Ensemble Model Approach For Imbalanced Class Handling on Dataset

Ensemble Model Approach For Imbalanced Class Handling on Dataset

Yoga Pristyanto
Department of Information System
Universitas Amikom Yogyakarta
Yogyakarta, Indonesia
yoga.pristyanto@amikom.ac.id

Anggit Ferdita Nugraha
Department of Computer Engineering
Universitas Amikom Yogyakarta
Yogyakarta, Indonesia
anggitferdita@amikom.ac.id

Irfan Pratama
Department of Information System
Universitas Mercubuana Yogyakarta
Yogyakarta, Indonesia
irfanp@mercubuana-yogya.ac.id

Akhmad Dahlan
Department of Information Management
Universitas Amikom Yogyakarta
Yogyakarta, Indonesia
alland@amikom.ac.id

***Abstract*— In the field of machine learning, the distribution of classes in the dataset is important to produce a good model. The existence of class imbalances in the dataset is often ignored by researchers in the field of machine learning. This will certainly make the resulting model have less than maximum performance because theoretically, the single classifier has a weakness to the class imbalance conditions in datasets, this is because the majority of single classifier tends to work by recognizing patterns in the majority class in datasets that are not balanced so that the performance is not can be the maximum. Therefore, it is necessary to deal with these problems. In this study proposing an algorithmic level approach using Random Forest and Stacking to deal with these problems, the basic idea of using an algorithmic level approach is not to change the composition or pattern contained in the dataset itself. Based on tests using 5 datasets with different imbalance ratios, it shows that Random Forests or Bagging Tree and Stacking Naïve Bayes and Decision Tree C4.5 can produce better performance than single classifiers such as SVM, Naïve Bayes, and Decision Tree C4.5. So, the proposed method can be a solution in handling class imbalance in the dataset with different imbalance ratios.**

***Keywords*— Classification; Imbalanced Class; Ensemble; Stacking; Random Forest.**

I. INTRODUCTION

Imbalanced class is a condition that is often found in real-world problem datasets. It is a condition where there is a difference between the number of minority class instances and the majority of class instances [1]. The existence of an imbalanced class in the dataset can affect the performance of a single classifier because the majority of single classifiers tend to work by recognizing patterns in the majority class in datasets that are not balanced so that their performance cannot be maximized [2]. This certainly can lead to the risk of misclassification, so that it can cause the performance of a single classifier to be not optimal [3]. To overcome that problem, we need an approach or method that can be used to solve imbalanced class problems in the dataset. Besides, the condition of class imbalance in the dataset is also a challenge for researchers and practitioners in the field of data science.

There are two approaches or methods that can be applied to deal with imbalanced class conditions in the dataset, the approach at the data level and the approach at the algorithmic level. The method used in the data level approach is the resampling technique. Several studies have been carried out related to the approaches for imbalanced class handling in the dataset. Gongzhu Hu [4], and Jishan [5] conducted research related to the effect of imbalanced class handling on the performance of classification algorithms. Both studies indicate that the application of imbalanced class handling can improve the performance of classification algorithms. Those conducted research both using the SMOTE (Synthetic Minority Over-Sampling Technique) method. Also, Imran [6] made a comparison using two oversampling methods, namely SMOTE and Random Over Sampling. The results of the study showed that both of them were able to improve the performance of classification algorithms. While Rashu [7] and Thammasiri [8] used Random Under Sampling, the results show that the method method causes a decrease in the performance of classification algorithms. On the other hand, the research conducted by Kraat [9] uses one of the undersampling methods, namely OSS (One-Sided Selection). The results showed that the application of the OSS method could improve the performance of classification algorithms. Imbalanced class handling with a similar approach was also carried out by Noorhalim [10] and Zhihao [11] using the SMOTE method. The results of both studies indicate that the application of class imbalance handling in the dataset can improve the performance of several classification algorithms. Also, Sajid Ahmed [12] conducted a study related to handling class imbalances in the dataset. In this study, the technique used was ensemble resampling, while the methods tested included SMOTE-Bagging, RUS-Bagging, ADASYN-Bagging, and RYSIN-Bagging. The results of the study indicate that all four methods used have succeeded in improving the performance of the classification algorithm used. Yingze Yang [3] in his research used one of the ensemble resampling methods, namely SMOTE-Boosting, the results of the study showed that the proposed method was able to improve the performance of classification algorithms, but had not been tested on various datasets with different levels of imbalanced ratios.

