Author Matching Classification with Anomaly Detection Approach for Bibliomethric Repository Data

Zaqqi Yamani[1,2], Siti Nurmaini[2*], Dian Palupi R[3], Firdaus Firdaus[2], Annisa Darmawahyuni[2]

[1]*Master of Computer Science, Faculty of Computer Science, Universitas Sriwijaya*
[2]*Intelligent System Research Group, Faculty of Computer Science, Universitas Sriwijaya, Indonesia*
[3]*Faculty of Computer Science, Universitas Sriwijaya, Indonesia*
**siti_nurmaini@unsri.ac.id*

ABSTRACT

Authors name disambiguation (AND) is a complex problem in the process of identifying an author in a digital library (DL). The AND data classification process is very much determined by the grouping process and data processing techniques before entering the classifier algorithm. In general, the data pre-processing technique used is pairwise and similarity to do author matching. In a large enough data set scale, the pairwise technique used in this study is to do a combination of each attribute in the AND dataset and by defining a binary class for each author matching combination, where the unequal author is given a value of 0 and the same author is given a value of 1. The technique produces very high imbalance data where class 0 becomes 98.9% of the amount of data compared to 1.1% of class 1. The results bring up an analysis in which class 1 can be considered and processed as data anomaly of the whole data. Therefore, anomaly detection is the method chosen in this study using the Isolation Forest algorithm as its classifier. The results obtained are very satisfying in terms of accuracy which can reach 99.5%.

Keywords: Author Name Disambiguation, Author Matching, Anomaly Detection, IsolationForest.

1. INTRODUCTION

Ambiguity Author Name or better known as Author Name Disambiguation (AND) is one of the problems that reduce the quality and reliability of information obtained from the Digital Library (DL) [1][2]. DL content and service quality are strongly influenced by the ambiguity problem of the author's name in the citation and are considered as one of the most difficult problems faced by digital library researchers [3]. AND becomes a problem when a set of publication notes contains the name of the author which gives rise to more than one interpretation, ie the same author can appear with a different name [4][5]. This becomes a point that reduces the quality of information and also reduces the reliability of the information because it impacts the information on the author, organization and other things that are displayed as part of the publication's notes [6].

In its development, AND creates a daunting challenge in the technique of disambiguation because it often draws wrong conclusions on incomplete publication data [7] especially on the issue of author matching. There are several solutions implemented for AND especially at author matching points, including un-supervised, which is based on the similarity of bibliographic records or general writing patterns [8], author name disambiguation methods on multi-step clustering (NDMC) or which is done by generalizing the author's name or combining brief characteristics from publication data information [9]. Besides, there are other techniques, that have been used [10][11]. Graph Structural Clustering [10] uses community detection algorithms and graph operations, which at the end of this technical phase still use the similarity function of the published data [10]. Then, a visual analysis system technique called, NameClarifier which interactively groups authors' names in publications in certain circles, calculates and visualizes similarities between ambiguous and confirmed names in the Digital Library [11]. However, the three methods do not attach great importance to the accuracy of the publication data classification process. To address the problems, some supervised technique with machine learning has been explored for AND cases.

Machine Learning has been widely used to carry out the AND classification process and produce satisfactory performance [3]. Among them, supervised AND techniques using boosted tree classification that focuses on filtering and matching names and affiliations in a publication [12]. Besides, based on the Dempster Shafer Theory (DST) approach, namely calculating the similarity of high-level features such as affiliation, place, contentions, co-authors, quotes, the object of web correlation [13]. DST built a two-dimensional matrix for joint writing and topic relations and calculated the distance between two vertices with the help of Euclidean distances [14]. Also, Graph-Based AND techniques are using a multi-level Graph Partinging (MGP) algorithm, and a Multi-Level Graph Partinging and Merging (MGPM) algorithm [15] that use similarities between bibliographic records and groups of new quotation for authors with the same citation in DL, or new authors when the evidence for similarity is not accurate. Some special heuristics are used to check whether references to new citation records belong to the author that is already in the DL or belong to the new author (i.e. authors without citation records in the DL). The technique avoids the process running of disambiguation throughout the DL [16]. However, machine learning methods produce less satisfying accuracy performance.

