International Journal of Advanced Smart Convergence Vol.10 No.3 51-58 (2021)
http://dx.doi.org/10.7236/IJASC.2021.10.3.51

IJASC 21-3-7

Deep learning classifier for the number of layers in the subsurface structure

Ho-Chan Kim[1], Min-Jae Kang[2]

[1] *Professor, Department of Electrical Engineering, Jeju National University, Korea,*
[2] *Professor, Department of Electronic Engineering, Jeju National University, Korea,*
hckim@jejunu.ac.kr, minjk@jejunu.ac.kr

Abstract

In this paper, we propose a deep learning classifier for estimating the number of layers in the Earth's structure. When installing a grounding system, knowledge of the subsurface in the area is absolutely necessary. The subsurface structure can be modeled by the earth parameters. Knowing the exact number of layers can significantly reduce the amount of computation to estimate these parameters. The classifier consists of a feedforward neural network. Apparent resistivity curves were used to train the deep learning classifier. The apparent resistivity at 20 equally spaced log points in each curve are used as the features for the input of the deep learning classifier. Apparent resistivity curve data sets are collected either by theoretical calculations or by Wenner's measurement method. Deep learning classifiers are coded by Keras, an open source neural network library written in Python. This model has been shown to converge with close to 100% accuracy.

Keywords: *deep learning classifier, earth structure, grounding system, apparent resistivity, Wenner method, Keras.*

1. Introduction

The grounding construction site analysis mainly calculates the grounding resistance by simplifying the underground structure into a horizontal multi-layer structure. Horizontal multi-layer structures can be modeled by earth parameters such as the number of layers, the depth and resistance of each layer. Apparent earth resistivity data is needed to estimate the depth and resistance of each layer. Apparent earth resistivity can be measured and theoretically calculated. The most widely used method of measuring apparent earth resistivity is Wenner's four-electrode method.

Estimation of the earth parameters is accomplished by repeatedly modifying the earth parameters so that the theoretical calculated apparent resistivity is close to the measured value [2,3]. This results in an optimization problem that minimizes the squared value of the difference between the calculated and measured values as a function of cost. Various optimization problem solving methods are used to solve this problem, but the all the methods are similarly computationally expensive and actually takes a lot of time and effort. Because, in each iterative procedure, the earth apparent resistivity and their partial derivatives with respect to the earth parameters should be calculated. However, in each expression of both the earth apparent resistivity and their partial derivatives with respect to the earth parameters, there exists an improper integral that has not only an

Manuscript Received: July. 5, 2021 / Revised: July. 7, 2021 / Accepted: July. 13, 2021
Corresponding Author: minjk@jejunu.ac.kr
Tel: +82-64-754-3666, Fax: +82-64-756-1745
Professor, Department of Electronic Engineering, Jeju National University, Korea

infinite limit of integration but also a Bessel's function. To reduce the computation time, paper [2]–[6] appealed to numerical integration or infinite series expression. They are still inconvenient and arduous if accuracy should also be ensured.

So knowing the exact number of layers can significantly reduce the amount of computation [8]. In most earth parameter inversion methods, the number of layers is an arbitrary user-defined parameter, or it is determined through trial and-error by running the inversion many times using different numbers of layers and choosing the number of layers that produces the best model-data fit. Here, we provide a method that solves the problem of choosing the correct number of layers.

In this paper, we propose a deep learning classifier to determine the number of layers. The data set for training the deep learning classifier consists of 150 pieces of the apparent resistivity curves. The apparent resistivity at 20 points in each curve are used as the features for the input of the deep learning classifier. The validation split randomly split the data into use for training and validating. The 20% of the dataset we provide in the model is set aside for validating model performance. In the simulation, the deep learning model for the earth layer classifier is coded by Keras which is a user-friendly neural network library written in Python.