As we know, the majority of these studies address class imbalances by using resampling techniques. On the other hand, the resampling technique has a drawback that is the risk of instance duplication and can cause of information and patterns loss that contained in the dataset, this certainly has an influence on the performance of a single classifier used, in addition to the data level approach also has the potential to change the composition contained in the dataset. Therefore this research will use an ensemble method or an algorithmic approach to minimize the risk of composition changes contained in the dataset. Two ensemble methods can be used, namely bagging (bootstrap aggregating) and stacking. There are two contributions to our research. First, the ensemble method that we propose can be one solution to deal with class imbalances in the dataset in the field of machine learning. Second, the ensemble method that we propose can be a reference for further research related to handling class imbalances in the dataset in the field of machine learning.

II. METHODS

Figure 1 shows the step of this research which consists of Data Acquisition, Preprocessing Data, Features Selection, Classification, and Evaluation.

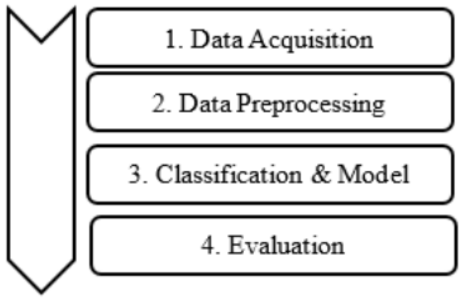

Figure 1. Research Step

A. Data Acquisition

Public datasets are used in this research. Datasets are obtained from UCI Machine Learning [13],[14] and Keel Datasets Repository [15]. This research uses 5 datasets with different imbalanced ratio. The following Table I shows the dataset information.

TABLE I. DATASETS INFORMATION

Datasets	Class	Instances	Majority Class	Minority Class	Imbalanced Ratio
Bank Direct Marketing [15]	2	4521	4000	521	7,68
Glass [15]	6	214	76	9	8,45
Web Phishing Binary Class [13]	2	2456	1362	1094	1,25
Web Phishing Multi Class [14]	3	1353	702	103	6,82
Credit Approval [15]	2	690	383	307	1,25

B. Data Preprocessing

This stage is the earliest in machine learning. The data preprocessing stage generally includes several things including filling in blank data, eliminating data duplication, and checking data inconsistencies. Usually, the blank data is caused by an error in the device when retrieving data and incomplete entry of new information. In the preprocessing stage, we fill in missing values in the dataset. The existence of missing values in the dataset will certainly also affect the results of the classification of the dataset itself. Therefore, at this stage, we fill in missing values using a constant value. In numerical data, missing values are replaced by average values, whereas for categorical data the missing values are replaced by mode values.

C. Classification and Model

• Decision Tree C4.5

Following steps are the steps of the C4.5 algorithm. First, choose a feature as root, then create a branch for each value of the root feature. The next step is to divide the cases into branches. Then the process is repeated for each branch until all records in the branch have the same class. The selection of root features is based on the strongest gain values of existing features [16]. Following equation (1) and equation (2) is to calculate gain value.

$$\text{Entropy}(S) = \sum_{i=1}^{n} - pi * \log_2 pi \quad (1)$$

where,

S : Record's amount

n : Partition amount of set S

pi : Si ratio toward S

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{i=1}^{n} \frac{|Si|}{|S|} * \text{Entropy}(Si) \quad (2)$$

where,

S : Record's amount

A : Feature

n : Partition amount of feature A

$|Si|$: record's amount on i-th partition

$|S|$: Record amount in S

• Naïve Bayes

Following are the stages of the process in the Naive Bayes algorithm in general [17]:

1) Calculate the whole class amount.
2) Calculate record's amount of each class.
3) Multiply all the class variable.
4) Compare values in each class to determine which class the record's belong to.

• Support Vector Machine

In this reseach, the SVM library to do the classification process. The following are the parameters in the SVM library.

1) The type of SVM used is C-SVC (Support Vector Classification). In the SVM library by default the value of the C parameter used is C = 1.0.
2) In this study used RBF (Radial Basis Function) Kernel. This kernel will map non-linearly separated data to a higher-dimensional space.
3) In this study, the gamma value used is ɣ = 0.

• Random Forest

Random Forest (RF) is a type of aggregating bootstrap method that has a way of working by generating several trees from the sample data where the making of one tree during the training process does not depend on the previous tree and then in making decisions based on the most votes [18].