Nowadays, there is one method to improve the superiority of the AND process by it called Deep Neural Networks (DNNs). This architecture has two main components; (i) the calculation of data taken as input and data representations [17], (ii) takes the basic set of features as its input and studies the features in it is a hidden layer to disguise the name of the maker [1]. Unfortunately, the DNNs process on large imbalanced class data makes the percentage of True Positive (TP) and True Negative (TN) are still has a gap. Hence, overall DNNs performance less than optimal even though it produces a very high level of accuracy in the range of 98%. The accuracy results are invalid due to data with the correct label and minority are not representative at all, or if it explained in the confusion matrix, then the True Positive data has 0 value.

From the aforementioned drawbacks, this paper conducted a study of AND by using one of the approachment based on unsupervised learning with anomaly detection. This approach commonly used to define threats, statistics, and machine learning algorithms [18][19]. In the process of threat detection, anomaly detection has the main ability to detect early and build information on data that represented in a minority [20]. This paper proposed the IsolationForest algorithm for anomaly detection. Based on our knowledge, the algorithm model has never been implemented in the AND classification process before.

2. MATERIAL AND METHOD

In this section, the author will explain the rare steps to prepare the AND data process before it is managed using a classifier. The steps of this study can be seen in Figure 1. It illustrates the whole process of this study. Starting from the (i) data preparation process consists of convert and deletion for data cleaning and structuring. Then (ii) data pre-processing starting from concatenation, combination, remove stopwords, similarity for the feature and comparison for the label. All the stages are carried out to get out the results in the form of vector values for features and binary class for the label. The next step (iii) classification process with the IsolationForest algorithm to define the anomaly detection method can be used or not. The last (iv) the evaluation stages to analyze the result of classification by using confusion matrix and performance report.

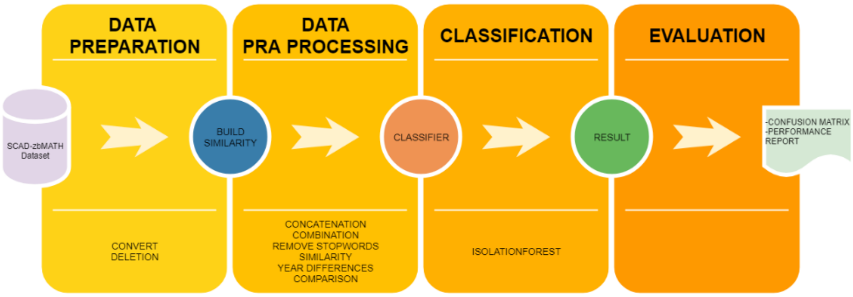


FIGURE 1. Author Matching on Anomaly Detection Process

2.1 MATERIAL PREPARATION

2.1.1 AUTHOR NAME DISAMBIGUATION RAW DATA

As for the important criteria in fulfilling the condition of a dataset into an AND dataset, there is a collection of publication data with several authors with various names of authors [6]. One of the datasets that match these criteria is the SCAD-zbMATH dataset. The dataset consists of several files with varying conditions, and one file that has the most complete structure is “featured-dataset-merged” in .xml format. The attributes contained in the file consist of publication id, name, title, venue, year, author in which there are author id, author names, and author short name. And all of these attributes are important resources in the AND data processing [4].

2.2 METHODOLOGY RESEARCH

The methodology becomes an important part of processing AND data before it is processed further by a classifier algorithm [3][21]. In this section, the author explains that each detailed step of the dataset processing is explained starting from the data after being converted to entering into the classification process.

2.2.1 DATA PREPROCESSING

Before entering the main process, the pre-processing is implemented to ensure the data that will produce will produce the correct prediction value in the classification process. This process called “cleaning and structuring data”, that carried out several stages as a cleaning process from the dataset (which has been converted). Each row of letters or text can be recognized properly, as well as structuring the data so that each feature can produce vector data that can be recognized properly by the classifier and each label can be defined correctly on existing features (Figure 2).

