-borne SPL (dBA)	Airborne SPL (dBA)
2	0.1050	0.1285	0.1026	0.1720
3	0.1342	0.1364	0.1586	0.1474
4	0.1161	0.1182	0.1426	0.1205
5	0.1114	0.1251	0.1175	0.1099
6	0.1065	0.1114	0.1393	0.1265
7	0.1117	0.1140	0.1391	0.1252
8	0.1183	0.1217	0.1302	0.1387
9	0.1141	0.1158	0.1447	0.1246
10	0.1142	0.1169	0.1292	0.1111
11	0.1063	0.1133	0.1169	0.1253
R^-	0	0	0	0
R^+	0.9327	0.9247	0.9073	0.9097

Table 4 Distance between current and best value of RMSE for SPL outputs at different numbers of neurons

No. of neurons	Spatial SPL (dBA)	Average spatial SPL (dBA)	Structure-borne SPL (dBA)	Airborne SPL (dBA)	S_j^+
2	0.0110	0.0165	0.0105	0.0296	0.2601
3	0.0180	0.0186	0.0252	0.0217	0.2890
4	0.0135	0.0140	0.0203	0.0145	0.2496
5	0.0124	0.0157	0.0138	0.0121	0.2323
6	0.0113	0.0124	0.0194	0.0160	0.2432
7	0.0125	0.0130	0.0194	0.0157	0.2460
8	0.0140	0.0148	0.0170	0.0192	0.2549
9	0.0130	0.0134	0.0209	0.0155	0.2508
10	0.0130	0.0137	0.0167	0.0123	0.2360
11	0.0113	0.0128	0.0137	0.0157	0.2312
R^-	0	0	0	0	0
R^+	0.8699	0.8551	0.8232	0.8276	1.8373

in Table 4.

$$S_j^+ = \sqrt{\sum_{i=1}^n (E_{ij} - R_i^+)^2}, \forall j = 1, \dots, m \quad (14)$$

- The Euclidean distance between the current and the worst-case value of RMSE for the SPL outputs is given in Eq. (15). The values for the Euclidean distance can be seen in Table 5.

$$S_j^- = \sqrt{\sum_{i=1}^n (E_{ij} - R_i^-)^2}, \forall j = 1, \dots, m \quad (15)$$

- Calculate the closeness or distance to the best value by computing the ratio of the worst-case RMSE value over the sum of both the worst-case and the best-case values of the RMSE for the SPL outputs in Eq. (16). The values

can be seen in Table 6.

$$C_j^+ = \frac{S_j^-}{S_j^- + S_j^+}, \forall j = 1, \dots, m \quad (16)$$

- Finally, perform ranking on the ratio obtained from Eq. (16). As seen in Table 6, the optimal number of neurons is the higher value that is closer to the best case, S_j^+ . In this case, the highest ratio or the optimal number of neuron is 11th neuron. Note that R^+ (or R^-) is not included as it is always the highest (or the lowest).
- The optimal number of neurons for first hidden layer is used and the same procedure repeats for the 2nd until it reaches the last hidden layer. During the process of finding the optimal number of the hidden neurons in one layer, the remaining hidden layer is set to 1.
- The final optimal number of neurons for all hidden layers is then used. The testing RMSE of the SPL outputs is validated to ensure that the optimal neuron numbers

Table 5 Distance between current and worst value of RMSE for SPL outputs at different numbers of neurons

No. of neurons	Spatial SPL (dBA)	Average spatial SPL (dBA)	Structure-borne SPL (dBA)	Airborne SPL (dBA)	S_j^-
2	0.6850	0.6340	0.6475	0.5442	1.5845
3	0.6376	0.6214	0.5605	0.5811	1.5494
4	0.6668	0.6506	0.5847	0.6229	1.5890
5	0.6745	0.6394	0.6238	0.6398	1.6054
6	0.6826	0.6615	0.5898	0.6135	1.5961
7	0.6740	0.6573	0.5901	0.6155	1.5928
8	0.6632	0.6449	0.6039	0.5944	1.5832
9	0.6702	0.6543	0.5815	0.6164	1.5882
10	0.6700	0.6526	0.6055	0.6379	1.6019
11	0.6830	0.6585	0.6248	0.6154	1.6068
R^-	0.8699	0.8551	0.8232	0.8276	1.8373
R^+	0	0	0	0	0

