Author Matching Classification on a Highly Imbalanced Bibliographic Data using Cost-Sensitive Deep Neural Network

1st Firdaus
Intelligent System Research Group
Universitas Sriwijaya
Palembang, Indonesia
firdaus@unsri.ac.id

2nd Suci Dwi Lestari
Intelligent System Research Group
Universitas Sriwijaya
Palembang, Indonesia
sucidl27@gmail.com

3rd Siti Nurmaini*
Intelligent System Research Group
Universitas Sriwijaya
Palembang, Indonesia
siti_nurmaini@unsri.ac.id

4th Reza Firsandaya Malik
Communication Networks and Information Security Research Lab
Universitas Sriwijaya
Palembang, Indonesia
rezafm@unsri.ac.id

5th Muhammad Naufal Rachmatullah
Intelligent System Research Group
Universitas Sriwijaya
Palembang, Indonesia
naufalrachmatullah@gmail.com

6th Annisa Darmawahyuni
Intelligent System Research Group
Universitas Sriwijaya
Palembang, Indonesia
riset.annisadarmawahyuni@gmail.com

7th Ade Iriani Sapitri
Intelligent System Research Group
Universitas Sriwijaya
Palembang, Indonesia
adeirianisapitri13@gmail.com

8th Mohammad El Qiliqsandy
Intelligent System Research Group
Universitas Sriwijaya
Palembang, Indonesia
elqiliqsandy@gmail.com

2021 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS) | 978-1-6654-2733-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIMCIS53775.2021.9699331

***Abstract*—One of the stages before classifying the author matching is to combine the data, in this case the resulting data becomes highly imbalanced dataset, between the author who matches or the author who does not match. This paper presents a method to solve the highly imbalanced problem in author matching classification. The method used Cost-Sensitive Deep Neural Network (CSDNN). CSDNN will consider costs that vary from the type of data misclassification. As text feature similarity measures, we use cosine similarity. And we use Digital Bibliography & Library Project (DBLP) data as a dataset. The result is outstanding in terms of specificity 0.99, precision 0.95, recall 0.96, f1-score 0.96, and accuracy 0.99.**

Keywords—author name disambiguation, author matching, Cost-Sensitive Deep Neural Network, highly imbalanced data, bibliographic data

I. INTRODUCTION

Author name ambiguity occurs when a set of publication records contains ambiguous author names, such as the same author name may appear under distinct names, and the distinct author names may have similar names [1]. Author name ambiguity can be a great source of errors in a digital library nowadays. It may reduce the quality of information related to the author or organization. Still, the problem of author name ambiguity is closely related to authority control [2], name variant problem [3], record linkage [4], etc. Hence, this study may handle the problems of author name ambiguity with two approaches of author name disambiguation (AND); followed by author grouping, then author assignment method.

Author grouping method is finding some similarities between the author to author from publication data, and the author assignment method will directly be assigning each author. Both methods will try to create, select and combine features based on the similarity of attributes (co-authors, keywords, affiliations, publication years, etc.) by using several measures such as Jaccard, Jaro, and others, or several heuristics [5][6].

Several data pre-processing methods have been used for AND cases, such as pairwise. A pairwise method combines each attribute of the dataset, which it can be labeled by 0 as a distinct author, and 1 as a similar author. However, it can be the worst case if large imbalanced data affect its condition. If the number of label 0 larger than label 1, the performance of label 0 can obtain satisfying results. In previous work [7], Yamani et al. have proposed an isolation forest algorithm for anomaly detection. The result has obtained 99.5% accuracy. However, it does not present the performance of all labels (0 and 1), due to the total of label 0 achieved 98.9% of the total data if compared to label 1.

Large imbalanced data can lead to unexpected errors and even serious consequences in data analysis, specifically for classification task. Due to the class distribution tends to be more demanding for the classification algorithm to be biased towards the majority class. As a result, a standard classifier tends to misclassify a minority class and gets poor performance [8].

