DESIGN DEEP LEARNING NEURAL NETWORK FOR STRUCTURAL HEALTH MONITORING

[1]REZA RAHUTOMO, [2]FERGYANTO E. GUNAWAN

[1] Information Systems Management Department, BINUS Graduate Program - Master of Information Systems Management, Bina Nusantara University, Jakarta, Indonesia 11480

[2]Industrial Engineering Department, BINUS Graduate Program - Master of Industrial Engineering, Bina Nusantara University, Jakarta, Indonesia 11480

E-mail: [1]reza.rahutomo@binus.edu, [2]fgunawan@binus.edu

ABSTRACT

Bridge structural failure happens as the lack of monitoring. The existence of bridge structural health monitoring system is necessary for bridge maintenance due to its ability to process data and provide the information of structural health level. This research is performed to design a deep neural network model for classifying structural integrity with high accuracy. The model requires input data in the form of F-statistic, which is derived from structural vibration data. In the current approach, the vibration data are obtained from numerical analysis by means of the finite element methods. As much as 17.493 vibration cases are generated for five levels of structural integrity, namely, healthy conditions and conditions of 1%, 5%, 10%, 20% damage level. The neural network model consists of one input layer of 20 neurons, six hidden layers with 12 neurons per layer, and one output layer of 5 neurons. The model is trained by using Adam optimizer. The results show that the model is able to accurately classify the structural damage at 83.3% accuracy, and the majority of the false predictions occur in differentiating the healthy structural condition from those of 1% damage.

Keywords: *Artificial Neural Network, Deep Learning, Structural Health Monitoring, Vibration-based, Classification Accuracy*

1. INTRODUCTION

In 2011, the fall of Kutai Kartanegara Bridge that crossing one of the biggest rivers in Indonesia shocked not only the public, but also the engineering communities in Indonesia. The incident became the momentum to adopt new technology in order to maintain and prevent structural failure. Today, Structural Health Monitoring (SHM) system is installed in one of the iconic bridges in Indonesia, the Suramadu Bridge[1]. The SHM system has main function to monitor a structure and detecting damages. By using the functions included in the system, safety can be increased and costs of maintenance can be reduced. So, SHM system plays a major role not only in scope of computer science, but also in the civil engineering perspective [2].

To collect data from actual condition, the usage of wireless sensor device like ITS400 sensor board is commonly used. The device is able to record various variables like temperature and deflection [3]. But, unfortunately, those variables are not accurate for predicting the structural integrity. Uncontrolled vibration may possess a great danger to structural integrity but, on the other hand, vibration is also useful to be a diagnostic variable [4]. Deep learning is a state-of-the-art from machine learning which allow computer to learn representations of data with multiple levels of abstraction [5]. This research is dedicated to make a design of Deep Learning Neural Network architecture related to SHM as exemplified by reference [6].

2. PREVIOUS WORKS

The usage of artificial neural network (ANN) in processing vibration data in structural health monitoring has been questioned whether it's capable to classify in higher accuracy or not. Reference [7] presented numerous vibration-based condition monitoring methods that already applied in SHM, namely Statistical Methods [8], Wavelet Transform Methods [9][10][11], and SVM [12][13][14][15].

In this part, we will discuss findings from previous related works. In general, the research in

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

relevant domain aims to monitor the health level of a civil construction. There are many kinds of ANN architecture have been implemented to calculate precise classification.

Bakhary et al. proved that if an ANN architecture implemented in SHM system, damage detection in a civil structure is possible to do with utilization of censor devices. With crack detection function as the output, the research describes the design of two-level ANN which consists of input layer, hidden layer, and output layer for every level [15] [16].

A research from Mehrjoo et al. stated that from time to time, many researches enhance the ability of ANN to handle more complex tasks. The research explained that damage detection of truss bridge joints is one of the contributions in the usage of ANN for civil engineering. By utilizing back-propagation neural network, the damage detection only had the average error of 1% in five modes of experiment [18].

On the other hand, a research from Nazarko & Ziemiański modernized the standard of ANN implementation in SHM. The combination between ANN with Mean Squared Error contributes in novelty detection and damage evaluation [19].

3. RESEARCH METHODOLOGY

This research is dedicated to design a deep learning ANN architecture to learn vibration pattern and predict the structural health category. To build a classification model using deep learning neural network, this research adopted the four steps from reference [20], namely, data preprocessing, learning or training the ANN model, evaluation, and finally, prediction. These steps are illustrated in Figure 1.

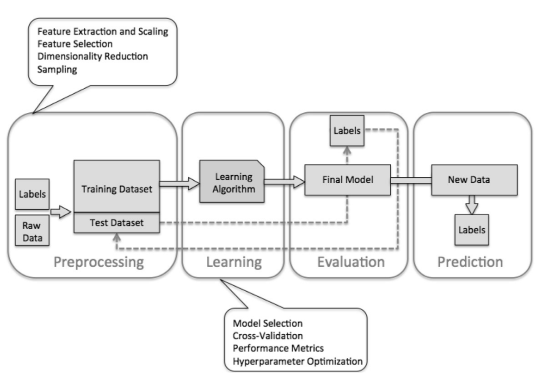

Figure 1. Four steps in designing ANN architecture [20].