Table 6 Optimal number of neurons by examining C_j^+ and its ranking

No. of Neurons	S_j^+	S_j^-	C_j^+	Rank
2	0.2601	1.5845	0.8589	10
3	0.2890	1.5494	0.8427	11
4	0.2496	1.5890	0.8642	7
5	0.2323	1.6054	0.8736	3
6	0.2432	1.5961	0.8677	5
7	0.2460	1.5928	0.8662	6
8	0.2549	1.5832	0.8613	9
9	0.2508	1.5882	0.8636	8
10	0.2360	1.6019	0.8715	4
11	0.2312	1.6068	0.8741	2
R^-	0	1.8373	1.0000	12
R^+	1.8373	0	0	1

(Table 7 [11 – 3 – 2]) give the lower RMSE values than the worst case (see Table 8 equal to [5 – 11 – 8]).

The results of the neuron optimization algorithm for all hidden layers can be seen in the next section.

5 Experimental results

5.1 Neuron optimization algorithm for hidden layers at different frequencies

The neuron optimization algorithm is applied to DBN-DNN. The optimal number of neurons can be seen in Table 7 for 125 Hz. It is found that the optimal number of neurons is 11, 3 and 2. The average RMSE for the testing using the proposed optimal neuron number is approximately 20% less

than the random assignment of the number of neurons. Similar phenomena can be seen in Fig. 5 where the SPL outputs have a closer match to the actual outputs as compared to other number of neurons.

Instead of using the same number of neurons for other frequencies, namely 250 Hz, 500 Hz, 1000 Hz, 2000 Hz, 4000 Hz and 8000 Hz, the results of the neuron optimization algorithm can be seen in Table 9. The average RMSE of the SPL outputs using different optimal numbers of neurons is higher than its counterpart that uses the same number of neurons [11 – 3 – 2]. In addition to the more training time required as a result of more number of optimization, the results for optimizing the number of neurons for each frequency do not show prominent improvement. In fact, the RMSE is slightly higher than the constant optimal number of neurons. Hence, the constant optimal number of neurons [11 – 3 – 2] is used for all frequencies.

5.2 Comparisons with other neural networks

Next, the DBN-DNN with the optimal number of neurons is then compared with the MLP, sparse autoencoder (AE), Softmax, self-taught learning (STL) and extreme learning machine (ELM). The results of other approaches tabulated in Tables 10 and 11 can be found in the paper Chin and Ji (2018). The sigmoid function was used except for ELM that used sinusoidal activation function. The number of neurons used for MPL, AE, Softmax, STL and ELM is as follows: 65, 25, 15, 200 and 2, respectively. The details can be seen in Tables 10 and 11.

The hardware environment includes a laptop with Intel 2.40 GHz CPU and 16G Memory. A total of 215 data for each input parameters for all octave frequency: 125 Hz, 250 Hz, 500 Hz, 1000 Hz, 2000 Hz, 4000 Hz and 8000 Hz, were used for machine learning. The thirteen input data can be

Table 7 RMSE for testing using an optimal number of neurons [11 – 3 – 2] at 125 Hz