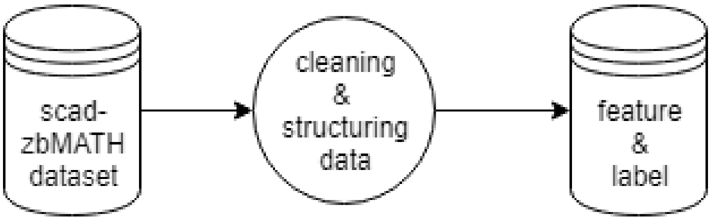
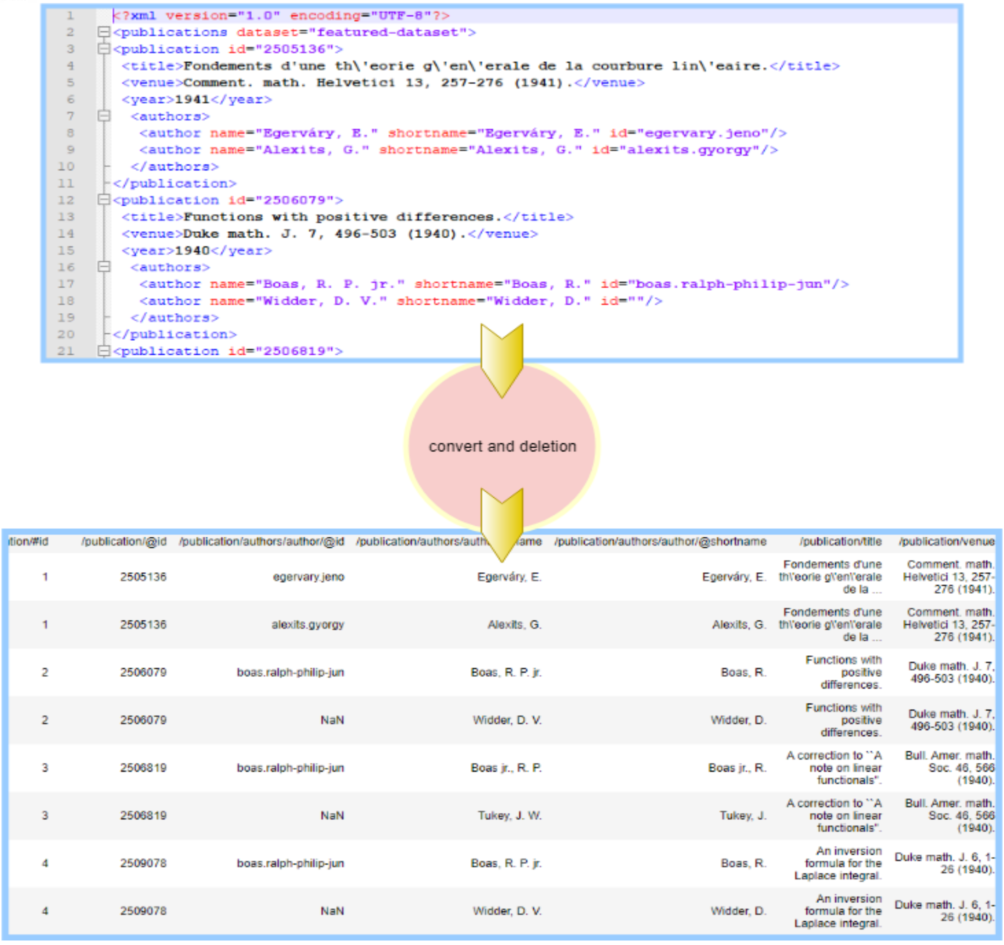


FIGURE 2. Preprocessing Scheme

2.2.1.1 CONVERT AND DELETION

Each attribute in the dataset contains the form of important information in the AND data processing. The “featured-dataset-merged” .xml file is converted to .csv, so that processing can be done more easily (Figure 3), by defining attributes to be id_authors, name, authors, venue, title, and year. The results of the definition will present 11.923 data rows with data that do not have an id_authors of 1.761 data. To avoid anomalies in the classification process of classifying data, data that does not have an id_authors is deleted and removed from the data to be processed. The process reduces the amount of data to 10.160 rows.



```xml
<?xml version="1.0" encoding="UTF-8"?>
<publications dataset="featured-dataset">
<publication id="2505136">
 <title>Fondements d'une th\'eorie g\'en\'erale de la courbure lin\'eaire.</title>
 <venue>Comment. math. Helvetici 13, 257-276 (1941).</venue>
 <year>1941</year>
  <authors>
   <author name="Egerváry, E." shortname="Egerváry, E." id="egervary.jeno"/>
   <author name="Alexits, G." shortname="Alexits, G." id="alexits.gyorgy"/>
  </authors>
</publication>
<publication id="2506079">
 <title>Functions with positive differences.</title>
 <venue>Duke math. J. 7, 496-503 (1940).</venue>
 <year>1940</year>
  <authors>
   <author name="Boas, R. P. jr." shortname="Boas, R." id="boas.ralph-philip-jun"/>
   <author name="Widder, D. V." shortname="Widder, D." id=""/>
  </authors>
</publication>
<publication id="2506819">
```