3.1 Data Collection

To acquire the dataset for this research, the data are produced by a numerical analysis of a 7-degree of freedom system, which is similar to that studied by Ref. [21]. The system is illustrated in Figure 2.

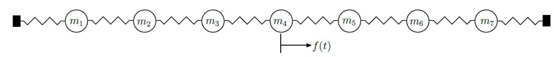

Figure 2. The model of seven-degree-of-freedom system studied in the current research.

The system consists of seven lumped masses and eight linear elastic springs. Each mass is 1 kg. Each spring has a stiffness of 1 N/m. The both ends of the system are fixed.

A dynamic force, f(t), having a random magnitude is applied to the center mass, m4. The force magnitude is drawn from a normal probabilistic distribution function with a mean of zero and a standard deviation of 0.09. Initially, the random force data have frequency contents up to 25 Hz. Then, the data are filtered with a Butterworth filter with a cutoff frequency of 20 Hz and an order of 12.

The structural damage is assumed to occur on the spring connecting m3 and m4. Furthermore, the damage is also assumed to affect the spring and to degrade its stiffness only. In the current study, four levels of the degradation are studied, namely, 1%, 5%, 10%, and 20%. This decision is made to understand how the damage level affects the accuracy of the classification of the structural integrity. We hypothesize that the classification accuracy is low when the damage level is low. We also hypothesize that the relation between the classification accuracy and the damage level is not linear. When the damage level is higher than certain threshold, the classification accuracy is expected to be independent of the damage level.

The numerical analysis is performed by using the commercial finite element package LS-Dyna. The analysis results are the displacement data of the seven masses. The data are sampled at a constant rate of 0.1 s and for a duration of 360 s. For each structural condition, the analysis is repeated for 500 times by varying the distribution of the dynamic force. This approach allows us to produce a large dataset where the damage classification methods can be studied extensively.

In this research, F-stat is used as the damage-sensitive feature. Its computation requires the power spectrum density data, which are

Journal of Theoretical and Applied Information Technology
15th March 2019. Vol.97. No 5

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

computed by the following procedure by using Barlett's method [22].

We consider an analog, time-varying, and nite-length signal xa(t). In the structural health monitoring, the signal may represent the historical data of the displacement at an observation point. The signal is assumed to be measured at a constant sampling rate of ts such that

$$x_i = x_a(i\,.t_s) \quad (1)$$

where $i = 0, 1, 2, \ldots, (N-1)$.

We transform the discrete time-domain signal x_i into the frequency domain by applying the discrete Fourier transform with the formula:

$$X(f_k) \;= \sum_{i=0}^{N-1} x_i \,.\, \exp(-ji2\pi f t_s) \quad (2)$$

Where $f \in [0, \frac{f_s}{2}]$ and $f_s = 1/t_s$, which is called the sampling frequency, and f_k are discrete frequencies of $f_i = i\,.\frac{f_s}{N}$. To shorten the expression, we use the symbol x_i to denote $x(f_i)$. We partition the signal intoM-equal-length subsignals as illustrated by Figure 3.

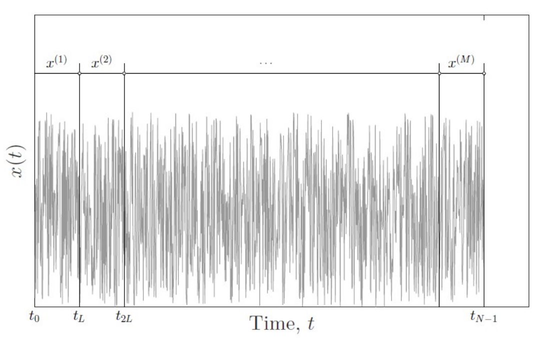

Figure 3. The partition of the signal x(t) into M-equal-length sub signals where each sub signal has a length of L.

Barlett's method computes a signal PSD by averaging PSDs of the sub signals. The resulted PSD is more reliable and less sensitive to the signal noises. However, the method is only applicable for long signals. The Barlett's formula for computing PSD is:

$$s_i(f) = \frac{1}{LM}\sum_{m=0}^{M-1}\left|X_i^{(m)}\right|^2 \quad (3)$$

The signal length N and the number of sub signals M is related by $N = LM$, where L is the length of the sub signal.

The F statistic, in conjunction with a simple classification method, had been used for damage detection. The method was presented theoretically by Ref. [23] and was experimentally verified by Ref. [24].

The method is simple and practical. It depends only on the data of structural responses, which can be collected on a few measurement points. The method turns the damage monitoring problem into a directionless hypothesis test problem that can be solved in three steps.