2. Wenner's Test and Apparent Soil Resistivity

Wenner configuration method is most widely used to measure the apparent resistivity as shown in Fig. 1. Wenner's method was published by Frank Wenner in 1915. In this method, to simplify the problem, it is assumed that the ground is composed of N layers horizontally, and each layer has the same resistivity [2]. Fig. 1 shows the electrode arrangement of Wenner's four-electrode method. In Wenner's method, the four electrodes are placed with the same distance a, and the voltage applied to the two internal electrodes(A, B) is measured when the current is supplied through two external electrodes(C, D). Assuming the depth of last layer is infinite, $h_i(i = 1, 2, ..., N\text{-}1)$ and ρ_i $(i = 1, 2, ..., N)$ represent the resistivity and the depth of each soil layer. The data measured with the more wide span a are used for analyzing the more deep earth structure. The measured apparent earth resistivity can be obtained as follows [2].

$$\rho_a = 2\pi a \frac{\Delta V}{I} = 2\pi a R \tag{1}$$

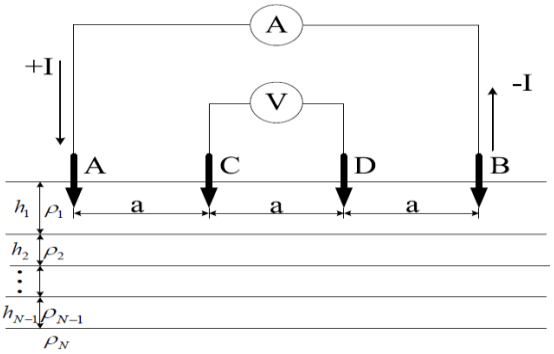


Figure 1. Wenner configuration for measuring apparent soil resistivity of N-layer earth structure.

By the inverse definition, if the earth parameters are known, the apparent resistivity can be by [2],

$$\rho_a = \rho_1\left[1 + 2a\int_0^\infty f(\lambda)[J_0(\lambda a) - J_0(2\lambda a)]d\lambda\right] \tag{2}$$

where ρ_1 is the soil resistivity of the first layer, a is electrode span, $J_0(\lambda r)$ is the zero order Bessel's function of the first kind, and

$$f(\lambda) = \alpha_1 - 1 \tag{3}$$

$$\alpha_1 = 1 + \frac{2K_1 e^{-2\lambda h_1}}{1 - K_1 e^{-2\lambda h_1}} \qquad k_1 = \frac{\rho_2\alpha_2 - \rho_1}{\rho_2\alpha_2 + \rho_1}$$

$$\alpha_2 = 1 + \frac{2K_2 e^{-2\lambda h_2}}{1 - K_2 e^{-2\lambda h_2}}, \qquad k_2 = \frac{\rho_3\alpha_3 - \rho_2}{\rho_3\alpha_3 + \rho_2}$$

$$\vdots$$

$$\alpha_{N-1} = 1 + \frac{2K_{N-1} e^{-2\lambda h_{N-1}}}{1 - 2K_{N-1} e^{-2\lambda h_{N-1}}}, \qquad k_{N-1} = \frac{\rho_N + \rho_{N-1}}{\rho_N + \rho_{N-1}}$$

$f(\lambda)$ is the kernel function. In the earth structure composed of N layers, the parameters are the depth of each layer, where h_i(i = 1, 2, ..., *N*-1) and resistivity ρ_i (i = 1, 2, ..., N) are the thickness and the resistivity of the *i*th layer respectively and *N* is the number of layers.

3. Deep learning Classifier for the number of layers

3. 1 Deep learning model for classifier

The architecture of the deep learning classifier is shown in Fig. 2. The classifier consists of three layers of the feedforward neural network (FNN): input, hidden, and output layer. The input layer has 20 nodes and the hidden layer has 128 nodes. The number of nodes in the input layer is determined by the number of features, but the number of nodes in the hidden layer can be hundreds or thousands. Increasing the number of nodes in the hidden layer increases the model capacity [7]. The features are selected from 20 points in the apparent resistivity curve. The last layer of our model has 3 nodes. One option is indicated for each number of layer: 2 layer, 3 layer or 4 layer. The activation function for the hidden layer is ReLU or Rectified Linear Activation. Although it is two linear pieces, it has been proven to work well in neural networks. The activation for the last layer is 'softmax'. Softmax makes the output sum up to 1 so the output can be interpreted as probabilities. The model will then make its prediction based on which option has a higher probability [8]. The dropout is used between the hidden and last layers to get the assemble effect.