For handling the problem of imbalanced data, some methods have been proposed and grouped into three categories, i.e., data level, algorithm level, and hybrid approaches methods. Data level will reduce the level of returns through various data sampling methods. Algorithm level will handle imbalanced data that usually applied with a weight or cost scheme, including modifying the underlying learner or output to reduce bias towards the majority class. The last, hybrid approaches will strategically combine sampling methods and algorithmic methods [9][10].

This study concerns to solve the problem of imbalanced data with algorithm level method. The solution of algorithm level is attempting to adapt current classifier learning

algorithms to enhance minority class learning, such as cost-sensitive learning, ensemble learning, and hypernetwork [11]. Among the aforementioned algorithm level, this study uses cost-sensitive learning with deep neural network classifier. It can be one of the solutions to imbalanced data problems by considering the cost value associated with sample misclassification, specifically, assigning different cost values to misclassified samples [8].

II. METHODOLOGY

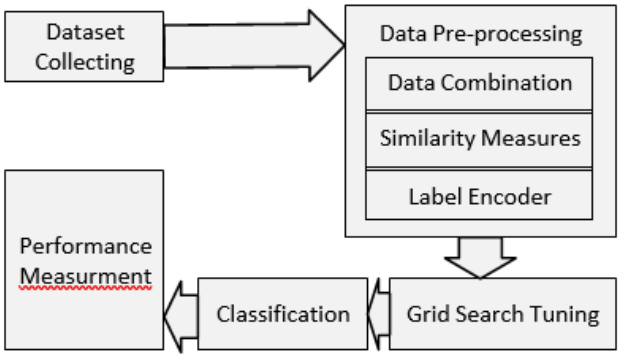

Fig. 1. Research Steps

A. Dataset

This study used DBLP labeled dataset obtained from previous work by Jinseok Kim et al. [12]. The number of datasets are 4419 data. The dataset provides seven attributes: author name, unique author ID, author list, title, year, venue, and paper ID. For this study, the paper ID attribute was not used due to the problem that the author matching or identification equation is not a document type. Only six attributes are sufficient to support and represent documentation from a publication.

B. Data Pre-processing

Pre-processing data on the dataset before it becomes the input to the classifier. There are several stages in data pre-processing until the data can be classified.

A combination process for all attributes is carried out. For the attributes of author name, author list, venue, and title, similarity measures are applied, and for year attribute, the difference is calculated (Fig. 2). Meanwhile, the author ID attribute is compared to produce author matching label (Fig. 3).

Equation 1 shows combination formula. Combinations are used to compare one row of data with all rows of data. From the total dataset of 4,419 lines, the combination process resulted in 9,761,571 data.

$$\frac{n!}{r!(n-r)!} = \binom{n}{r} \tag{1}$$

Cosine similarity is one of the most popular similarity measures applied to text documents, such as author name disambiguation [13][14]. Two documents initialized with X and Y, then their cosine similarity shows at equation (2).

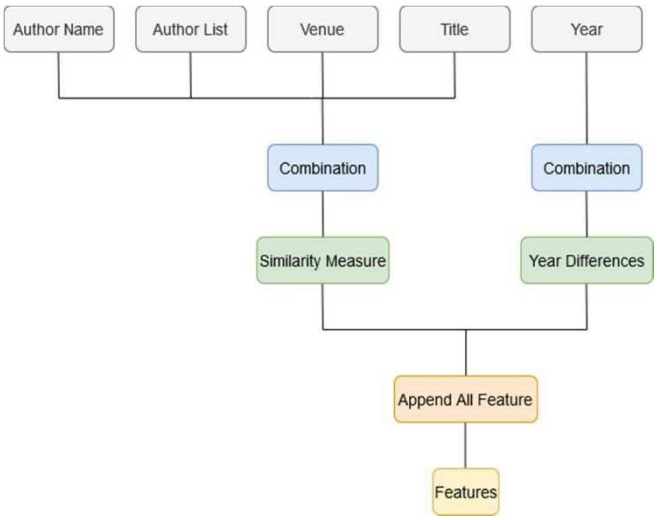

Fig. 2. Features Pre-Processing

Fig. 3. Label Pre-Processing

$$COS(x,y) = \frac{X.Y}{||X||\,||Y||} = \frac{\sum_{i=1}^{n} XiYi}{\sqrt{\sum_{i=1}^{n} X_i^2}\sqrt{\sum_{i=1}^{n} Y_i^2}} \tag{2}$$

