Author Classification on Bibliographic Data Using Capsule Networks Architecture

Firdaus, Wais Alqarni, Siti Nurmaini*, Annisa Darmawahyuni, Ade Iriani Sapitri, Muhammad Naufal Rachmatullah, Suci Dwi Lestari
Intelligent System Research Group, Universitas Sriwijaya, Palembang 30137, Indonesia
E-mail : firdaus@unsri.ac.id, wais.alq99@gmail.com, siti_nurmaini@unsri.ac.id, riset.annisadarmawahyuni@gmail.com, adeirianisapitri13@gmail.com, naufalrachmatullah@gmail.com, sucidl27@gmail.com

***Abstract*—The problem with Author Name Disambiguation is to determine whether the same name in the bibliographic archive refers to the same author or not. Currently, author identification on The Labeled Digital Bibliography and Library Project (DBLP) is triggered by a request for an author who finds his publication mixed with other people's writing. Name ambiguity leads to incorrect identification and attribution of credit to authors. Despite much research in the last decade, the issue of ambiguity of the author's name remains largely unsolved. In this paper, the Capsule Networks (CapsNets) method is proposed to resolve the ambiguity of the author's name. The proposed method obtains the best accuracy in four Name Disambiguation problems including homonyms, synonyms, and non-homonyms synonyms, which is an average of 99% on training and testing data. Likewise, the overall data tested has an accuracy of 99.83% with a low error value. In addition, CapsNets were tested with Performance Measurements including Sensitivity, Precision, and F1-Score. Capsnets can identify authors in DBLP bibliographic data by using a number of attributes such as author name, co-author, venue, title, and year.**

Keywords—Author Name Disambiguation; Capsule Networks; DBLP; Homonym; Authors Identification; Synonym

I. Introduction

Digital libraries such as DBLP, CitiSeer, Pubmed, and BDBComp provide features and services that facilitate research and discovery of scientific literature. According to Dongwon Lee et al, the challenge for Digital Libraries to have high-quality content is generally the issue of ambiguity in the author's name. The main challenge in this problem is determining whether two identical or identical names in the bibliographic archive record refer to the same author or not. This condition is complicated by two characteristics of the author's name, namely that different authors have and publish the same name on the author's note so that the author's name is exactly the same, but the author's entity is different. Likewise, an author sometimes uses a different name such as abbreviating the first or middle name so as to produce an authorship note with a different author name, but referring to the same author entity. The initial approach to solving this problem mostly involved manual disambiguation. However, the rapidly increasing number of researchers in digital libraries makes the manual disambiguation method impractical.

In general, the solutions that have been carried out to overcome these problems can be divided into two approaches, namely author grouping and author assignment. The author grouping method applies the similarity function to author reference attributes to decide whether the reference set refers to the same author entity or not [1][2]. Similarity functions can be grouped (clustering) using supervised machine learning techniques , or the correlation between authors and co-authors is represented in graphical form [3][4]. In general, not all functions used in this method are transitive and usually require a large number of samples and adequate features to function properly, which are usually very expensive to obtain. In addition to this technique, there is an approach that also gives the best results in identifying the author, namely the author assignment technique.

The author assignment method directly assigns each reference to a particular author by building a model that represents the author (for example, the possibility of an author publishing an article with another author (co-author), in a certain place and using a certain glossary in the title of the publication) using supervised machine learning techniques [5][6]. This method is most effective when found on a large number of sample citations for each author. Capsules with matrix transformation allow the network to study the relationship of each feature as a whole, increase the diversity of features, and capture pluralistic features of local sentences that can express text features more comprehensively [7]. This study specifically uses the DBLP database to identify authors based on a collection of bibliographic attribute data such as author id, author name, title of paper, venue, year, and author list. The results of this study were analyzed through Performance measurement with several indicators such as Accuracy, Error rate, Precision, F1-Score, and Recall.

II. Research Method

In this study, the approach used to solve the author disambiguation case is the author assignment method. This approach directly assigns each reference to a particular author by constructing a model that represents the author using a supervised classification technique. This study proposes an author identification process which consists of four stages; (i) data preparation, (ii) data pre-processing, (iii) classification, and (iv) model validation or evaluation. The labeled digital bibliography & library project (DBLP) dataset was used in this study.