The two concepts that form the basis of random forest are building an ensemble of trees via bagging with replacement and random selection of features for each tree that is built. Following is an explanation of the basic concepts in the Random Forest algorithm [19].

1) First, every sample taken from the dataset for the training process each tree can be used again for another training tree process.

2) Second, the features used during the training process in each tree are a subset of the features possessed by the dataset.

3) The random forest has two main parameters, namely parameter m which is a presentation of the number of trees to be used and parameter k which is a representation of the maximum number of features that are considered when branching in a tree. The more parameter values m, the better the results of the classification, while for the value k is recommended for square root or logarithm of the total number of features in the dataset.

4) The following Figure 2 is an illustration of the concept of the Random Forest algorithm [18], [19].

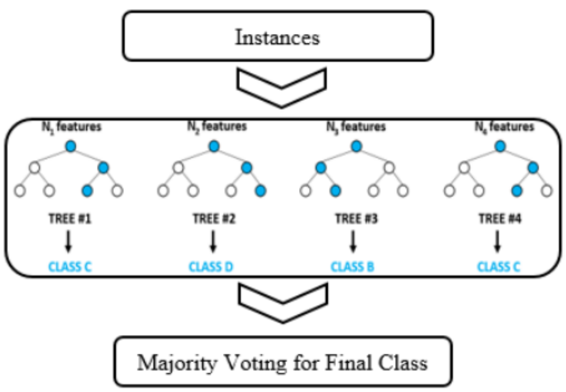

Figure 2. Random Forest Illustration [18], [19]

In this study, the number of trees used in the process of forming a Bagging Tree or Random Forest was 11 trees, we chose an odd number of trees so that during the voting process there was no redundant class.

• Stacking

Stacking is one of the ensemble algorithms, as shown in figure 3. N different subsets of the training dataset are created using stratified sampling with replacements where the relative proportions of different classes are maintained in all subsets. Each subset of the training set is used to determine the performance of the classifier on the training set. The meta-classifier in the form of relative weights for each classifier is created by assigning weights to classifiers that are proportional to their performance. Meta-classifiers can be described in various stages in a simplified meta-learning scenario. [20].

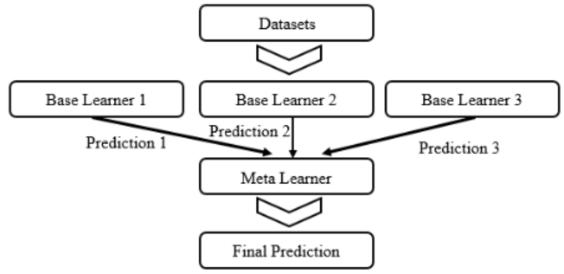

Figure 3. Stacking Algorithm Illustration

Following are the steps in the stacking algorithm:

- Basic classifiers are trained from the initial training set (basic level).
- Predictions are generated by classifiers who are trained in separate validation sets.
- The meta-level training set consists of the validation and prediction sets generated by the classifier in the validation set.
- The-meta classifier or final classifier is trained from the meta-level training set.

In this study, the stacking algorithm was used with C4.5 and Naïve Bayes-based learners, assuming that one of the single classifiers is not good enough in the case of an unbalanced class classification.

D. Evaluation

At this stage, a comparison between the performance of a single classifier algorithm, namely Decision Tree C4.5, SVM and Naive Bayes with the proposed method, is the ensemble classifier which includes Random Forest and Stacking. Evaluation indicators used in this study include accuracy, precision, recall, and AUC (Area Under Curve) [21]. The results of a data classification process can be categorized into four types, namely:

- TP True Positive is the number of positive records classified as positive.
- FP = False Positive is the number of negative records classified as positive.
- FN = False negatives are the number of positive records classified as negative.
- TN = True negative is the number of negative records classified as negative.

Accuracy in classification is a value or percentage of accuracy of data records that are classified correctly after testing the results of classification. Precision is the ratio of true positive predictions compared to the overall positive predicted results. While Recall is a true positive prediction ratio compared to overall true positive data. Here are the equations for calculating accuracy, precision and recall..

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

III. Results And Discussion

In this study, the datasets divided into two parts, namely as training data and testing data. We use data sharing scenarios as follows, 80% is used to train the model and 20% is used for model validation or model testing. The classification algorithm used is C4.5, SVM, NB Random Forest, and Stacking. In this study, testing was carried out using 5 datasets with different class imbalance ratios. The following Table 2 shows the average value of accuracy, precision, recall and AUC (Area Under Curve) of the classification algorithm used in each dataset.