No. of neurons	2nd layer	3rd layer	4th layer	Localized SPL (dBA)	Room average SPL (dBA)	Structure-borne SPL (dBA)	Airborne SPL (dBA)
2	0.8645	0.8523	[0.8734]	9.6755	9.6739	10.3739	9.4692
3	0.8650	[0.8666]	0.8560	9.8224	9.8089	10.4888	9.8263
4	0.8635	0.8500	0.8600	10.2038	10.1822	10.8532	9.7141
5	0.8631	0.8450	0.8688	7.9167	7.8851	8.6260	10.6013
6	0.8634	0.8302	0.8461	10.1263	10.1048	10.8095	9.9359
7	0.8636	0.8169	0.8425	9.0605	9.1735	11.2480	10.4736
8	0.8631	0.8480	0.8352	10.4275	10.1482	10.6753	13.7164
9	0.8644	0.8570	0.8657	10.5758	10.5463	11.2578	10.0299
10	0.8636	0.8662	0.8394	10.9348	10.6485	11.3260	13.9328
11	[0.8662]	0.8045	0.8675	10.4299	10.3875	11.0487	10.2850
Average				9.9173	9.8559	10.6707	10.7985

Table 8 RMSE for testing using random assignment of number of neurons [5 – 11 – 8] at 125 Hz

No. of neurons	Localized SPL (dBA)	Room average SPL (dBA)	Structure-borne SPL (dBA)	Airborne SPL (dBA)
2	12.6750	11.9985	11.5623	11.8973
3	11.1339	10.6383	11.1423	14.0363
4	14.3013	13.8551	13.5527	16.2555
5	4.7404	14.2091	14.1974	12.2804
6	9.6866	9.2504	9.3485	9.0593
7	9.6679	9.1342	9.1349	10.6857
8	14.7574	14.0909	13.7873	19.2516
9	15.2133	14.5506	14.6168	13.3477
10	16.6444	15.9197	14.7827	13.1876
11	13.5756	12.7343	13.9823	13.3356
Average	13.2396	12.6381	12.6107	13.3337

seen in Table 1. As such, a total of 19565 points were used for the noise prediction. Around 50% of the dataset was utilized for training and remaining was for testing. The performance was evaluated using the root mean square error (RMSE) of the four outputs: spatial SPL, average spatial SPL, structure-borne SPL and airborne SPL. The time spent on training is tabulated in Tables 10 and 11 for all the octave frequency. A three-hidden-layer DBN-DNN (excluding the input and output layer) was used.

The number of hidden units or neurons for the three-layer is configured as [11 – 3 – 2]. All the RBMs used in DBN-DNN are Bernoulli–Bernoulli RBMs. The maximum iteration was set to 500 that can be adjusted to fine-tune the results. The momentum value is between 0 and 1. To jump from local minima, the momentum value has to be high. If it becomes too large, the learning rate needs to be smaller. A substantial momentum causes a faster convergence. With both momentum and learning rate at high values, it will skip the minimum with a large step.

Conversely, a small momentum can increase the training time and can trap in local minimal. However, it can help to smooth out the variations for changing gradient. In this paper, the right value of momentum was obtained through trial and error. The initial momentum was set to 0.5 for the first five iterations followed by 0.9. The weight cost was set to 0.0002. It is used to decrease the cost function. By dropping out units (both hidden and visible), it helps to reduce overfitting issue. A dropout rate of 0.5 was used. The learning step size was set to 0.035. The inputs were normalized to 0 to 1 in order to reduce the training time and the risk of the local optima.

As observed in Tables 10 and 11, the DBN-DNN has the lowest testing RMSE among MLP, AE, Softmax, STL and ELM in all the selected octave frequency. However, the time taken to complete the training is moderately faster than MLP and Softmax. However, the DBN-DNN is slower than the ELM. The ELM is faster as it is based on a straightforward formula or structure that can achieve quite a fast training speed with only one hidden layer (not trained via a constant weight on the incoming connections). The training time for

Fig. 5 SPL output responses using optimal and random number of neurons for DBN-DNN

Table 9 Average RMSE for testing using different and similar optimal numbers of neurons across different frequencies