tion/#id	/publication/@id	/publication/authors/author/@id	/publication/authors/auth ame	/publication/authors/author/@shortname	/publication/title	/publication/venue
1	2505136	egervary.jeno	Egerváry, E.	Egerváry, E.	Fondements d'une th\'eorie g\'en\'erale de la ...	Comment. math. Helvetici 13, 257-276 (1941).
1	2505136	alexits.gyorgy	Alexits, G.	Alexits, G.	Fondements d'une th\'eorie g\'en\'erale de la ...	Comment. math. Helvetici 13, 257-276 (1941).
2	2506079	boas.ralph-philip-jun	Boas, R. P. jr.	Boas, R.	Functions with positive differences.	Duke math. J. 7, 496-503 (1940).
2	2506079	NaN	Widder, D. V.	Widder, D.	Functions with positive differences.	Duke math. J. 7, 496-503 (1940).
3	2506819	boas.ralph-philip-jun	Boas jr., R. P.	Boas jr., R.	A correction to ``A note on linear functionals".	Bull. Amer. math. Soc. 46, 566 (1940).
3	2506819	NaN	Tukey, J. W.	Tukey, J.	A correction to ``A note on linear functionals".	Bull. Amer. math. Soc. 46, 566 (1940).
4	2509078	boas.ralph-philip-jun	Boas, R. P. jr.	Boas, R.	An inversion formula for the Laplace integral.	Duke math. J. 6, 1-26 (1940).
4	2509078	NaN	Widder, D. V.	Widder, D.	An inversion formula for the Laplace integral.	Duke math. J. 6, 1-26 (1940).

FIGURE 3. The Result of Convert and Deletion Process

Attributes of data names, authors, titles, venues and years will be preprocessed and defined as features by finding similarity with the results in vector form (Figure 4). Then, the id_authors attribute will be defined as a label with results in the binary class at 0 and 1 (Figure 5).