X and Y are vectors of dimension m as long as the set of terms T = {t1, ..., tm}. Each dimension represents a weighted term in the document, which is not negative. As a result, the cosine similarity becomes non-negative and is limited in value between (0,1). An important tool of cosine similarity is the independence of the document length. For example, a document that has been copied identically from a document d to get a new pseudo d0, then the cosine similarity between d and d0 is 1. It means that both documents are considered identical documents [15].

For the year attribute, an absolute difference is carried out to produce a difference in years, and a minmax scaller process is carried out to obtain a smaller data range between the values 0-1.

After a combination of data is carried out, data comparison is carried out, namely comparing the Unique Author ID 1 data with Unique Author ID 2 data, whether or not they are the same data. If it is the same data, the value is True and if the data is different, the value is False. Because the data obtained is a boolean data type, then pre-processing is carried out using

the Label Encoder method. The Label Encoder's output is a value of 1 for True data and 0 for False data. The results from the Label Encoder will later become the label data feature.

C. Grid Search

Grid search is a hyperparameters model optimization technique [16]. Grid search will combine each parameter with several predetermined values, this combination is done to find one of the parameters that gets the best results [17]. The hyperparameters for which a combination of values will be sought are Batch Size, and Epoch. The Batch Size values used are 8, 16, 32, and 64. Finally, the Epoch values used are 100, 200, and 300. Meanwhile, other parameters that are not mentioned still use the same default settings as the classification process in general.

D. Cost-Sensitive Deep Neural Network

Most classifiers in general tend to pay less attention to rare cases in imbalanced datasets. Thus, resulting in minority data is often misclassified and tends to the majority class. Cost-sensitive classification will consider costs that vary from the type of data misclassification. The Bayesian optimal decision will play a role in obtaining cost-sensitive predictions. Equation (3) shows the label prediction class that achieves the lowest estimated cost.

$$ypred = \underset{1 \le k \le K}{argmin} \sum_{i=1}^{K} P(y = i|x, W, b)C(k, i) \quad (3)$$

Where C (k; i) denotes the cost of predicting a sample of class k as class i. K is the number of classes.

$$P(y = i\,|x, W, b) \quad (4)$$

Equation 4 is an estimate of the probability class i given by x. The probability estimator can be a classifier whose output is probability. In a neural network that considers the cost value by using a Deep Neural Network architecture, it is called a Cost Sensitive Deep Neural Network [18].

In a Cost Sensitive Deep Neural Network (CSDNN), it consists of an input layer, an output layer, and several hidden layers. There are m neurons in the input layer, where m is the dimension vector input feature. The hidden layer is completely connected to the previous layer. Then, the output layer is placed after the hidden layer. Thus, the Cost-Sensitive Deep Neural Network (CSDNN) is suitable for use in this study.

The proposed CSDNN structure consists of two Hidden Layers with 10 nodes. The activation function used in the Hidden Layer is Rectified Linear Units (ReLu). The Kernel Initializer used in the Hidden Layer is He uniform. We used Sigmoid as activation function in the Output Layer in order to ensure the prediction of probabilities is in the range of 0 and 1. The model will be optimized using Stochastic Gradient Descent (SGD) with a Learning Rate value of 0.001 and using the Binary Cross Entropy as a loss function. The fit function in the training data process uses the class weight argument, in this study, the class weight used is {0: 1, 1: 100}.