The first is the statement of the null and alternative hypotheses, which for this case, are:

$$H_0: \quad (4)$$
$$S_h(\omega) = S_u(\omega)$$
$$H_1: \quad (5)$$
$$S_h(\omega) \neq S_u(\omega)$$

The symbols $S_h(\omega)$ denotes the structural power spectral density (PSD). The subscripts h denotes the healthy condition. The subscript u denotes the unknown-to-be-sought condition.

The healthy condition is the reference condition, from which the other structural conditions are measured. It should be determined previously. The structure is assumed healthy if the null hypothesis H_0 prevails. It is considered healthy if its PSD are very much similar to the PSD of healthy condition. The degree of the similarity is measured statistically. The structure is assumed to contain damage if the alternative hypothesis Ha prevails, that is the PSD have changed significantly. Simply speaking, the structure that associated with $S_u(\omega)$ is considered damaged if $S_u(\omega)$ deviates significantly from $S_h(\omega)$.

The second step is to compute the relevant F-statistic. This statistic is simply a comparison of two PSDs: $S_h(\omega)$ and $S_u(\omega)$.The statistic has the value of 1.0 when the two PSDs are exactly identical. When the structure contains damages, some values of the F statistic may deviate from 1 becoming very big or very small. The level of change in PSD determines the magnitude of the F statistic.

The statistic is computed by:

$$F = \frac{\hat{S}_h(\omega)/S_h(\omega)}{\hat{S}_u(\omega)/S_u(\omega)} \quad (6)$$

The hat denotes the estimated PSD. This expression can be made simpler. Under the condition of (4), Equation (4) can be simplified to

$$F = \frac{\hat{S}_h(\omega)}{\hat{S}_u(\omega)} \quad (7)$$

The third is to establish the upper and lower limits of the statistic from which the change of PSD can be categorized as significant or not. The lower limit is $F_{(1-\alpha/2,2K,2K)}$ and the upper limit is $F_{(1-\alpha/2,2K,2K)}$. The symbol K denotes the statistical significance, which represents the probability of rejecting the null hypothesis given the structure condition is healthy. The symbol K denotes the degree of freedom, which representing the number of windows in the Barlett's method.

The expression similar to Equation 7, as discussed by Ref. [25], is extremely sensitive to perturbation and produces highly actuating result, often exceeds the lower and upper limits on the healthy structural condition. Thus, the F statistic is unreliable and must be computed with a great care.

Before the research started, cases of vibration frequency classification to health category must be collected. The Data set is collected by using the Finite Element Analysis tool from the simulation software called ANSYS®. The tool allows user to create simulation of any structure that responds to a phenomenon [26].

3.2 Pre-processing

Data Preprocessing means preparation before the data is processed. In the big picture, the obtained raw data will be extracted meaningful features which correlate to the remaining project. The obtained raw data somehow come up with unstructured and various features, the quality of the data itself probably will bring us unexpected result. To anticipate that, a group of activities namely data preprocessing that covers random-shuffling, selection, extraction, and splitting applied to get the raw data into shape.

As this research using programming language called Python, all of the activities were done by Python. With Python library named Numpy, the pre-processing step can be run. At first, the dataset must be translated to the form of Numpy Array. The pre-processing step followed by random-shuffle the sequence of the dataset, so in the learning phase, the ANN architecture will learn the characteristic of data from all categories. After that, the feature and label must be separated as feather and label will annotate like X to Y.

Feature scaling techniques will be implemented in this research. MinMax Scaling or widely known as Normalization that will simplify the range between 0 to 1 as the value will earned from the division between the original value subtracted by the minimum value and the maximum value subtracted by the minimum value. The formula explained by equation 8. It will help the algorithm the accelerate learning the pattern out of the data set.

$$x^{(i)}_{norm} = \frac{x^{(i)} - x_{min}}{x_{max} - x_{min}} \quad (8)$$

The following feature scaling technique to be applied is Standardization. It will subtract every single value to mean value and divided with standard deviation. The formula is mentioned as equation 9:

$$x^{(i)}_{std} = \frac{x^{(i)} - \mu}{\sigma} \quad (9)$$

3.3 Learning

The learning phase is about giving computer the ability to learn from training data set. The learning phase become important due to the Artificial Neural Network (ANN) creation in this phase defines the whole research succession. An ANN architecture will be designed and an activation function will be defined on it to make the learning phase works properly.

There are several elements needed to design the Artificial Neural Network (ANN) architecture. Basically, ANN is Machine Learning model inspired by biological neural network. It consists of layers of nodes to process values and weights. Three main elements in ANN are Input Layer, Hidden Layer, Output Layer, and Activation Function. Each layer contains nodes, and to define how many nodes required for each layer we have to follow several rules based on Karsoliya's research [27]. The number of neurons in hidden layer is 70% - 90% of the number of neurons in input layer, or the number of neurons in hidden layer must be less than twice the number of neurons in input layer.

ANN implementation in an information system will establish computing ability to learn and observe data in order to solve a problem [28]. The basic idea of ANN is Artificial Neuron. The concept of Artificial Neuron is illustrated as Figure 4.