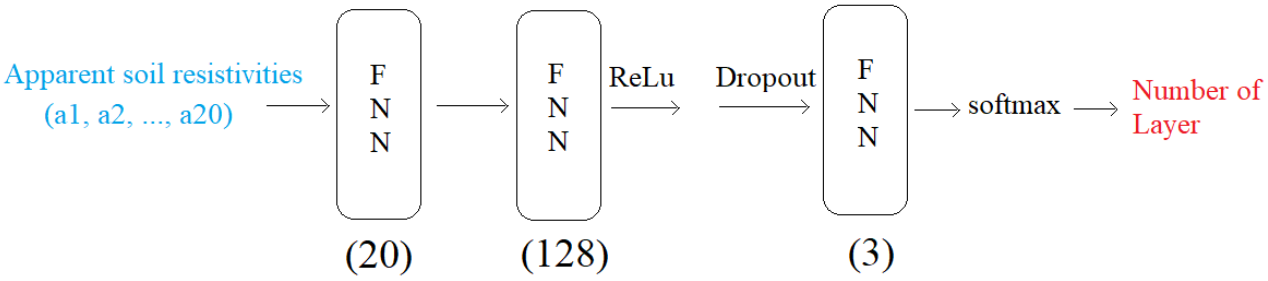


Figure 2. The architecture for deep learning classifier.

3. 2 Dataset

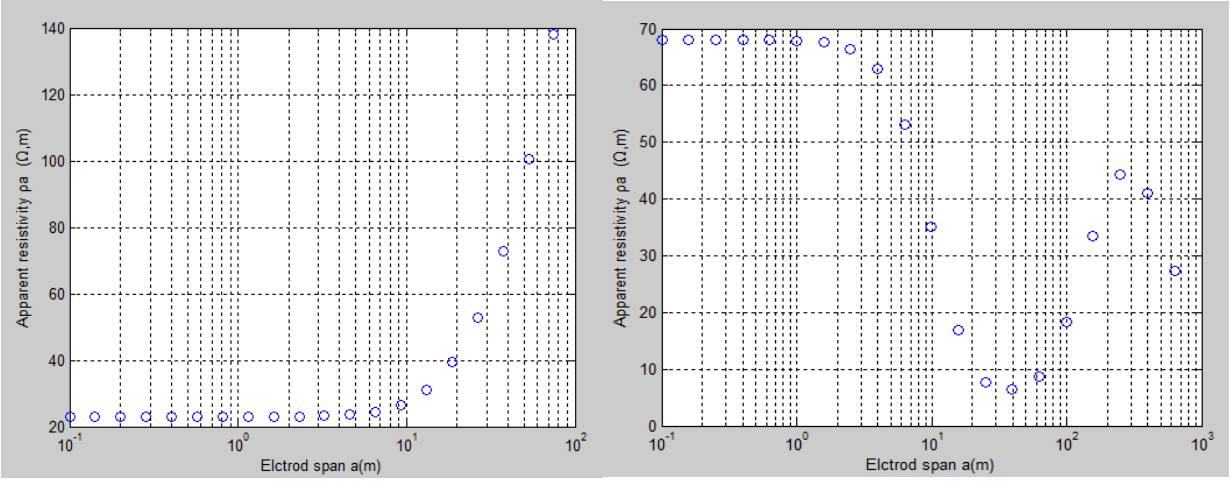
The data set consists of 150 apparent resistance curves. Three layers 2, 3 and 4 were selected to generate this data set. Because these 3 layers are actually the most applied. Each layer has 50 pieces of data. The data looks very simple in Table 1. Number of layers from left to right, depth and resistance of layers, and 20 points of apparent resistivity. The apparent resistivity is of field measurements and theoretically generated ones. To train a deep learning classifier, we use the number of layers as labels and 20 points on the apparent resistance curve as features.

For feature selection, the 20 points in the apparent resistivity curve (a1, a2, …, a20) are used as features as shown in Table 1. The 20 points are equally spaced logarithmic scales, as shown in Figure 3. The number of layers, the depth of the layers, and the resistivity are parameters of the subsurface structure. Knowing the number of layers is important before estimating other parameters. This is because knowing the number of layers makes it much easier to estimate other parameters. This is because the number of layers determines the number of parameters. As shown in Fig. 1, the resistivity number(ρ_i (i = 1, 2, ..., N)) in N-layer structure is equal to the number of layers and the number of thickness (h_i(i = 1, 2, ..., *N*-1)) is one less than the number of layers.