A. *Data Preparation*

The GILES is one of the labeled data originating from DBLP that is used by a number of researchers to test various models of the Author Name Disambiguation (AND) algorithm. The dataset was created by Dr. Giles's and his team at Pennsylvania State University in 2004 [5]. The GILES data was generated by first collecting ambiguous author name publication records from DBLP and author web pages. Then, the researcher determined name identity by comparing the author's full name, co-author's name, affiliation, and email address. The research resulted in 8,453 data consisting of unique id, ambiguous author name, coauthor name(s), affiliation, title, venue, etc.

Several recent studies evaluated the The GILES dataset which noted that the data contained duplicate data and some incorrect records [8]–[10]. The dataset used in this study was taken from the research of Jinseok Kim et al [11]. The study improved the DBLP dataset from The GILES by removing duplicate records in the original The GILES data to correct errors in the data (eg, missing co-author names). The corrected dataset is matched with publication records in the DBLP library which is seen through a comparison of author name, year, title and venue. If a record in The GILES record does not match the DBLP record, the record is not used. This cleaning process resulted in a total of 5023 citation data with a number of ambiguous names and associated records (59% of the original The GILES data) then labeled data for 480 different authors [5]. Table 1 shows the information about The Giles dataset that has been improved in the study of Kim et al.

The GILES DBLP dataset which is the data in building the Author Name Disambiguation classification model will then be categorized into two main conditions that form the basis for the name disambiguation problem, namely synonymous cases and homonymous cases. Synonym is a condition when an author is identified with various name variations in his publications which often causes ambiguity by assuming the name variations are the names of different people. Meanwhile, homonym is a condition when the same name is used by different authors which causes ambiguity when the same name is assumed to refer to the same author. To group data based on these conditions, the attributes used are Author Name and Unique Author ID. Equation 1 shows the equation to identify synonyms and equation 2 shows the equation to identify homonyms [12].

$$synonym = X \rightarrow Y1 \mapsto m, m \geq 2 \qquad (1)$$

$$homonym = Y \rightarrow X1 \mapsto n, n \geq 2 \qquad (2)$$

In this equation, where X is the presented name, Y is the author. m is the sum of Y, and n is the sum of X. For the synonym condition, one Y has the number X more than or equal to two, whereas, for the homonym condition, one X has the number having Y more than or equal to two. The first step is to characterize the data that represents the synonym condition by separating the initialization of the data from the main data in the new label column by labeling the number 1 (one) if the synonym condition meets while if it does not meet it will be labeled 0 (zero). The same is done for the homonym condition.

In homonym conditions, the data is characterized if the data has the same author name but is different in the unique author id labeling. The data is initialized with a new column for homonym conditions by labeling the number 1 (one) when the condition is homonym and labeling the number 0 (zero) if it is non-homonym. After the two conditions have been completed (synonyms and homonyms), then these conditions are further developed with two further conditions, namely the synonym-homonym conditions in Equation 3 and the non-synonym-homonym conditions in Equation 4 [12].

$$synonymhomonym = synonym \cap homonym \qquad (3)$$

$$nonsynonymhomonym = (synonym \cup homonym)^{c} \quad (4)$$

After all data is categorized by case in author name disambiguation, The-GILES DBLP data has four conditions, namely synonyms, homonyms, synonyms-homonyms and non-synonyms-homonyms.

TABLE I. DATASET INFORMATION

No	*Author's Name*	*Number of Authors*	*Number of Citation Data*
1	A,Gupta	26	470
2	A.Kumar	14	187
3	C.Chen	61	475
4	D.Johnson	15	242
5	J.Lee	100	854
6	J.Martin	16	94
7	J.Robinson	12	142
8	J.Smith	31	479
9	K.Tanaka	10	173
10	M.Brown	13	109
11	M.Jones	13	166
12	M.Miller	12	125
13	S.Lee	86	960
14	Y.Chen	71	547
	Total	**480**	**5023**

B. *Data Preprocessing*

Data preprocessing, such as data normalization, feature extraction, and dimension reduction, are needed to make good classification data input. The purpose of preprocessing is to find the most informative set of features to improve classifier performance.

In the data pre-processing stage, each feature is processed with different features according to the characters in the data. The input feature consists of attributes and labels where there are five feature attributes, namely author name, author list, venue, title, and year and one label. The purpose of the normalization of the text is to produce data with the best format which is then entered into a data transformation that converts text data into numeric form.