TABLE II. Average Performance Comparison

Datasets	Methods	Accuracy	Precision	Recall	AUC
Bank Direction Marketing IR = 7,68	Decision Tree	0,881	0,865	0,881	0,576
	Naïve Bayes	0,87	0,863	0,87	0,836
	SVM	0,84	0,829	0,84	0,674
	Stacking Tree_NB	0,891	0,865	0,891	0,84
	Random Forest	**0,893**	**0,87**	**0,893**	**0,852**
Glass IR = 8,45	Decision Tree	0,717	0,665	0,717	0,931
	Naïve Bayes	0,639	0,512	0,639	0,818
	SVM	0,734	0,58	0,734	0,885
	Stacking Tree_NB	**0,752**	0,619	**0,753**	0,92
	Random Forest	0,722	**0,716**	0,722	**0,949**
Web Phishing Binary Class IR = 1,25	Decision Tree	0,943	0,943	0,943	0,964
	Naïve Bayes	0,942	0,942	0,942	0,987
	SVM	0,835	0,835	0,835	0,919
	Stacking Tree_NB	0,951	0,951	0,951	0,99
	Random Forest	**0,964**	**0,964**	**0,964**	**0,993**
Web Phishing Multi Class IR = 6,82	Decision Tree	0,877	0,877	0,877	0,935
	Naïve Bayes	0,836	0,816	0,836	0,943
	SVM	0,845	0,835	0,845	0,943
	Stacking Tree_NB	**0,888**	**0,879**	**0,888**	**0,961**
	Random Forest	0,88	0,879	0,88	0,957
Credit Approval IR = 1,25	Decision Tree	0,836	0,836	0,836	0,812
	Naïve Bayes	0,862	0,862	0,862	**0,92**
	SVM	0,854	0,856	0,854	0,915
	Stacking Tree_NB	**0,867**	**0,867**	**0,867**	**0,92**
	Random Forest	0,863	0,862	0,863	**0,92**

Table 2 shows the results of applying meta-learning or ensemble learning in terms of the average value of accuracy, precision, recall, and AUC. Based on tests using 5 datasets with different imbalance ratios, It shows that Random Forests or Bagging Tree and Naïve Bayes plus Decision Tree C4.5 Stacking can provide better performance than single classifiers such as SVM, Naïve Bayes, and Decision Tree C4.5. The Random Forest algorithm produces the best performance on the binary class phishing web dataset and bank direction marketing, while the Stacking algorithm between Decision Tree C4.5 and Naïve Bayes produces the best performance on a multi-class phishing web dataset and credit approval. In the glass dataset, both Random Forest and Stacking algorithms produced the best performance.

Theoretically, the single classifier has a weakness to the class imbalance conditions in datasets, this is because the majority of single classifier tends to work by recognizing patterns in the majority class in datasets that are not balanced so that the performance cannot be maximized. Therefore meta-learning algorithms such as Stacking and Random Forest or Bagging Tree can be a solution for the classification process in datasets with unbalanced classroom conditions. One of the advantages of using meta-learning in the classification process with cases of class imbalance in the dataset is that it does not change the composition or pattern contained in the dataset itself.

IV. Conclusion

Imbalanced class handling in the dataset is important, especially in the classification of data mining. Based on testing using 5 datasets with different imbalance ratios, it shows that Random Forests or Bagging Tree and Naïve Bayes plus Decision Tree C4.5 Stacking can produce better performance than single classifiers such as SVM, Naïve Bayes, and Decision Tree C4.5. So, the proposed method can be a solution in handling class imbalance in the dataset with different imbalance ratios. For further research, it is hoped that research can be carried out by testing using a method using a hybrid method between ensemble and resampling in the classification process with unbalanced class conditions.

Acknowledgment

Thanks to the Research Department, Universitas Amikom Yogyakarta who helped this research.

References

[1] K. Yang *et al.*, "Hybrid Classifier Ensemble for Imbalanced Data," *IEEE Trans. Neural Networks Learn. Syst.*, vol. PP, pp. 1–14, 2019.

[2] S. Barua, M. M. Islam, X. Yao, and K. Murase, "MWMOTE - Majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 2, pp. 405–425, 2014.

[3] Y. Yang, P. Xiao, Y. Cheng, W. Liu, and Z. Huang, "Ensemble Strategy for Hard Classifying Samples in Class-Imbalanced Data Set," *Proc. - 2018 IEEE Int. Conf. Big Data Smart Comput. BigComp 2018*, pp. 170–175, 2018.