Normalization of data sets is a common requirement for many deep learning estimators to avoid features in greater numeric ranges dominating those in smaller numeric ranges meanwhile reducing the calculation expense. Each data feature is normalized by dividing by the maximum of data before input to the deep learning classifier [9].

Table 1. Dataset for earth structure and apparent resistivity.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Number of Layer	Depth h(m)			Resistivity ρi (Ω·m)				Apparent soil resistivity (Ω·m)					
2		h1	h2	h3	ρ1	ρ2	ρ3	ρ4	a1	a2	a3	.....	a19	a20
3	2	15	0	0	23	750	0	0	23	23	23	.....	100.5	138.1
4	3	1.2	10.6	0	164	71.6	203.7	0	161.15	157.17	145.1	.....	177.4	189.12
5	4	7.3	3.98	125.4	68	1.08	627.9	5.64	67.9	67.9	67.9	.....	44	31.3
6	3	10	5	0	352	110	210	0	351.99	351.94	351.85	.....	204.19	206.71
7	2	12	0	0	1023	32	0	0	1023	1022.9	1022.5	.....	57.435	60.717
8	2	4	0	0	28	400	0	0	28.041	28.11	28.231	.....	175.28	210.46
9	4	20	10	12	352	1100	210	2008	352	352.01	352.03	.....	1328.4	1609.6
10	3	3	8	0	122	26	900	0	121.75	121.34	119.64	.....	276.12	396.95
11	3	12	15	0	30	284	1005	0	30.002	30.006	30.021	.....	406.63	575.03
12	3	24	10	0	68	627	340	0	68	68.002	68.007	.....	279.19	308.53
13	3	13	20	0	17	125	500	0	17.001	17.003	17.011		233.51	323.69



(a) 2-layer earth structure. **(b) 4-layer earth structure.**

Figure 3. Apparent resistivity for the different number of layer earth structure.

4. Simulation and Results

The deep learning model for the earth layer classifier has been coded by Keras as shown in Fig. 4. Keras is a user-friendly neural network library written in Python. 'Dense' is the layer type. Dense is a standard layer type that works for most cases. In a dense layer, all nodes in the previous layer connect to the nodes in the current layer [10].

As mentioned above, we have 20 nodes in the input layer, 128 nodes in the hidden layer, and 3 nodes in the last layer. The activation function of the hidden layer and last layer are ReLU and softmax respectively. The dropout is used with the 0.5 rate. Next, we need to compile our model. Compiling the model takes two parameters: optimizer and loss. The optimizer controls the learning rate. We use 'adam' as our optmizer. Adam is generally a good optimizer to use for many cases. The adam optimizer adjusts the learning rate throughout training. We use 'categorical_crossentropy' for our loss function. This is the most common choice for classification [10]. A lower score indicates that the model is performing better. The number of epochs is the number of times the model will cycle through the data. The more epochs we run, the more the model will improve, up to a certain point. After that point, the model will stop improving during each epoch. In addition, the more epochs, the longer the model will take to run. To monitor this, we use 'early stopping'. Early stopping will stop the model from training before the number of epochs is reached if the model stops improving. We will set our early stopping monitor to 10. This means that after 10 epochs in a row in which the model doesn't improve, training will stop

```
EPOCHS = 120

model = keras.Sequential([
    keras.layers.Dense(20),
    keras.layers.Dense(128, activation="relu"),
    keras.layers.Dropout(rate=0.5),
    keras.layers.Dense(3, activation='softmax')
])

model.compile(optimizer="adam", metrics=["accuracy"],
              loss="categorical_crossentropy")
early_stop = keras.callbacks.EarlyStopping(monitor='val_loss', patience=10)
model.fit(train_X, train_Y, epochs=EPOCHS, batch_size=5,
                    validation_data=(test_X, test_Y),
                    callbacks=[early_stop])

print("\n\n************ TRAINING START ************ ")
early_stop = keras.callbacks.EarlyStopping(monitor='val_loss', patience=10)
 model.fit(dr.train_X, dr.train_Y, epochs=EPOCHS, batch_size=5,
                    validation_data=(dr.test_X, dr.test_Y),
                    callbacks=[early_stop])
```