Fig 1 shows the stage of pre-processing data. Each feature and label is processed. Feature processing is then transformed into a vector form which becomes the input data for the classifier. The label or unique id attribute is transformed using a label encoder and then entered into the encoding process using One Hot Encoder (OHE). The results of the label transformation produce a total of 266 features in the overall data category. Meanwhile, for feature attributes such as author name, author list, venue, and title, the data transformation process is carried out using word embeddings transformation. The process converts words in the form of alphanumeric characters into vector form. Each word is a vector that represents a point in space with a certain dimension. Each feature attribute is combined and then transformed with several word embedding transformation models such as Bag of Words, Tf-IDF, Word2Vec, and Glove.

C. *Classification*

Classification is the stage where the model is applied to identify the author of disambiguation in bibliographic data. The classifier used in this study is the Capsule Neural Network (CapsNets) which is a classification method in deep learning

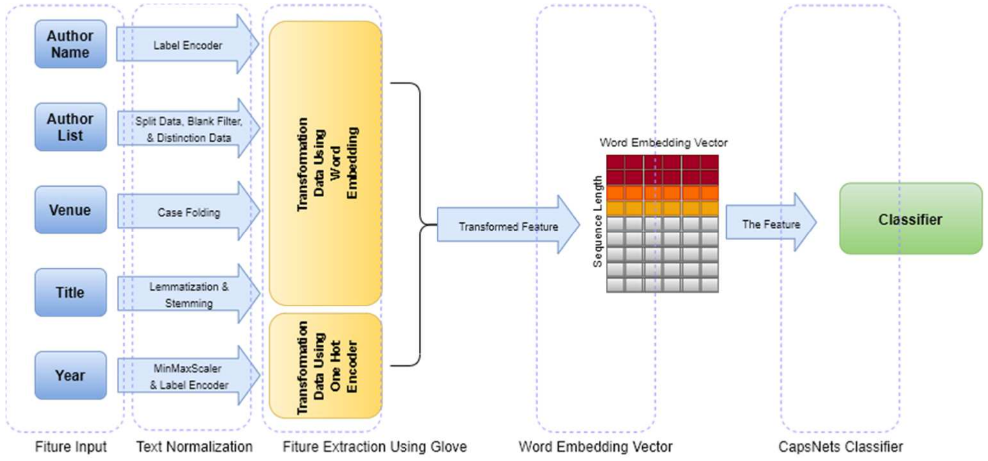

Fig 1. Data-Preprocessing Stage

consisting of a number of neurons whose activation vectors represent specific parameters for an entity such as objects or parts of objects. The experiment was conducted by testing a number of parameters such as number of capsules, optimization method, number of neurons, batch size and learning rate. The number of capsules tested is 5, 8 and 10. The optimization algorithm used during tuning is Adam Optimizer which calculates individual learning rates for different parameters. The learning rate tested is rated from the range of 0.1 to the lowest of 0.00001. The nodes used are rated at 32, 64, and 128. Batch size or the number of samples processed are given values of 32, 64, and 128. Furthermore, the epoch value is constant at a value of 50 epochs. Table II, shows CapsNets Classification Tuning. At the input layer the activation function applied is Rectified Linear Units (ReLu) while at the output layer the Softmax activation function is applied using Loss Categorical Cross Entropy.

TABLE II. CAPSNETS CLASSIFICATION TUNING

No	Test Size	Num Capsule	Learning Rate	Nodes	Batch Size	Epoch
1	0.2	5	0.1 ~ 0.00001	32	64	50
2	0.2	5	0.1 ~ 0.00001	64	64	50
3	0.2	5	0.1 ~ 0.00001	128	64	50
4	0.2	5	0.1 ~ 0.00001	32	128	50
5	0.2	5	0.1 ~ 0.00001	64	128	50
6	0.2	5	0.1 ~ 0.00001	128	128	50
7	0.2	8	0.1 ~ 0.00001	32	64	50
8	0.2	8	0.1 ~ 0.00001	64	64	50
9	0.2	8	0.1 ~ 0.00001	128	64	50
10	0.2	8	0.1 ~ 0.00001	32	128	50
11	0.2	8	0.1 ~ 0.00001	64	128	50
12	0.2	8	0.1 ~ 0.00001	128	128	50
13	0.2	10	0.1 ~ 0.00001	32	64	50
14	0.2	10	0.1 ~ 0.00001	64	64	50
15	0.2	10	0.1 ~ 0.00001	128	64	50
15	0.2	10	0.1 ~ 0.00001	32	128	50
17	0.2	10	0.1 ~ 0.00001	64	128	50
18	0.2	10	0.1 ~ 0.00001	128	128	50