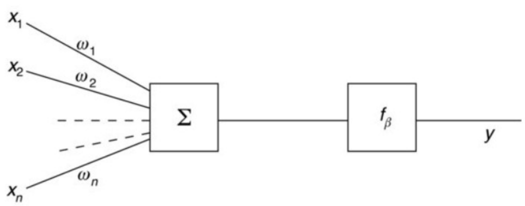

Figure 4. Artificial neuron architecture

Based on the last graph comes the following equation which described neuron x_n with weight w_n accumulated and activated with activation function f_β. Equation 10 describes the activation function.

$$f\beta = \sum_{i=1}^{n} w_m x_m \qquad (10)$$

There are numerous of algorithms to allow machines learn datasets. In this research, Supervised Learning is used as the training algorithm in order to create a model that able to learn the pattern inside the data and use it to solve the problem. Supervised Learning is a learning algorithm which guide the learning process with separating the feature and label. The ANN itself runs in Deep Learning architecture, which contains of more than one hidden layer and required a bias neuron.

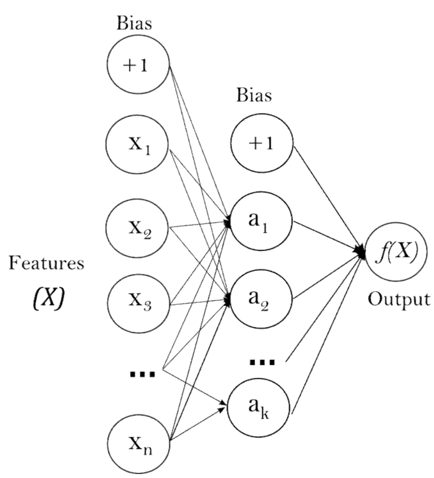

Figure 5. Deep learning neural network architecture

To learn all cases acquired from Training dataset, an activation function is needed to renew the result from each learning cycle or epoch. Reference [29] recommends to utilized ReLU and Softmax. It is explained that the best activation function for classification is ReLu [30] on hidden layer which contain a simple function of $y = max(0, x)$ and Softmax [31] on output layer for computing the probabilities of the class labels.

The specification of the Deep Learning Neural Network contains 20 nodes in Input Layers as the input is a 20-column dataset. According to reference [27], there are several ways to define the number of neurons in the architecture of ANN.

1. The number of neurons in hidden layer is 2/3 or 70% - 90% from the number of neurons in input layer.
2. The number of neurons in hidden layer must less than twice the number of neurons in input layer.
3. The number of neurons in hidden layer is between the number of neurons in input layer and output layer.

3.4 Evaluation

In Evaluation phase, the ANN architecture is measured, optimized, and evaluated using the technique called hyper parameter tuning. Hyper parameter tuning is a fine-tuning method to set the parameter weight to produce the best performance without overfitting and underfitting. The technique played a major role to be a successor by delivering a good score of validation accuracy. In some projects, the evaluation phase run alongside the learning phase. We can tune the ANN architecture run the phases together. Dropout is utilized to improve the performance of the ANN architecture to prevent overfitting. Dropout will affecting the architecture by making neurons in hidden layers paralyzed or unreliable [32].

3.5 Prediction

The final phase of this research intended predicted result as the outcome. The predicted result shown probability of every labels on every case. Each label in one single case have values indicating its prediction probability. The one who get the highest value, mostly the closest to 1, is the guessed. To check the points of accuracy and loss function, confusion matrix is used to view the accuracy and errors from the prediction phase. The output of the phase is accuracy and error analysis in the form of Confusion Matrix. It shows the correlation between all category labels in test dataset and the prediction result. The correct prediction is shown diagonally, where the rest of the score are the incorrect ones [33].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4. RESULT & DISCUSSION

This section will deliver results from each section and discuss the result comparing to the latest result in the same research domain.

4.1 Data Collection

In this research, data architecture must contain 2 parts namely features and labels. Features and labels will be annotated and the accuracy will be measured in the end. Features are the frequency of vibration that recorded by 10 wireless sensor devices and out-print the maximum and minimum value out of it. The labels have 5 categories of structural health level, there are: 100%, 99%, 95%, 90% and 80%.