Frequency (Hz)	Number of hidden neurons	Localized SPL (dBA)	Room average SPL (dBA)	Structure-borne SPL (dBA)	Airborne SPL (dBA)
125	[11-3-2]	9.9173	9.8559	10.670	10.798
250	[4-4-7]	10.002	10.024	11.352	10.123
500	[5-4-9]	6.8762	7.0071	10.601	14.897
1000	[5-2-2]	6.9489	7.0175	9.6937	11.217
2000	[2-2-11]	14.090	12.432	17.105	14.075
4000	[2-2-2]	9.6184	9.5127	17.797	13.407
8000	[6-2-2]	6.9584	6.9737	9.5532	8.1520
	Average	9.2016	8.9748	12.396	11.810
125	[11-3-2]	9.9173	9.8558	10.670	10.798
250	[11-3-2]	10.036	9.9991	12.382	11.319
500	[11-3-2]	10.703	10.291	13.540	12.975
1000	[11-3-2]	10.506	10.228	13.599	13.062
2000	[11-3-2]	8.1799	7.4173	11.196	9.7153
4000	[11-3-2]	5.3298	5.2760	8.0262	8.5458
8000	[11-3-2]	6.8729	6.8742	5.9918	6.0875
	Average	8.7923	8.5632	10.772	10.357

DBN-DNN is around 20 s. Although ELM has a simple structure such as less neurons can be used for the hidden layer, the RMSE of ELM is not sufficiently small. Hence, a multi-layer neural network such as DBN-DNN is proposed to reduce the testing RMSE.

As shown in Figs. 6, 7, 8, 9, 10, 11 and 12, the mean outputs of each neuron in the hidden layers of the DBN-DNN were plotted. As observed, each hidden layer is trying to learn the features of the thirteen input parameters. The first hidden layer is usually meant to learn lower features, and it progresses to learn even higher-order features at a higher

Table 10 Performance of DBN-DNN as compared to other approaches at 125 Hz, 250 Hz, 500 Hz and 1000 Hz

Frequency (Hz)	Approaches	Testing root mean square error (RMSE)					Training time (in second)
		Spatial SPL (dBA)	Average spatial SPL (dBA)	Structure-borne SPL (dBA)	Airborne SPL (dBA)		
125	DBN-DNN(sig, [13,11,3,2,4])	9.917	9.856	10.67	10.79		19.98
	MLP(sig,65 nodes)	33.06	32.91	25.37	25.33		27.94
	AE (sig,25 nodes)	18.75	16.92	20.13	21.11		16.14
	Softmax (sig,15 nodes)	40.03	40.62	41.31	44.74		74.52
	STL(sig,200 nodes)	31.40	51.152	45.35	28.74		0
	ELM(sin,2 nodes)	28.23	13.08	23.22	24.41		2.340
250	DBN-DNN	10.03	9.99	12.38	11.32		17.81
	MLP	25.95	16.57	25.44	21.46		1.990
	AE	43.18	43.12	25.47	25.54		28.13
	Softmax	12.84	13.15	14.72	14.37		15.87
	STL	46.16	49.07	40.54	44.81		66.84
	ELM	55.13	30.87	33.47	23.15		0
500	DBN-DNN	10.70	10.29	13.51	12.97		18.89
	MLP	19.62	26.74	30.41	17.94		2.400
	AE	43.80	43.71	25.30	25.34		26.56
	Softmax	7.750	7.689	14.34	17.82		15.76
	STL	48.63	46.68	40.32	48.34		67.47
	ELM	72.24	50.05	48.70	5.78		0
1000	DBN-DNN	10.50	10.22	13.60	13.06		20.41
	MLP	15.08	22.93	26.15	21.52		2.390
	AE	43.89	43.81	25.26	25.31		23.58
	Softmax	9.040	8.743	15.30	16.12		15.52
	STL	42.87	44.29	46.08	45.83		66.85
	ELM	55.78	66.42	67.96	29.24		0

layer. Unlike face recognition, each layer can learn edges and face features such as nose, eyes and mouth. In noise prediction, each hidden layer will learn the importance of the input parameters such as level of the noise source, the volume of the room, distance from the noise source and noise insulation or absorption coefficient of the room. However, the outputs of the neurons may not coincide with the input parameters, and the image plots may not provide information such as edge and features of the face (in the case of face recognition problem). Nevertheless, some insights into the performance of DBN-DNN in each hidden layer can be obtained.