E. Performance Measurement

There are several things will be evaluated to determine the performance of our method [19]. We applied the value of accuracy equation (5), specificity equation (6), precision equation (7), recall equation (8), and F1-score equation (9).

$$Average\ Accuracy = \frac{\sum_{i=1}^{l} \frac{tp_i + tn_i}{tp_i + fn_i + tp_i + fp_i + tn_i}}{l} \quad (5)$$

$$Average\ Specificity = \frac{\sum_{i=1}^{l} \frac{tni}{tni + fpi}}{l} \quad (6)$$

$$Precision_M = \frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fp_i}}{l} \quad (7)$$

$$Recall_M = \frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fn_i}}{l} \quad (8)$$

$$F_1 - score = 2 \cdot \frac{Precision - Recall}{Precision + Recall} \quad (9)$$

Where tp is true positive, tn is true negative, fp is false positive, fn is false negative, and is the number of classes.

III. Result and Discussion

Before classification, the data is split into two parts, 80% for training and 20% for testing. After the grid search trial, the best parameters are obtained for the classification process using batch size 16 and epoch of 300 epoch. The grid search result is shown in table 1.

TABLE I. GRID SEARCH RESULT TABLE

Batch Size	Epoch	Accuracy	Loss
8	100	0.999463	0.000216
	200	0.999479	0.000229
	300	0.999469	0.000245
16	100	0.999480	0.000230
	200	0.999461	0.000229
	300	0.999480	0.000231
32	100	0.999472	0.000223
	200	0.999479	0.000231
	300	0.999480	0.000230
64	100	0.999283	0.000418
	200	0.999464	0.000223
	300	0.999480	0.000231

From the training and testing process, the performance of the model is presented in the training and testing confusion matrix as shown in table 2 and table 3. And the accuracy curve for training and testing can be shown in figure 4.

TABLE II. TRAINING CONFUSION MATRIX

	0	1
0	7,744,593	2,585
1	2,017	60,061

TABLE III. TESTING CONFUSION MATRIX

	0	1
0	1,936,151	629
1	503	15,032

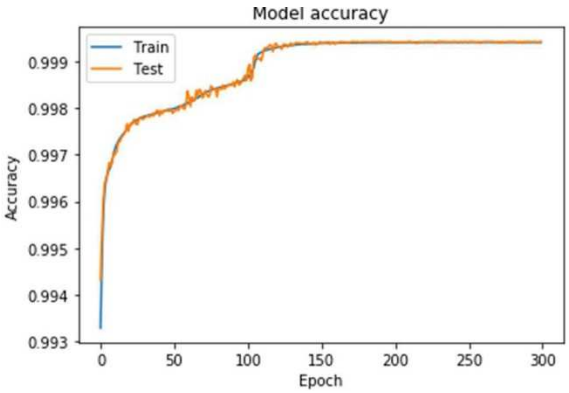

Fig. 4. Training and testing accuracy epoch result

From the confusion matrix, performance measurement is obtained from the proposed method. For the whole training data, the specificity and accuracy value is very good at 99%, for recall and F1-Score the value obtained is very good at 96%, also precision value is very good at 95%. Meanwhile, Performance Measurement for data testing, specificity and accuracy has a very good value of 99%, for recall and F1-Score the value is very good at 96%, and precision value is very good at 95%. The performance measurement of the proposed method can be shown in table 4.

TABLE IV. PERFORMANCE MEASUREMENT

Measurement	Training	Testing
Specificity	0.999666	0.999675
Precision	0.958736	0.959837
Recall	0.967509	0.967621
Error-Rate	0.000589	0.000579
F1-Score	0.963103	0.963713
Accuracy	0.99941	0.99942

IV. CONCLUSION

The challenge in the problem of author matching in the author name disambiguation is the highly imbalanced data. This paper proposes Cost-Sensitive Deep Neural Network (CSDNN) for author matching problems. For the classification process, Digital Bibliography & Library Project (DBLP) data with five attributes are used. The experimental results show very good results in terms of specificity, precision, recall, f1-score, and accuracy are 99%, 95%, 96%, 96%, and 99%, respectively.

ACKNOWLEDGMENT

We thank the Ministry of Research, Technology, and Higher Education, Republic of Indonesia (Kemenristekdikti RI), for funding the research on “Penelitian Disertasi Doktor” Research Grant with contract number 211/SP2H/LT/DRPM/IV/2019.