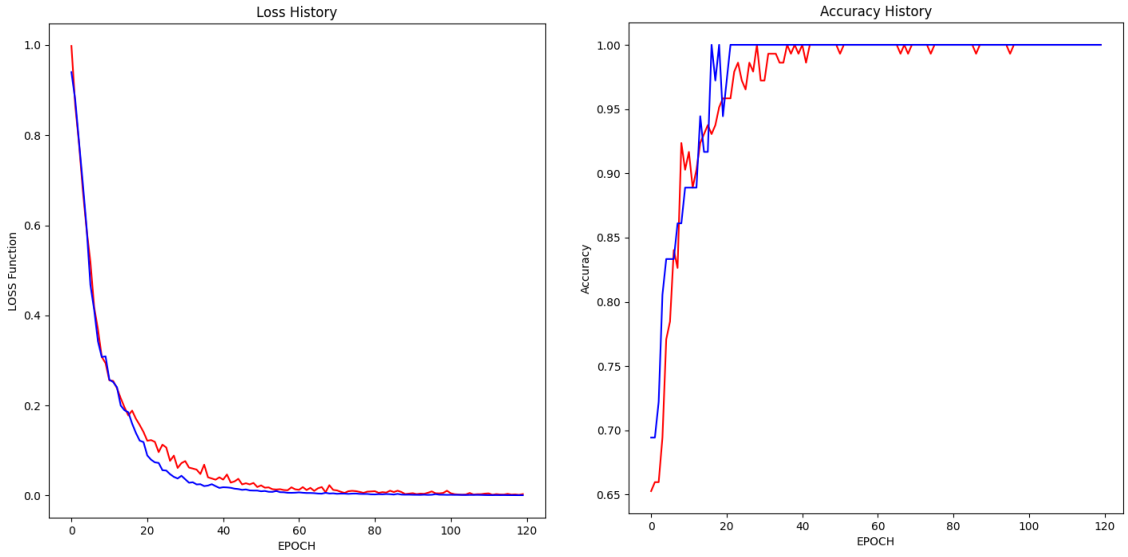
(a) Model configuration by Keras.

```
************ TRAINING START ************
2021-08-24 10:43:07.304584: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:176] None of the MLIR Optimization Passes a
Epoch 1/100
29/29 [==============================] - 0s 5ms/step - loss: 1.0446 - accuracy: 0.4722 - val_loss: 0.9006 - val_accuracy: 0.7778
Epoch 2/100
29/29 [==============================] - 0s 1ms/step - loss: 0.8255 - accuracy: 0.6806 - val_loss: 0.7188 - val_accuracy: 0.7778
Epoch 3/100
29/29 [==============================] - 0s 1ms/step - loss: 0.7261 - accuracy: 0.6875 - val_loss: 0.5959 - val_accuracy: 0.7778
Epoch 4/100
29/29 [==============================] - 0s 1ms/step - loss: 0.6142 - accuracy: 0.7569 - val_loss: 0.4931 - val_accuracy: 0.8889
Epoch 5/100
29/29 [==============================] - 0s 1ms/step - loss: 0.5270 - accuracy: 0.8056 - val_loss: 0.4152 - val_accuracy: 0.8611
Epoch 6/100
29/29 [==============================] - 0s 1ms/step - loss: 0.4425 - accuracy: 0.8472 - val_loss: 0.3488 - val_accuracy: 0.8889
```

(b) Iteration steps of the loss and accuracy function of traing and validation dataset.

Figure 4. The deep Learning Model for the earth layer classifier.

To make things even easier to interpret, we use the ‘accuracy’ metric to see the accuracy score on the validation set at the end of each epoch. As shown in Fig. 4b, the first two (loss and accuracy of each epoch) are for the training dataset and the next two (val_loss, val_accuracy) are for the validation dataset. The loss of both the training and validation sets decreases, and the accuracy of the training and validation sets increases as epochs progress. Fig. 5 shows the loss history and accuracy history for the training and validation sets. The blue line in Fig. 5 is the training set and the red line is the validation set. As shown in Fig. 5, the deep learning model of Earth Layer Classifier shows that it converges to 100% accuracy.



(a) Loss function. **(b) Accuracy function**

Figure 5. The loss and accuracy function of traing (blue line) and validation (red line) dataset.