For example, in Fig. 6 (for 125 Hz), more emphasis is placed on first and second hidden neurons to achieve the targeted spatial SPL, average spatial SPL, structure-borne SPL and airborne SPL. As the training continues to the next layer, the mean activation outputs become more prominent. The 4th layer gives the outputs of DBN-DNN in the range of 0 to 1 that are then converted back to the actual SPL values. As seen

in Fig. 6, 7, 8, 9, 10, 11 and 12, the mean activation value of each neuron is quite different in each layer. In general, higher activation outputs can be found in spatial SP and average spatial SPL. On the other hand, the structure-borne SPL has the smallest activation output as compared to the rest. From the training, it can observe that a few layers are required to train the dataset in order to learn the necessary features that affect the SPL. Increasing the hidden layer beyond three does not have much improvement on the testing RMSE.

As observed from Figs. 13, 14, 15, 16, 17, 18 and 19, the responses of the noise prediction in each frequency can be plotted. In general, the RMSE of the SPL outputs is not more than 10dBA. The prediction of the overall SPL, in particular the airborne noise, performs better at a higher frequency of 2000 Hz and beyond. It is due to the fact that the airborne noise is characterized by higher frequency. As observed, there exist some deviations between the actual and prediction by DBN-DNN. It is due to the presence of both

Table 11 Performance of DBN-DNN as compared to other approaches at 2000 Hz, 4000 Hz and 8000 Hz

Frequency (Hz)	Approaches	Testing root mean square error (RMSE)				Training time (in second)
		Spatial SPL (dBA)	Average spatial SPL (dBA)	Structure-borne SPL (dBA)	Airborne SPL (dBA)	
2000	DBN-DNN (sig, [13,11,3,2,4])	8.180	7.417	11.19	9.715	21.51
	MLP(sig,65 nodes)	21.38	23.08	29.24	25.68	2.960
	AE (sig,25 nodes)	39.65	39.53	25.23	25.25	24.20
	Softmax (sig,15 nodes)	9.830	8.204	13.33	18.08	15.19
	STL(sig,200 nodes)	48.14	47.20	38.66	46.94	65.28
	ELM(sin,2 nodes)	57.89	82.34	67.69	16.54	0
4000	DBN-DNN	5.330	5.276	8.026	8.546	22.01
	MLP	23.63	25.41	22.30	19.30	2.590
	AE	37.53	37.38	25.26	25.24	26.36
	Softmax	16.39	10.46	10.40	15.39	15.24
	STL	47.45	35.12	37.71	48.29	28.46
	ELM	39.15	43.66	59.73	18.97	0
8000	DBN-DNN	6.873	6.874	5.992	6.088	23.32
	MLP	27.22	25.34	24.42	26.50	2.180
	AE	37.73	37.61	25.28	25.30	29.74
	Softmax	9.927	9.805	11.43	10.14	14.61
	STL	46.29	35.59	39.81	48.07	25.22
	ELM	47.44	52.84	58.12	45.06	0

Fig. 6 Mean activation function outputs for each layer for 125 Hz using DBN-DNN

Fig. 7 Mean activation function outputs for each layer for 250 Hz using DBN-DNN

Fig. 8 Mean activation function outputs for each layer for 500 Hz using DBN-DNN

continuous and discrete variables in the dataset. Although the normalization was performed to have the same range of values for each of the inputs to guarantee stable con-

vergence of weight and biases, the normalization can cause errors in the training of the data (with outliers) when they are passed through the nonlinearities in each layer before the

Fig. 9 Mean activation function outputs for each layer for 1000 Hz using DBN-DNN

Fig. 10 Mean activation function outputs for each layer for 2000 Hz using DBN-DNN

outputs. However, the testing RMSE can further be improved by fine-tuning the hyper-parameters of DBN-DNN such as step size and momentum values, but at the expense of higher

computation time in training and testing if the number of neurons or layer increases. Moreover, it may not reduce the RMSE.