III. RESULT AND DISCUSSION

The results of the transformation or feature extraction produce a number of vectors for each feature which are then combined to become input data for the classifier. Among a number of word embeddings feature extraction models used in this study, one of the results from the model used is Global Vectors for Word Representation (GloVe). The model is used because GloVe not only relies on local statistics (word local context information), but combines global statistics (co-occurring words) to obtain word vectors. Details of the feature transformation process with several word embeddings extraction models are described in Table III and Table IV which show the results of the transformation using One Hot Encoder.

TABLE III. FEATURE TRANSFORMATION RESULT

No	Fitur Extraction	*Bag of Words*	*Tf-IDF*	*Word2Vec*	*Glove*
1	Author Name	559	262	276	277
2	Author List	2341	3938	3733	6702
3	Venue	597	914	987	946
4	Title	1477	3220	3243	3243
	Total	**4974**	**8334**	**8239**	**11168**

TABLE IV. TRANSFORMATION WITH ONE HOT ENCODER

No	Feature	Number of Feature
1	Unique Author ID	266
2	Year	45

Table V shows the evaluation results of the Capsule Network model in several experimental scenarios or parameter tuning consisting of several performance measures such as training loss, training accuracy, testing loss, testing accuracy and model accuracy. Color gradation displays the level of value of each column.

From the results of the classification of the Capsule networks model on the ambiguity data of the DBLP author's name by applying a number of experimental parameters, it was found that the model showed the best performance in the 11th and 12th experiments with the same learning rate value of 0.01. the value in the 12th experiment showed better results than the 11th experiment by giving higher accuracy results and lower loss results so that the 12th experiment was better than the 11th experiment and, generally better compared to 17 other trials. The accuracy value in the 12th trial testing data is better than the accuracy value in the training data, which is 0.9980 : 0.9983 so it can be seen, if the model is able to give a good classification value to the new data or test data. Likewise with the error value generated in the 12th experiment, the error value in the testing data is lower than in the training data, which is 0.0071: 0.0078 so that the model shows good performance on the test data by minimizing misclassification of the training data. This is directly proportional to the higher accuracy value than other experiments, as well as the loss value shows lower than other experiments.

TABLE V. MODEL EVALUATION RESULTS

No.	Training Loss	Training Accuracy	Testing Loss	Testing Accuracy	Model Accuracy
1	0.0173	0.9963	0.0155	0.9964	0.9964
2	0.0239	0.9962	0.0234	0.9962	0.9962
3	0.0239	0.9962	0.0234	0.9962	0.9962
4	0.0240	0.9962	0.0233	0.9962	0.9962
5	0.0237	0.9962	0.0233	0.9962	0.9962
6	0.0145	0.9966	0.0129	0.9968	0.9968
7	0.0239	0.9962	0.0234	0.9962	0.9962
8	0.0110	0.9973	0.0093	0.9977	0.9977
9	0.0093	0.9977	0.0082	0.9980	0.9980
10	0.0239	0.9962	0.0234	0.9962	0.9962
11	0.0090	0.9977	0.0076	0.9981	0.9981
12	0.0078	0.9980	0.0071	0.9983	0.9983
13	0.0242	0.9962	0.0235	0.9962	0.9962
14	0.0185	0.9963	0.0158	0.9968	0.9968
15	0.0231	0.9962	0.0232	0.9962	0.9962
16	0.0240	0.9962	0.0233	0.9962	0.9962
17	0.0239	0.9962	0.0234	0.9962	0.9962
18	0.0238	0.9962	0.0233	0.9962	0.9962

The visualization or graph of the applied model can be seen in Fig 2 for the model performance graph on the train and testing accuracy categories. Fig 3 shows the loss model performance graph. The graph is a visualization of the results of training and testing data measured through changes in the loss and accuracy values. The graph shows the best experimental results are, in the 12th experiment with a number of parameter initializations. The graph shows good performance on the model marked with the best fit graph or the results between the testing and training values are quite good, and do not experience Overfitting or Underfitting on the graph.