Table 1. Data structure generated from Ansys

No	Column Name	Description
1	F_{max} 1	F_{max} value recorded by Sensor 1
2	F_{max} 2	F_{max} value recorded by Sensor 2
3	F_{max} 3	F_{max} value recorded by Sensor 3
4	F_{max} 4	F_{max} value recorded by Sensor 4
5	F_{max} 5	F_{max} value recorded by Sensor 5
6	F_{max} 6	F_{max} value recorded by Sensor 6
7	F_{max} 7	F_{max} value recorded by Sensor 7
8	F_{max} 8	F_{max} value recorded by Sensor 8
9	F_{max} 9	F_{max} value recorded by Sensor 9
10	F_{max} 10	F_{max} value recorded by Sensor 10
11	F_{min} 1	F_{min} value recorded by Sensor 1
12	F_{min} 2	F_{min} value recorded by Sensor 2
13	F_{min} 3	F_{min} value recorded by Sensor 3
14	F_{min} 4	F_{min} value recorded by Sensor 4
15	F_{min} 5	F_{min} value recorded by Sensor 5
16	F_{min} 6	F_{min} value recorded by Sensor 6
17	F_{min} 7	F_{min} value recorded by Sensor 7
18	F_{min} 8	F_{min} value recorded by Sensor 8
19	F_{min} 9	F_{min} value recorded by Sensor 9
20	F_{min} 10	F_{min} value recorded by Sensor 10
21	Category 1	Structure health: 100%
22	Category 2	Structure health: 99%
23	Category 3	Structure health: 95%
24	Category 4	Structure health: 90%
25	Category 5	Structure health: 80%

The tool Finite Element Analysis from the software ANSYS® successfully generated at least 17.493 cases of how structure health monitored by analyzing frequency of vibrations. Every case consists of 25 columns where the first 10 columns are F_{max} value, and the following 10 columns are F_{min} value, and the last 5 are structural health categories recorded in the form of one-hot encoding. Table 1 describes the structure of the collected data.

4.2 Pre-processing

The data preprocessing started by formatting the raw data into csv. Following the process, randomization is occurred if the data set come up sequentially as explained in Table 2. Randomization is basically an optional step, in order to gain high accuracy prediction, the use of randomization is recommended. In some certain programming language like Python, doing randomization is simply as calling the random shuffle function from the library NumPy to the variable containing data set.

Table 2. The original order of the dataset

Name	Order
Category 1	1 - 3493
Category 2	3493 - 6993
Category 3	6994 - 10493
Category 4	10494 - 13993
Category 5	13994 - 17493

On the other side, features and labels must be separated in order to make the data set usable for classification. The selection continues with separating all 20 columns of features to another new variable and the rest 5 columns of labels to another one. In the end, we have 2 variables which contains features and labels.

Table 3. The original value ranges

Name	Maximum Value	Minimum Value
F_{max} Sensor 1	2111.329	1.348
F_{max} Sensor 2	231.789	1.328
F_{max} Sensor 3	89.761	1.256
F_{max} Sensor 4	41.527	1.224
F_{max} Sensor 5	20.342	1.197
F_{max} Sensor 6	11.755	1.186
F_{max} Sensor 7	10.924	1.175
F_{max} Sensor 8	7.706	1.138
F_{max} Sensor 9	6.887	1.109
F_{max} Sensor 10	6.72	1.109
F_{min} Sensor 1	0.59	0.000153
F_{min} Sensor 2	0.634	0.00019
F_{min} Sensor 3	0.676	0.000357
F_{min} Sensor 4	0.739	0.0014
F_{min} Sensor 5	0.784	0.00223
F_{min} Sensor 6	0.791	0.0053
F_{min} Sensor 7	0.81	0.0094
F_{min} Sensor 8	0.816	0.02
F_{min} Sensor 9	0.851	0.02
F_{min} Sensor 10	0.865	0.03

The range of the data is too varied as shown in Table 3, and if the dataset is used in the following phase it will not resulting a high accuracy prediction. To optimize the learning algorithm in the learning phase, the value range of the features must be simplified using feature scaling method named normalization.

The equation is explaining how to normalize the value range. The normalized value is the result of one value reduced by the minimum value divided by maximum value reduced by the minimum value. The normalization followed by standardization which Normalized value is the result from a value reduced by mean value and divided by standard deviation. When Feature Scaling applied to every feature value, the range will be simplified and it will be ready to be processed on the next phase. Table 4 represents the rescaled version of the value ranges.

Finally, the dataset must be divided into several groups namely 'Training Dataset', 'Validation Dataset' and 'Testing Dataset'. The percentage of Training data set is 60%, meanwhile the percentage of both Validation and Test data set are 20%. To achieve the best percentage, the total number of data and the learning model we train are the things to consider. There are various of learning models that highly depending on the hyper parameters. Models with few hyper parameters will be easy to validate and tune, but the more hyper parameter, the larger number of validation data set is required.

Table 4. The value ranges after feature rescaling

Name	Maximum Value	Minimum Value
F_{max} Sensor 1	0.934	-0.065
F_{max} Sensor 2	0.939	-0.060
F_{max} Sensor 3	0.931	-0.068
F_{max} Sensor 4	0.911	-0.088
F_{max} Sensor 5	0.892	-0.107
F_{max} Sensor 6	0.847	-0.152
F_{max} Sensor 7	0.859	-0.140
F_{max} Sensor 8	0.813	-0.186
F_{max} Sensor 9	0.808	-0.191
F_{max} Sensor 10	0.825	-0.174
F_{min} Sensor 1	0.793	-0.206
F_{min} Sensor 2	0.771	-0.228
F_{min} Sensor 3	0.734	-0.265
F_{min} Sensor 4	0.695	-0.304
F_{min} Sensor 5	0.668	-0.331
F_{min} Sensor 6	0.633	-0.366
F_{min} Sensor 7	0.607	-0.392
F_{min} Sensor 8	0.581	-0.418
F_{min} Sensor 9	0.571	-0.428
F_{min} Sensor 10	0.552	-0.447

4.3 Learning

Several compositions of nodes in the hidden layer were tried and that describes in the Table 5. We found that 17 neurons give the best performance in developing training accuracy. The research was continued by testing the training accuracy development by adding hidden layers with 17 neurons each. The result that recorded in Table 6 proved that the number of hidden layers is influencing the performance. With 3 hidden layers with 17 neurons on every layer, the training accuracy is successfully improved.