Fig. 11 Mean activation function outputs for each layer for 4000 Hz using DBN-DNN

Fig. 12 Mean activation function outputs for each layer for 8000 Hz using DBN-DNN

Fig. 13 Spatial, average spatial, structure-borne and airborne SPL for 125 Hz using DBN-DNN

Fig. 14 Spatial, average spatial, structure-borne and airborne SPL for 250 Hz using DBN-DNN

Fig. 15 Spatial, average spatial, structure-borne and airborne SPL for 500 Hz using DBN-DNN

Fig. 16 Spatial, average spatial, structure-borne and airborne SPL for 1000 Hz using DBN-DNN

Fig. 17 Spatial, average spatial, structure-borne and airborne SPL for 2000 Hz using DBN-DNN

Fig. 18 Spatial, average spatial, structure-borne and airborne SPL for 4000 Hz using DBN-DNN

Fig. 19 Spatial, average spatial, structure-borne and airborne SPL for 8000 Hz using DBN-DNN

6 Conclusion

The deep belief network (DBN) including the last layer where it was initialized with trained DBN called DBN-DNN was used to model the sound pressure level (SPL) of the compartments in the oil drilling rig. The supervised pre-training changed the weights in a greedy layer-wise fashion. The two layers at a time were trained as a restricted Boltzmann machine (RBM) where one of the hidden layers acted as the visible layer for the next. It was followed by the supervised pre-tuning that adjusted these parameters using the standard back-propagation.

To improve on the neural network structure, the normalized Euclidean distance from the worst possible error for each hidden layer was used to determine the optimal number of the hidden neurons in RBM. It was performed in a progressive manner from the first to the last hidden layer. The results indicated that the spatial SPL, average spatial SPL, structure-borne SPL and airborne SPL improved the testing root mean square error (RMSE) to approximately 20% as compared to its counterpart using the random assignment of number of neurons.

In addition, as compared with the other machine learning approaches such as multilayer perceptron (MLP), sparse autoencoder (AE), Softmax, self-taught learning (STL) and extreme learning machine (ELM), the experimental results

showed that the RMSE in the spatial SPL, average spatial SPL, structure-borne SPL and airborne SPL had improved.

In the future, more dataset will be sought for training, and further adaptive-based fine-tuning will be incorporated into the DBN-DNN to reduce the testing RMSE.

Acknowledgements The authors would like to thank Newcastle University in Singapore and National Natural Science Foundation of China for the support. This study was funded by National Natural Science Foundation of China (Grant No. 51775565).

Compliance with ethical standards

Conflict of interest There is no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