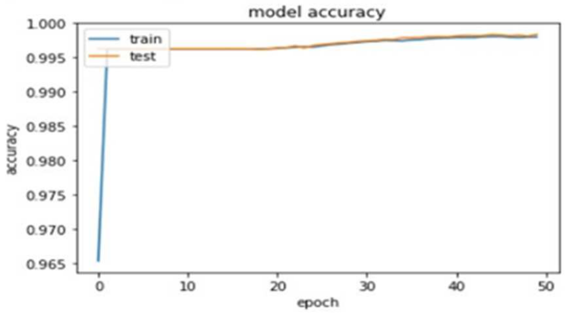

Fig 2. Accuracy Model Performance Graph

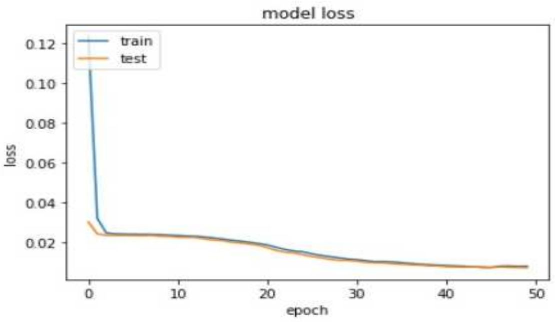

Fig 3. Loss Model Performance Graph

TABLE VI. MODEL EVALUATION RESULTS

Feature Extraction Model	Accuracy	Loss	Precision	Recall	F1 Score
Word2Vec	99.63	0.0167	42.00	93.00	58.00
TF-IDF	93.35	0.0020	90.00	92.00	90.00
Bag of Words	95.47	0.0019	94.00	94.00	93.00
Glove	99.83	0.0071	100.00	96.00	98.00

The comparison of the matrix performance values in the four feature extraction models can be seen in table VI. Four feature extraction models were also tested for performance such as accuracy, precision, recall, f1-score, and loss values using the CapsNets classification model to test the most appropriate feature extraction model used in the author's classification according to the classification model used.

Feature extraction models such as TF-IDF and Bag of Words have the same characteristics in the text feature extraction process. The matrix performance results generated in the Bag of Words model are slightly better than the TF-IDF on the overall score. The next feature extraction model used is the Word Embedding model including Word2Vec and Glove. The application of the Word Embedding model is based on the development of increasingly complex classification models such as the CapsNets model which tends to be more precise by using the word embedding model for the feature extraction process.

The matrix performance value generated by the word embedding model is also better than the other two feature extraction models, which is 99% accuracy. However, for the overall matrix performance results, the Glove model gives significantly better results than the Word2vec model and in general for the overall data.

Table VII shows the results of the model evaluation using Performance Measurement in each class from the author disambiguation classification which consists of homonym class, synonym class, synonym-homonym (SH) class, and non synonym-homonym (non-SH) class.

From the evaluation results of the entire existing class model, the model tends to show good performance on the sensitivity or recall value with the result that is 100% in each class, namely homonym, synonym, synonym-homonym and non-synonym-homonym class. Meanwhile, the scores that tend to be less good are the precision performance which scores quite low compared to other performances such as sensitivity, recall, and accuracy, in the synonym, homonym, and non-synonym-homonym classes. However, the model provides a good precision value for the synonym-homonym class. Meanwhile, the best average Performance Measurements value is obtained in the class of homonyms which shows that the model is very good at predicting the class of synonyms. But overall the model has given a good performance in classifying each author class in disambiguation through the evaluation of Performance Measurements.

TABLE VII. MODEL EVALUATION RESULT

Dataset	Sensitivity	Precision	F1-Score	Error	Accuracy
All	0.94	1.00	0.97	0.0078	0.9980
	0.96	1.00	0.98	0.0071	0.9983
Homonym	1.00	0.88	0.93	0.0157	0.9934
	1.00	0.83	0.91	0.0337	0.9913
Synonym	1.00	0.92	0.96	0.0156	0.9952
	1.00	0.91	0.95	0.0225	0.9943
SH	1.00	1.00	1.00	0.0121	0.9968
	1.00	1.00	1.00	0.0015	1.0000
Non-SH	1.00	0.90	0.95	0.0166	0.9961
	1.00	0.96	0.96	0.0156	0.9967

IV. CONCLUSION

Based on the results of research that has been carried out on the Author Name Disambiguation problem by classifying the author based on the author assignment approach using the Capsule Neural Network model, the conclusions are:

1) Author Name Disambiguation problem can be solved by applying the Capsule Neural Network (CapsNets) classification algorithm to identify author entities based on a number of bibliographic attributes with satisfactory classification results.