Table 5. How the number of neurons affecting training accuracy

Number of Neurons	Training Accuracy
16	85.24 %
17	86.52 %
18	84.24 %

Table 6. How the number of hidden layers affecting training accuracy

Number of Hidden Layers	Training Accuracy
2	83.15 %
3	87.73 %
4	86.52 %

Figure 6 represents the visualization of the Deep Learning Neural Network architecture. On the figure, we can see all the nodes are connected each other and make matrix computation [34]. The architecture was built using TensorFlow wrapped in the framework called Keras for the simpler usage.

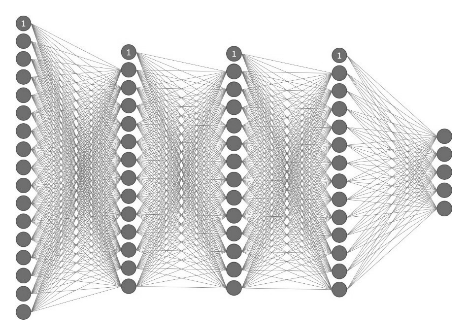

Figure 6. Deep Learning Neural Network architecture

To complete the ANN architecture, we need activation function added to hidden layers and output layer for achieving continuous result from every epoch. From the recommendation in this reference, we add ReLu as activation function in Hidden Layers and Softmax as activation function in the Output Layer.

4.4 Evaluation

From table 7 and 8, we learned that the learning rate and dropout affecting the validation accuracy of the ANN architecture. The ideal weight of both Dropout and Learning rate is 0.001, and if we include the hyper parameter tuning result to the specification of the ANN architecture, it will be seen as follows:

- Dropout: 0.001
- Learning Rate: 0.001
- Epoch: 150
- Loss Function: Categorical Cross Entropy
- Batch size: 128

Table 7. How Learning Rate affecting validation accuracy

Learning Rate	Validation Accuracy
0.0001	81.56 %
0.001	84.04 %
0.01	82.53 %

Table 8. How Dropout affecting validation accuracy

Dropout	Validation Accuracy
0.0001	82.36 %
0.001	84.21 %
0.01	82.53 %

In the end of this phase, we plot the learning accuracy and the validation accuracy. Figure 7 illustrates how the evaluation accuracy matches the learning phase accuracy in 150 epochs. It can be seen both the accuracy of learning and evaluation phase are suddenly peaking up close to 0.8 or 80% at the early start. What happened on the learning accuracy and the evaluation accuracy is similar. Both the graphs are gradually increasing at the same time with a slight of difference. Overall, the accuracy in both phases reached 80% and still increasing.

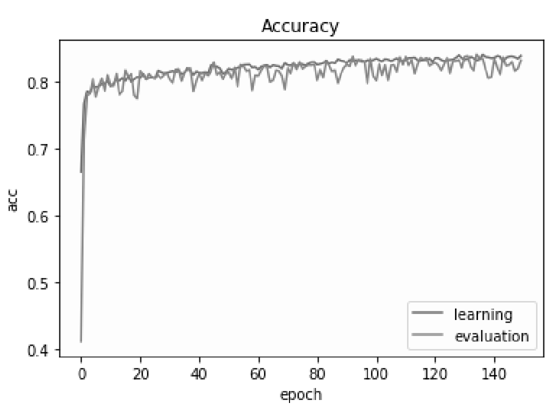

Figure 7. The accuracy of learning phase and evaluation phase

Figure 8. represents how the loss function in the learning phase and evaluation phase matches together. Both loss function dropped in the early start down to nearly 0.4 or 40%. The similarity from both loss functions indicating the difference and likely are exist.

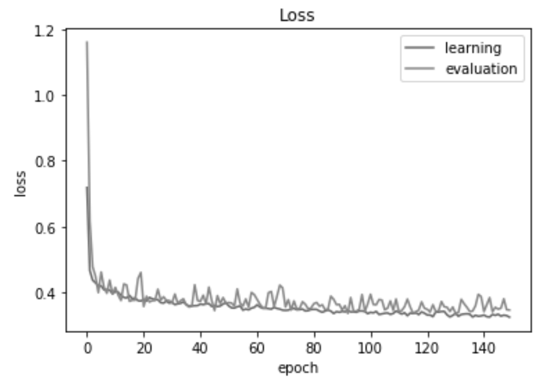

Figure 8. The loss function from learning phase and evaluation phase

4.5 Testing

The model creates the prediction out of the test data set. It comes out 83.3% accuracy when predict the numerous cases in test data set. On the other hand, several errors are still occurred when the process being held. Table 5 indicates errors and accuracy from prediction process.