5. Conclusion

We propose a deep learning classifier for estimating the number of layers in underground structures. The classifier consists of three layers of a feedforward neural network. Apparent resistance curves were used to train a deep learning classifier. Features are selected from 20 equally spaced log points on the apparent resistance curve. Apparent resistivity curve data sets are collected either by theoretical calculations or by Wenner's measurement method. Subterranean structures can be modeled using earth parameters such as number of layers, depth, and resistance of each layer. Knowing the exact number of layers can significantly reduce the amount of computation to estimate Earth parameters. The proposed method has been shown to converge with 100% accuracy in the layer count classification.

Acknowledgement

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF-2018R1D1A1B07045976) in (2018).

References

[1] A. I. Aderibigbe, Isaac Samuel, B. Adetokun and S. Tobi, "Monte Carlo Simulation Approach to Soil Layer Resistivity Modelling for Grounding System," International Journal of Applied Engineering Research, Volume 12, pp. 13759-13766, 2017.

[2] B. Zhang, X. Cui, L. Li, and J. L. He, "Parameter estimation of horizontal multilayer earth by complex image method," IEEE Trans. on Power Delivery, Vol. 20, pp. 1394—1401, 2005.

[3] H.C Kim, C.J Boo, and M.J Kang, "Estimation of kernel function using the measured apparent earth Resistivity," International Journal of Advanced Smart Convergence Vol.9 No.3 pp.97-104, 2020 http://dx.doi.org/10.7236/IJASC.2020.9.3.97

[4] H.C Kim and M.J Kang, "A Fast Calculation of Apparent Soil Resistivity Using Exponential Sampling Method," International Journal of Advanced Culture Technology, Vol.7 No.4 pp.268-273, 2019

https://doi.org/10.17703/IJACT.2019.7.4.268.

[5] T. Takahashi and T. Kawase, "Analysis of apparent resistivity in a multi-layer earth structure," IEEE Trans. on Power Delivery, Vol. 5, pp. 604—612, 1990.

[6] J. Alamo, "A comparison among eight different techniques to achieve an optimum estimation of electrical grounding parameters in two-layered earth," IEEE Trans. Power Del., vol. 8, pp. 1890–1899, Oct. 1993.

[7] H. Yang, J. Yuan, and W. Zong, "Determination of three-layer earth model from Wenner four-probe test data," IEEE Trans. Magn., vol. 37, pp. 3684–3687, Sep. 2001.

[8] Ammar Alali1, Frank Dale Morgan, and Darrell Coles, "Novel Approach for 1D Resistivity Inversion Using the Systematically Determined Optimum Number of Layers" Journal of Geology & Geophysics, Vol. 9, No: 481, 2020. http://hdl.handle.net/1721.1/117908

[9] U.K. Singh, R.K. Tiwari and S. B. Singh, "One-dimensional of geo-electrical resisitivity sounding data using artificial neural networks- a case study," Computer & Geoscience, Vol. 31, pp.99—108, 2005. https://doi.org/10.1016/j.cageo.2004.09.014

[10] Liheng Zhong, Lana Hu, and Hang Zhou, "Deep learning based multi-temporal crop classification," Remote Sensing Environment, vol. 221, pp430-443, Feb. 2019. DOI: 10.1016/j.rse.2018.11.032

[11] Jiacheng Sun, Xiangyong Cao, and Zhenguo LI, "New Interpretations of Normalization Methods in Deep Learning," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34 No. 04, 2020 DOI: https://doi.org/10.1609/aaai.v34i04.6046

[12] K. H. Jin, M. T. Mccann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," IEEE Trans. Image Process., vol. 26, no. 9, pp. 4509–4522, Sep. 2017.

[13] H.C Kim and M.J Kang, "A comparison of methods to reduce overfitting in neural networks," International Journal of Advanced Smart Convergence, Vol.9 No.2 pp.173-178, 2020. http://dx.doi.org/10.7236/IJASC.2020.9.2.173

[14] M.J Kang and H.C Kim, "Comparison of Weight Initialization Techniques for Deep Neural Networks," International Journal of Advanced Culture Technology, Vol.7 No.4 pp.283-284, 2019 DOI 10.17703/IJACT.2019.7.4.283