2) The results of the author's classification using the Capsule Neural Network model get very good results on the overall data test with the test results obtained which are 99.83% better than the results of training data which obtain an accuracy value of 99.80%, this shows that the model can adapt to the test data. Likewise, the results of training and testing loss obtained are quite low so it can be concluded that the model built is quite good in the author disambiguation classification process on the DBLP bibliographic dataset.

3) The results obtained based on Performance Measurements in a number of classes show that the model built tends to give very good results on sensitivity with a value of 100% in four data classes, namely synonyms, homonyms, synonyms, and non synonyms-homonyms for each. training and testing data.

4) Although overall the research produces data on accuracy and error rate that is quite good, the model built still does not show maximum performance on the precision value except for the synonyms-homonyms class.

5) The best classification results are shown in the synonym-homonym class with the overall Performance Measurements value reaching 100% on average. However, the built model still shows less than optimal results in the Homonym class compared to other class tests.

REFERENCES

[1] B.-W. On and D. Lee, “Scalable name disambiguation using multi-level graph partition,” in *Proceedings of the 2007 SIAM International Conference on Data Mining*, 2007, pp. 575–580.

[2] A. Culotta, P. Kanani, R. Hall, M. Wick, and A. McCallum, “Author disambiguation using error-driven machine learning with a ranking loss function,” in *Sixth International Workshop on Information Integration on the Web (IIWeb-07), Vancouver, Canada*, 2007.

[3] B.-W. On, E. Elmacioglu, D. Lee, J. Kang, and J. Pei, “Improving grouped-entity resolution using quasi-cliques,” in *Sixth International Conference on Data Mining (ICDM’06)*, 2006, pp. 1008–1015.

[4] X. Fan, J. Wang, X. Pu, L. Zhou, and B. Lv, “On graph-based name disambiguation,” *J. Data Inf. Qual.*, vol. 2, no. 2, pp. 1–23, 2011.

[5] H. Han, L. Giles, H. Zha, C. Li, and K. Tsioutsiouliklis, “Two supervised learning approaches for name disambiguation in author citations,” in *Proceedings of the 2004 Joint ACM/IEEE Conference on Digital Libraries, 2004.*, 2004, pp. 296–305.

[6] A. A. Ferreira, A. Veloso, M. A. Gonçalves, and A. H. F. Laender, “Effective self-training author name disambiguation in scholarly digital libraries,” in *Proceedings of the 10th annual joint conference on Digital libraries*, 2010, pp. 39–48.

[7] M. Chen, C. He, L. Lei, and D. Li, “Text Classification Based on A New Joint Network,” in *2020 5th International Conference on Control, Robotics and Cybernetics (CRC)*, 2020, pp. 13–18.

[8] M.-C. Müller, F. Reitz, and N. Roy, “Data sets for author name disambiguation: an empirical analysis and a new resource,” *Scientometrics*, vol. 111, no. 3, pp. 1467–1500, 2017.

[9] A. F. Santana, M. A. Gonçalves, A. H. F. Laender, and A. A. Ferreira, “On the combination of domain-specific heuristics for author name disambiguation: the nearest cluster method,” *Int. J. Digit. Libr.*, vol. 16, no. 3, pp. 229–246, 2015.

[10] D. Shin, T. Kim, J. Choi, and J. Kim, “Author name disambiguation using a graph model with node splitting and merging based on bibliographic information,” *Scientometrics*, vol. 100, no. 1, pp. 15–50, 2014.

[11] J. Kim and J. Kim, “The impact of imbalanced training data on machine learning for author name disambiguation,” *Scientometrics*, vol. 117, no. 1, pp. 511–526, 2018.

[12] S. Nurmaini *et al.*, “Author identification in bibliographic data using deep neural networks.,” *Telkomnika*, vol. 19, no. 3, 2021.