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 copy-

right holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abramowski T (2013) Application of artificial intelligence methods to preliminary design of ships and ship performance optimization. *Naval Eng J* 125(3):101
- Cepowski T (2017) Prediction of the main engine power of a new container ship at the preliminary design stage. *Manag Syst Prod Eng* 2(26):97–99
- Chin CS, Ji X (2018) Adaptive online sequential extreme machine learning for frequency dependent noise data on offshore oil rig. *Eng Appl Artif Intel* 74:226–241
- Chin CS, Ji X, Woo WL, Kwee TJ, Yang WX (2017) Modified multiple generalized regression neural network models using fuzzy C-means with principal component analysis for noise prediction of offshore platform. *Neural Comput Appl* 2017:1–16
- Cui J, Wang D (2013) Application of knowledge-based engineering in ship structural design and optimization. *Ocean Eng* 72:124–139
- Daranda A (2016) A neural network approach to predict marine traffic. Technical Report MII-DS-07T-16-9-16. Vilnius University Institute of Mathematics and Informatics, Lithuania
- Diez-Olivan A, Pagan JA, Sanz R, Sierra B (2019) Deep evolutionary modeling of condition monitoring data in marine propulsion systems. *Soft Comput* 23:9937–9953
- Elassad ZEA, Mousannif H, Moatassime HA, Karkouch A (2020) The application of machine learning techniques for driving behavior analysis: a conceptual framework and a systematic literature review. *Eng Appl Artif Intell* 87:103312
- Essien AE, Giannetti C (2020) A deep learning model for smart manufacturing using convolutional LSTM neural network autoencoders. *IEEE Trans Ind Inf* 99:1
- Fan Y, Xu K, Wu H, Zheng Y, Tao B (2020) Spatiotemporal modeling for nonlinear distributed thermal processes based on KL decomposition, MLP and LSTM network. *IEEE Access* 8:25111
- Fischer A, Igel C (2012) An Introduction to Restricted Boltzmann machine, progress in pattern recognition, image analysis. *Comput Vis Appl* 7441:14–36
- Hinton GE (2002) Training products of experts by minimizing contrastive divergence. *Neural Comput* 14(8):1771–1800
- Hinton GE, Osindero S, Teh Y (2006) A fast learning algorithm for deep belief nets. *Neural Comput* 18:1527–1554
- Hou L, Zou J, Du C, Zhang J (2019) A fault diagnosis model of marine diesel engine cylinder based on modified genetic algorithm and multilayer perceptron. *Soft Comput*
- LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521:436–444
- Li Z, Cai D, Wang J, Li Y, Gui G, Sun X, Wang N, Zhang J, Liu H, Wang G (2019) Machine learning based dynamic correlation on marine environmental data using cross-recurrence strategy. *IEEE Access* 7:17
- Liang S, Yang J, Wang Y, Wang M (2014) Fuzzy neural network in condition maintenance for marine electric propulsion system, 2014 IEEE Conference and Expo Transportation Electrification Asia-Pacific, August 31–Sept 3. Beijing, China
- Liang Y, Wang L (2019) Applying genetic algorithm and ant colony optimization algorithm into marine investigation path planning model. *Soft Comput*
- Praczyk T (2020) Neural collision avoidance system for biomimetic autonomous underwater vehicle. *Soft Comput* 24:1315–1333
- Resolution MSC.337 (91). Code on noise levels on board ships. [http://www.imo.org/en/KnowledgeCentre/IndexofIMOResolutions/Maritime-Safety-Committee-\(MSC\)/Documents/MSC.337\(91\).pdf](http://www.imo.org/en/KnowledgeCentre/IndexofIMOResolutions/Maritime-Safety-Committee-(MSC)/Documents/MSC.337(91).pdf). Accessed 27 Feb 2020
- Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. *Nature* 323:533–536
- Su Y, Zhang K, Wang J, Madani K (2019) Environment sound classification using a two-stream CNN based on decision-level fusion. *Sensors* 19(7):1733
- Sun X, Zhang H, Meng W, Zhang R, Li K, Peng T (2018) Primary resonance analysis and vibration suppression for the harmonically excited nonlinear suspension system using a pair of symmetric viscoelastic buffers. *Nonlinear Dyn* 94(2):1243–1265
- Tanaka M, Okutomi M (2014) A novel inference of a restricted Boltzmann machine. In: 22nd International conference on pattern recognition, pp 1526–1531
- Wang S, Manning C (2013) Fast dropout training. *ICML* 2013:118–126
- Xiong H, Zhu X, Zhang R (2018) Energy recovery strategy numerical simulation for dual axle drive pure electric vehicle based on motor loss model and big data calculation, complexity, 2018, Article ID 4071743, 14 pages
- Yang HZ, Chen JF, Ma N, Wang DY (2012) Implementation of knowledge-based engineering methodology in ship structural design. *Comput Aided Des* 44:196–202
- Zadeh SM, Powers DMW, Sammut K, Yazdani AM (2018) A novel versatile architecture for autonomous underwater vehicle? Motion planning and task assignment. *Soft Comput* 22:1687–1710
- Zhang R, He Z, Wang H, You F, Li K (2017) Study on self-tuning tyre friction control for developing main-servo loop integrated chassis control system. *IEEE Access* 5:6649–6660

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