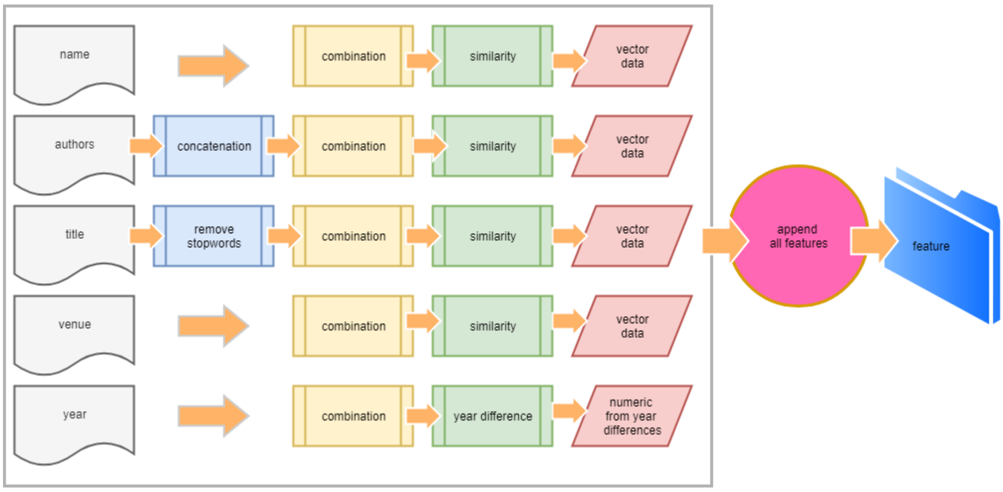


FIGURE 4. Data Cleaning & Structuring Scheme for feature

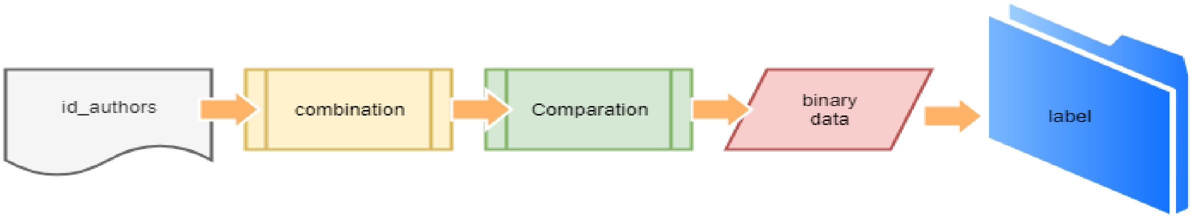


FIGURE 5. Data Cleaning & Structuring Scheme for Label

2.2.1.2 COMBINATION

The processing of the dataset is generated by doing a combination for the comparison process between the first line and the second line, the first line and the third line and so on. The results of the combination of the dataset used which has 10.160 data rows with 6 columns. It produces the number of new data rows of 51.607.720 rows. The number of the combination is express by,

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

where n is the number of objects that can be selected and r is the number that must be selected.

2.2.1.3 CONCATENATION

In the concatenation process, the authors' name concatenation process is carried out because the results of the authors' name provide a significant impact on precision results in the disambiguation method [22]. Then, the process of deleting data rows that do not have an id_authors is done do to the id_author will be an important point in the classification process and affect the results in the process [23]. The results of the two processes bring up a full dataset with complete attributes with 10,160 rows of data and 6 attributes.

2.2.1.4 REMOVE STOPWORD

The remove stop word process is an important process to add validity for the matching keyword section. This process eliminates un-useful words as word prepositions in the English language [2][12][23] [24][25]. In this study, the authors' did it in the title section of the dataset increase the value of keyword matching and produce good performance when processed in the classifier.

2.2.1.5 SIMILARITY AND YEARS DIFFERENCES

In the next step, the process finding the similarity of the combined data. The process of generating, selecting and combining features to produce similarities between attributes is done by giving a score or distance between attributes using the Jaccard coefficient [1]. The Jaccard coefficient was chosen because it has to produce a value of similarity after all the process is just which items are divided by the total items compared. Among all the attributes, the distance calculation is done by name, author, title, venue only, while year attribute is done by finding the difference between years based on the combination of results. Then, the id attribute is done by labeling. Simply put the formula used in the Jaccard coefficient is as follows:

$$jaccard\ (a,b) = \frac{|a\ \cap b|}{|a\ \cup b|} = \frac{|a\ \cap b|}{|a| + |b| - |a\ \cap b|} \tag{2}$$

TABLE 1.
The Result from Similarity and Year Difference (feature)

	name	authors	title	venue	year
0	0.33	1	1	1	0
1	0.21	0.14	0.35	0.5	1
2	0.28	0.26	0.24	0.56	1
3	0.21	0.14	0.25	0.54	1
4	0.28	0.25	0.26	0.47	2
5	0.21	0.15	0.38	0.53	3
6	0.21	0.15	0.23	0.53	3
7	0.21	0.15	0.23	0.48	3
8	0.21	0.15	0.38	0.5	3
9	0.41	0.4	0.33	0.5	3
.	.	.	.	.	.
.	.	.	.	.	.
51607720	0.27	0.35	0.26	0.44	7

The results of the combination and the distance calculation using the Jaccard coefficient can be seen in Table 1. The results only define the name, author, title, venue, year attributes that will be defined as a feature.