REFERENCES

[1] S. S. Khan and M. G. Madden, “A survey of author name disambiguation techniques: 2010–2016,” Knowl. Eng. Rev., vol. 00, no. January, pp. 1–24, 2017.

[2] R. C. Carrasco, A. Serrano, and R. Castillo-Buergo, “A parser for authority control of author names in bibliographic records,” Inf. Process. Manag., vol. 52, no. 5, pp. 753–764, 2016.

[3] G. Louppe, H. T. Al-Natsheh, M. Susik, and E. J. Maguire, “Ethnicity sensitive author disambiguation using semi-supervised learning,” in International Conference on Knowledge Engineeringand the Semantic Web, 2016, pp. 272–287.

[4] H.-C. Kum, A. Krishnamurthy, A. Machanavajjhala, M. K. Reiter,and S. Ahalt, “Privacy preserving interactive record linkage (PPIRL),” J. Am. Med. Informatics Assoc., vol. 21, no. 2, pp. 212–220, 2014.

[5] H. N. Tran, T. Huynh, and T. Do, “Author name disambiguation by using deep neural network,” in Asian Conference on Intelligent Information and Database Systems, 2014, pp. 123–132.

[6] A. A. Ferreira and M. A. Gonçalves, “A Brief Survey of Automatic Methods for Author Name Disambiguation,” vol. 41, no. 2, 2012.

[7] Z. Yamani, S. Nurmaini, and D. P. Rini, “Author Matching Classification with Anomaly Detection Approach for Bibliomethric Repository Data,” Comput. Eng. Appl. J., vol. 9, no.2, pp. 79–92, 2020.

[8] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, and P. J. Kennedy, “Training deep neural networks on imbalanced data sets,” in 2016 international joint conference on neural networks (IJCNN), 2016, pp. 4368–4374.

[9] J. M. Johnson and T. M. Khoshgoftaar, “Survey on deep learning with class imbalance,” J. Big Data, vol. 6, no. 1, p. 27, Dec. 2019.

[10] B. Krawczyk, “Learning from imbalanced data: open challenges and future directions,” Prog. Artif. Intell., vol. 5, no. 4, pp. 221– 232, 2016.

[11] F. Hu, X. Liu, J. Dai, and H. Yu, “A novel algorithm for imbalancedata classification based on neighborhood hypergraph,” Sci. World J., vol. 2014, 2014.

[12] J. Kim, “Evaluating author name disambiguation for digital libraries : a case of DBLP,” Scientometrics, vol. 116, no. 3, pp. 1867–1886, 2018.

[13] X. Lin, J. Zhu, Y. Tang, F. Yang, B. Peng, and W. Li, “A novel approach for author name disambiguation using ranking confidence,” in International Conference on Database Systems for Advanced Applications, 2017, pp. 169–182.

[14] K. Kim, S. Rohatgi, and C. Lee Giles, “Hybrid deep pairwise classification for author name disambiguation,” in International Conference on Information and Knowledge Management, Proceedings, 2019, pp. 2369–2372.

[15] R. Subhashini and V. J. S. Kumar, “Evaluating the performance of similarity measures used in document clustering and information retrieval,” in 2010 first international conference on integrated intelligent computing, 2010, pp. 27–31.

[16] T. Eitrich and B. Lang, “Efficient optimization of support vector machine learning parameters for unbalanced datasets,” J. Comput. Appl. Math., vol. 196, no. 2, pp. 425–436, 2006.

[17] W. Fu and T. Menzies, “Easy over hard: A case study on deep learning,” in Proceedings of the 2017 11th joint meeting on foundations of software engineering, 2017, pp. 49–60.

[18] H. Wang, Z. Cui, Y. Chen, M. Avidan, A. Ben Abdallah, and A. Kronzer, “Predicting hospital readmission via cost-sensitive deep learning,” *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 15, no. 6, pp. 1968–1978, 2018.

[19] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” Inf. Process. Manag., vol. 45, no. 4, pp. 427–437, 2009.