The confusion matrix shown 83.3% accuracy. The number generated from the accumulation of the correct predictions divided by the accumulation of all prediction results. From table 5, we can indicate Category 5 is the easiest category to recognize as its precision reach 100%, while Category 1 is the hardest condition to predict as the difference is less of be equal to 1% for each category.

Table 5. Confusion Matrix

Category		Actual Data					Precision
		1	2	3	4	5	
Prediction	1	408	284	1	0	0	58.8%
	2	161	550	9	0	0	76.3%
	3	3	49	615	26	0	88.7%
	4	0	0	50	660	0	92.9%
	5	0	0	0	0	685	100%
Recall		71.3%	62.2%	91.1%	96.2%	100%	83.3%

4.6 Discussion

The ANN architecture for structural health monitoring successfully designed with 83.3% of testing accuracy. Although the dataset includes balance amount of every structural health categories, the similarity of category 1 and category

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2 raise a bottleneck in distinguishing the characteristic of the vibrations. In other words, Deep Learning Neural Network is having difficulty in annotating features to label of category 1 and category 2 due to its similarity recorded in the features, meanwhile the uniqueness of category 5 gives an outstanding result of 100% precision and recall. Some options are needed to mitigate the misclassification of category 1 and 2. We can either add more data for category 1 and 2, or create new categorizations with vibrant structural health categories.

To be fair, we compare our research to other similar research. Using machine learning model SVM, the testing accuracy reached 81.5% [12]. Through this research, the discovery of designing ANN for structural health monitoring has become the state-of-the-art in AI for civil engineering.

5. CONCLUSION

Through Structural Health Monitoring, we can afford detecting damages and monitor the load-handling capability of a structure. With structural health monitoring, we could get benefits such as increased safety, increased risk mitigation, longer structural life time, and cost-efficient maintenance. As the state-of-the-art in the evolution of computational systems, implementing the Artificial Neural Network to calculate structural healthy level integrity is necessary. Some research proved that ANN model gives more accurate prediction rather than any predecessors.

In this research, we proved that ANN gained higher accuracy than the previous learning model in predicting structural integrity from structural vibration. We designed the ANN architecture using four-step activity: Preprocessing, Learning, Evaluation, and Prediction. We started the activity with collecting the data, standardizing the data in the preprocessing step, develop the ANN architecture in the learning step with utilizing the training dataset to gain the training accuracy, tune the ANN performance in the evaluation step with utilizing the validation dataset to gain validation accuracy, and finalize the research with testing with utilizing the testing dataset.

More research is needed to contribute and add more improvement. The most important improvement is increasing the accuracy of the ANN in predicting cases with adding more training data. The other improvement which from the civil engineering side is formulating the following action based on the prediction. So, structural health monitoring is not only viewed as a standard operational procedure in maintaining structural integrity, but also a system which supported by the state-of-the-art in examining data into high accuracy prediction.

6. ACKNOWLEDGEMENT

This research was finished under support from BINUS – NVIDIA Artificial Intelligence Research and Development Center in BINUS University Anggrek Campus, Jakarta, Indonesia. Special thanks to Mr. Tjeng Wawan Cenggoro for the useful discussion in designing neural network and managing the server used in this research. All of the research phases were supported using NVIDIA GPU Tesla P4 and P100 interchangeably.

REFERENCES:

[1] P. Suprobo and F. A. Faimun, "Infrastructure Health Monitoring System (SHM) Development, a Necessity for Maintenance and Investigation," *Int. J. Eng. Technol. Dev.*, pp. 68–71, 2013.

[2] F. G. Baptista, D. E. Budoya, V. A. D. Almeida, and J. A. C. Ulson, "An Experimental Study on the Effect of Temperature on Piezoelectric Sensors for Impedance-Based Structural Health Monitoring," *Sensors*, vol. 14, no. 1, pp. 1208–1227, 2014.

[3] G. Hackmann, W. Guo, G. Yan, Z. Sun, C. Lu, and S. Dyke, "Cyber-Physical Codesign of Distributed Structural Health Monitoring with Wireless Sensor Networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 63–72, 2014.

[4] D. Goyal and B. S. Pabla, "The Vibration Monitoring Methods and Signal Processing Techniques for Structural Health Monitoring: A Review," *Arch. Comput. Methods Eng.*, vol. 23, no. 4, pp. 585–594, 2016.

[5] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[6] D. Belanger and A. McCallum, "Structured Prediction Energy Networks," *Proc. 33rd Int. Conf. Mach. Learn.*, vol. 48, pp. 983–992, 2016.