2.2.1.6 COMPARISON

The last, the combined id_authors attribute will be continued with the comparison process to determine the classification pattern of the processed dataset. The comparison process is done by comparing the 'id' column in each row of the resulting combination, where when the results are identical they will be labeled with the number 1 and the non-identical results will be labeled 0. The comparison function is express by,

$$comparison = \begin{cases} 0 = different\ author \\ 1 = same\ author \end{cases} \quad (3)$$

TABLE 2.
The Result from Comparison (Label)

	id_authors
0	0
1	0
2	0
3	0
4	0
5	0
6	0
.	.
.	.
51607720	0

2.3 CLASSIFICATION METHOD

2.3.1 ANOMALY DETECTION

Anomaly detection is used to detect threats, statistics and other machine learning algorithms [18][19][20]. Based on our knowledge, the anomaly detection method has never been used in the AND process for the Bibliomethric dataset. This approach is part of unsupervised learning by identifying unusual patterns that are not following the expected behavior, and this is often called an outlier. So it can be concluded that this anomaly detection is a process of identifying rare things from data and raises the suspicion that is significantly different from the majority of data. From the result comparisons and combinations for the label data, a significant imbalance data occurs in both class. It raised suspicions that minority data would not be detected in the study as true-value data. So, the anomaly detection approach becomes the best approach in producing accuracy values in the AND process.

2.3.2 ISOLATIONFOREST ALGORITHM

Isolation forest is one of the newest algorithm models for classifying anomaly detection. The algorithm is based on the fact that there still major and minority data ratio, it can be explained that anomalies are vulnerable to a mechanism called isolation. This method is fundamentally useful because it introduces the use of isolation as an effective and efficient way of detecting anomalies. Besides, this method is an algorithm with low linear time complexity and small memory requirements that can build a model that performs well by using a small subsample of fixed size, regardless of the size of the data set [26].

The main idea of the isolation forest algorithm is that the number of abnormal points is usually small, and there is a significant difference between normal points and attributes. This algorithm has a basic pattern on the decision tree model that can break down complex decision-making processes to be simple so that the decision-making process will better interpret the solution of the problem [26].

3. RESULTS & DISCUSSION

In this study, the classification process uses an anomaly detection algorithm, based on the experiment of isolation forest 2 times. The first experiment uses the overall data from the combined results to get the results of the classification. Then, in the second experiment, the data is splitting to 80% as the training set and the remaining for the testing set. The results of the comparison between Experiments 1 and 2 based on the performance matrix, i.e. accuracy, sensitivity, specificity, precision, and f1-score (Tables 3 and 4). From the comparison results, there are no significant changes in the performances after 2 times experiments. It can be concluded, the anomaly detection algorithm still robust for AND classification we can see that there were no significant changes in the performance side after being done with 2 times different experiments.

TABLE 3.
Ccomparison of The Results of Experiment 1 and 2 Performance (in Training Data)

Metrics Performance	Experiment 1	Experiment 2 Training Set
Accuracy	99.5%	99.5%
Sensitivity	78.5%	79.5%
Specificity	78.7%	78.5%
Precision	99.7%	99.7%
F1-Score	78.0%	78.0%

TABLE 4.
Comparison of the Results of Experiment 1 and 2 Performance (in Testing Data)

Metrics Performance	Experiment 1	Experiment 2 Testing Set
Accuracy	99.5%	99.5%
Sensitivity	78.5%	78.5%
Specificity	78.7%	78.5%
Precision	99.7%	99.6%
F1-Score	78.0%	78.0%

Furthermore, the performance report comparison for each class is presented in Table 5. It describes the performance report on each class to get a conclusion on which class the classification process is more dominant. In the precision, the level of accuracy between the information requested by the user and the answers given by the system can produce perfect scores in the majority class, and above 75% in the class in the minority class. Besides, in recall where the success rate of the system in finding back information is also able to produce perfect scores on the majority class, and above 75% for the minority class. Finally, the f1 score which is a comparison of the average precision and recall also produces performance values that are not much different. The experiment was conducted 2 times to get a different interpretation, but the performance produced in the 2 times the experiment did not show significant differences in results.

The results of the classification process performance can also be seen in a confusion matrix table that represents the amount of all available data to produce a predictive value. It represents the predictions and actual (actual) conditions of the data generated by the Isolation Forest algorithm, and to prove manually whether the performance report generated is true or not. It can also be seen the comparison between the results of the confusion matrix between the first and second experiments on the data taken as sample testing.

TABLE 5.
Performace Report Comparison in Each Class

	Experiment 1		Ex 2 (Training Set)		Ex 2 (Testing Set)	
	0	1	0	1	0	1
Precision	100%	78%	100%	78%	100%	78%
Recall	100%	79%	100%	79%	100%	79%
F1-Score	100%	78%	100%	78%	100%	78%

TABLE 6.
Confusion Matrix Result Comparison

	Experiment 1		Ex 2 (Testing Set)	
	0	1	0	1
1	458.92	125.022	93.177	23.611
0	131.704	50.892.074	25.033	10.179.723

The results of the author matching data classification using similarity techniques with the anomaly detection method can be seen visually by using the ROC curve. It was chosen because the visualization curve for the binary class is better represented by the ROC curve. Where the results of the curve describe the classification value close to number 1 which can be said to be the result of the process of this research very well.

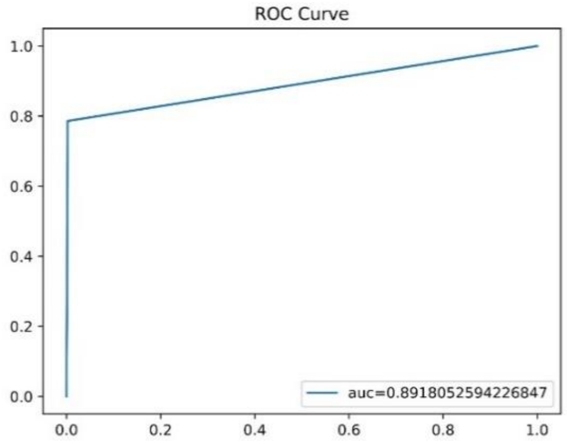


FIGURE 5. ROC Curve Anomaly Detection on IsolationForest Algorithm

In this study, the pre-processing is used to present data before entering the classifier are important points. The amount of data generated using these techniques is the basis for decisions that will use the classifier method and algorithm to be used. The number of data lines that reach above 50 million and the very imbalanced data from 2 classes that are used as a result of combination techniques to be followed by similarity techniques chooses methods and algorithms useless. The anomaly detection method with the Isolation Forest algorithm is able to produce high accuracy on a large amount of data and super highly imbalanced. From the result, 2 experiments is conducted in this study, the results showed a percentage is not significantly changed. It can be concluded that a classifier can be implemented for the large imbalance class in the AND process.

4. CONCLUSION

Author Name Disambiguation (AND) requires several steps in the preprocessing as an effort to establish a similarity of data that has string values. With the exact pre-processing results, the selection of classifier methods and algorithms can be very useful in presenting for very high predictive results. On the results of pre-processing data that produces a high level of returns, the anomaly detection method using the isolation forest algorithm is able to produce a very good level of accuracy compared to the previous literature using deep neural networks. from this study, the isolation forest algorithm achieves an accuracy of 95.5%. it concludes that anomaly detection and isolation forests can be used in the classification process of high-dimensional text data. However, the problem for the classification of text data that has a very large imbalanced data. some techniques and algorithms can be used in the anomaly detection method by increasing the assumption that much less data will be detected as an anomaly so that it can be declared in the classification process.

REFERENCES

[1] H. N. Tran, T. Huynh, and T. Do, "Author name disambiguation by using deep neural network," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8397 LNAI, no. PART 1, pp. 123–132, 2014.

[2] R. Hazra, A. Saha, S. B. Deb, and D. Mitra, "An efficient technique for author name disambiguation," *2016 IEEE Int. Conf. Curr. Trends Adv. Comput. ICCTAC 2016*, 2016.

[3] I. Hussain and S. Asghar, "A survey of author name disambiguation techniques: 2010–2016," *Knowl. Eng. Rev.*, vol. 32, p. e22, 2017.

[4] A. A. Ferreira, M. A. Gonçalves, and A. H. F. Laender, "A brief survey of automatic methods for author name disambiguation," *ACM SIGMOD Rec.*, vol. 41, no. 2, p. 15, 2012.

[5] F. Firdaus, "Improving Data Integrity of Individual-based Bibliographic Repository Using Clustering Techniques," *Comput. Eng. Appl. J.*, vol. 7, no. 1, pp. 49–56, 2018.

[6] 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.

[7] M. Song, E. H. J. Kim, and H. J. Kim, "Exploring author name disambiguation on PubMed-scale," *J. Informetr.*, vol. 9, no. 4, pp. 924–941, 2015.