[7] E. P. Carden and P. Fanning, "Vibration based condition monitoring: A review," *Struct. Heal. Monit.*, vol. 3, no. 4, pp. 355–377, 2004.

[8] C. R. Farrar and S. W. Doebling, "Damage Detection II: Field Applications to Large Structures," *MS P946, Los Alamos Natl. Lab.*, pp. 1–29, 1998.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[9] K. M. Liew and Q. Wang, "Application of Wavelet Theory for Crack Identification in Structures," *J. Eng. Mech.*, vol. 124, no. 2, 1998.
[10] C.-J. Lu and Y.-T. Hsu, "Vibration Analysis if an Inhomogeneous Dtring for Damage Detection by Wavelet Transform," *Int. J. Mech. Sci.*, vol. 44, no. 4, pp. 745–754, 2002.
[11] Z. Hou, M. Noori, and R. St. Amand, "Wavelet-Based Approach for Structural Damage Detection," *J. Eng. Mech.*, vol. 7, no. 126, 2000.
[12] F. E. Gunawan, "Improving The Prediction Reliability of F-Statistic Method by using Support Vector Machine for Structural Health Monitoring," *Unpubl. Manuscr.*
[13] K. Worden and G. Manson, "The application of machine learning to structural health monitoring," *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, vol. 365, no. 1851, pp. 515–537, 2007.
[14] L. Bornn, C. R. Farrar, G. Park, and K. Farinholt, "Structural Health Monitoring With Autoregressive Support Vector Machines," *J. Vib. Acoust.*, vol. 131, no. 2, p. 021004, 2009.
[15] R. Ghiasi, P. Torkzadeh, and M. Noori, "A machine-learning approach for structural damage detection using least square support vector machine based on a new combinational kernel function," *Struct. Heal. Monit.*, vol. 15, no. 3, pp. 302–316, 2016.
[16] N. Bakhary, H. Hao, and A. J. Deeks, "Structure Damage Detection Using Neural Network with Multi-Stage Substructuring," *Adv. Struct. Eng.*, vol. 13, no. 1, 2009.
[17] N. Bakhary, H. Hao, and A. J. Deeks, "Damage detection using artificial neural network with consideration of uncertainties," *Eng. Struct.*, vol. 29, no. 11, pp. 2806–2815, 2007.
[18] M. Mehrjoo, N. Khaji, H. Moharrami, and A. Bahreininejad, "Damage detection of truss bridge joints using Artificial Neural Networks," *Expert Syst. Appl.*, vol. 35, no. 3, pp. 1122–1131, 2008.
[19] P. Nazarko and L. Ziemiański, "Application of Artificial Neural Networks in the Damage Identification of Structural Elements," *Comput. Assist. Mech. Eng. Sci.*, vol. 18, pp. 175–189, 2011.
[20] S. Raschka, *Python Machine Learning*. Packt Publishing Ltd., 2015.
[21] C. R. Farrar and K. Worden, *Structural Health Monitoring: A Machine Learning Perspective*. John Willey and sons, 2012.
[22] R. J. Schilling and S. L. Harris, *Fundamentals on Digital Signal Procesing Using MATLAB*. Cengage Learning, 2011.
[23] S. D. Fassois and J. S. Sakellariou, "Statistical Time Series Methods for SHM," *Encyclopedia of Structural Health Monitoring*. John Wiley & Sons, pp. 443–472, 2009.
[24] F. P. Kopsaftopoulus and S. D. Fassois, "Vibration based health monitoring for a lightweight truss structure: Experimental assessment of several statistical time series methods," *Mech. Syst. Signal Process.*, vol. 24, no. 7, pp. 1977–1997, 2010.
[25] F. E. Gunawan, "Impact force reconstruction using the regularized Wiener filter method," *Inverse Probl. Sci. Eng.*, vol. 24, no. 7, pp. 1107–1132, 2016.
[26] E. Madenci and I. Guven, *The Finite Element Method and Applications in Engineering Using ANSYS®*. Springer, 2015.
[27] S. Karsoliya, "Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture," *Int. J. Eng. Trends Technol.*, vol. 3, no. 6, pp. 714–717, 2012.
[28] X. Qiu, L. Zhang, Y. Ren, P. N. Suganthan, and G. Amaratunga, "Ensemble deep learning for regression and time series forecasting," *2014 IEEE Symp. Compputational Intell. Ensemble Learn.*, 2014.
[29] A. Géron, *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Inc., 2017.
[30] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," *Icml '13*, vol. 28, p. 6, 2013.
[31] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," pp. 1–18, 2012.
[32] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.
[33] J. V. Haxby, A. C. Connolly, and J. S. Guntupalli, "Decoding Neural Represntational Spaces Using Multivariate Pattern Analysis," *Annu. Rev. Neurosci.*, no. 37, pp. 435–456, 2014.
[34] L. Rampasek and A. Goldenberg, "TensorFlow: Biology's Gateway to Deep Learning?," *Cell Syst.*, vol. 2, no. 1, pp. 12–14, 2016.