[8] S. Milojević, "Accuracy of simple, initials-based methods for author name disambiguation," *J. Informetr.*, vol. 7, no. 4, pp. 767–773, 2013.

[9] S. Gu, X. Xu, J. Zhu, and L. Ji, "Name Disambiguation Method Based on Multi-step Clustering," *Procedia Comput. Sci.*, vol. 83, no. Ant, pp. 488–495, 2016.

[10] I. Hussain and S. Asghar, "LUCID: Author name disambiguation using graph Structural Clustering," *2017 Intell. Syst. Conf. IntelliSys 2017*, vol. 2018-Janua, no. September, pp. 406–413, 2018.

[11] Q. Shen, T. Wu, H. Yang, Y. Wu, H. Qu, and W. Cui, "NameClarifier: A Visual Analytics System for Author Name Disambiguation," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 1, pp. 141–150, 2017.

[12] J.-P. Wang *et al.*, "Effective string processing and matching for author disambiguation," vol. 15, pp. 1–9, 2013.

[13] H. Wu, B. Li, Y. Pei, and J. He, "Unsupervised author disambiguation using

Dempster–Shafer theory," *Scientometrics*, vol. 101, no. 3, pp. 1955–1972, 2014.

[14] Y. Zhu and Q. Li, "Enhancing object distinction utilizing probabilistic topic model," *Proc. - 2013 Int. Conf. Cloud Comput. Big Data, CLOUDCOM-ASIA 2013*, pp. 177–182, 2013.

[15] B. W. On, I. Lee, and D. Lee, "Scalable clustering methods for the name disambiguation problem," *Knowl. Inf. Syst.*, vol. 31, no. 1, pp. 129–151, 2012.

[16] A. P. de Carvalho, A. A. Ferreira, A. H. F. Laender, and M. A. Gonçalves, "Incremental Unsupervised Name Disambiguation in Cleaned Digital Libraries," *J. Inf. Data Manag.*, vol. 2, no. 573871, p. 289, 2011.

[17] F. Firdaus, M. Anshori, S. P. Raflesia, A. Zarkasi, M. Afrina, and S. Nurmaini, "Deep Neural Network Structure to Improve Individual Performance based Author Classification," *Comput. Eng. Appl. J.*, vol. 8, no. 1, pp. 77–83, 2019.

[18] A. Dawoud, S. Shahristani, and C. Raun, "Dimensionality Reduction for Network Anomalies Detection: A Deep Learning Approach," in *Advances in Intelligent Systems and Computing*, vol. 927, Springer Verlag, 2019, pp. 957–965.

[19] M. Fugate and J. R. Gattiker, "Anomaly detection enhanced classification in computer intrusion detection," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2002, vol. 2388, pp. 186–197.

[20] V. O. Ferreira, V. V. Galhardi, L. B. L. Gonçalves, R. C. Silva, and A. M. Cansian, "A model for anomaly classification in intrusion detection systems," in *Journal of Physics: Conference Series*, 2015, vol. 633, no. 1.

[21] M. J. Lerchenmueller and O. Sorenson, "Author disambiguation in PubMed: Evidence on the precision and recall of author-ity among NIH-funded scientists," *PLoS One*, vol. 11, no. 7, pp. 1–13, 2016.

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

[23] J. Zhao, P. Wang, and K. Huang, "A semi-supervised approach for author disambiguation in KDD CUP 2013," in *Proceedings of the 2013 KDD Cup 2013 Workshop on - KDD Cup '13*, 2013.

[24] K. Berzins, D. Pinheiro, D. Hicks, F. Xiao, J. Melkers, and J. Wang, "A boosted-trees method for name disambiguation," *Scientometrics*, vol. 93, no. 2, pp. 391–411, 2012.

[25] I. Hussain and S. Asghar, “Author Name Disambiguation by Exploiting Graph Structural Clustering and Hybrid Similarity,” *Arab. J. Sci. Eng.*, vol. 43, no. 12, pp. 7421–7437, 2018.

[26] C. Yao, X. Ma, B. Chen, X. Zhao, and G. Bai, “Distribution Forest: An Anomaly Detection Method Based on Isolation Forest,” 2019, pp. 135–147.