[4] G. Hu, T. Xi, F. Mohammed, and H. Miao, "Classification of wine quality with imbalanced data," *Proc. IEEE Int. Conf. Ind. Technol.*, pp. 1712–1717, 2016.

[5] S. T. Jishan, R. I. Rashu, N. Haque, and R. M. Rahman, "Improving accuracy of students' final grade prediction model using optimal equal width binning and synthetic minority over-sampling technique," *Decis. Anal.*, vol. 2, no. 1, pp. 1–25, 2015.

[6] M. Imran, M. Afroze, S. K. Sanampudi, A. Abdul, and M. Qyser, "Data Mining of Imbalanced Dataset in Educational Data Using Weka Tool," *Int. J. Eng. Sci. Comput.*, vol. 6, no. 6, pp. 7666–7669, 2016.

[7] R. I. Rashu, N. Haq, and R. M. Rahman, "Data mining approaches to predict final grade by overcoming class imbalance problem," *2014 17th Int. Conf. Comput. Inf. Technol. ICCIT 2014*, pp. 14–19, 2014.

[8] D. Thammasiri, D. Delen, P. Meesad, and N. Kasap, "A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition," *Expert Syst. Appl.*, vol. 41, no. 2, pp. 321–330, 2014.

[9] M. Kubat and S. Matwin, "Addressing the Curse of Imbalanced Training Sets: One Sided Selection," in *International Conference on Machine Learning*, 1997, vol. 97, pp. 179–186.

[10] N. Noorhalim, A. Ali, and S. M. Shamsuddin, "Handling Imbalanced Ratio for Class Imbalance Problem Using SMOTE," in *Proceedings of the Third International Conference on Computing, Mathematics and Statistics (iCMS2017)*, 2017, pp. 19–29.

[11] Z. Peng, F. Yan, and X. Li, "Comparison of the different sampling techniques for imbalanced classification problems in machine learning," *Proc. - 2019 11th Int. Conf. Meas. Technol. Mechatronics Autom. ICMTMA 2019*, pp. 431–434, 2019.

[12] S. Ahmed, A. Mahbub, F. Rayhan, R. Jani, S. Shatabda, and D. M. Farid, "Hybrid Methods for Class Imbalance Learning Employing Bagging with Sampling Techniques," *2nd Int. Conf. Comput. Syst. Inf. Technol. Sustain. Solut. CSITSS 2017*, pp. 1–5, 2018.

[13] UCI ML, "UCI Machine Learning Repository : Website Phishing Data Set." [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Website+Phishing#.

[14] UCI ML, "UCI Machine Learning Repository : Phishing Websites Data Set." [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Phishing+Websites.

[15] J. Alcalá-Fdez *et al.*, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *J. Mult. Log. Soft Comput.*, vol. 17, no. 2–3, pp. 255–287, 2011.

[16] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

[17] M. Mayilvaganan and D. Kalpanadevi, "Comparison of classification techniques for predicting the performance of students academic environment," in *Communication and Network Technologies (ICCNT), 2014 International Conference on Computational Intelligence and Computer Research*, 2014, pp. 113–118.

[18] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[19] A. T. Wibowo, "Implementasi Algoritma Deteksi Spam Yang Tersisipi Informasi Citra Dengan Metode Svm Dan Random Forest," 2016.

[20] N. Chanamarn, K. Tamee, and P. Sittidech, "Stacking technique for academic achievement prediction," *Int. Work. Smart Info-Media Syst. Asia (SISA 2016)*, no. Sisa 2016, pp. 14–17, 2016.

[21] Q. Wang, "A hybrid sampling SVM approach to imbalanced data classification," *Abstr. Appl. Anal.*, vol. 2014, 2014.

Ensemble Model Approach For Imbalanced Class Handling on Dataset

ORIGINALITY REPORT

20%
SIMILARITY INDEX

PRIMARY SOURCES

1	Yoga Pristyanto, Sumarni Adi, Andi Sunyoto. "The Effect of Feature Selection on Classification Algorithms in Credit Approval", 2019 International Conference on Information and Communications Technology (ICOIACT), 2019 Crossref	149 words — 5%
2	Yoga Pristyanto, Noor Akhmad Setiawan, Igi Ardiyanto. "Hybrid resampling to handle imbalanced class on classification of student performance in classroom", 2017 1st International Conference on Informatics and Computational Sciences (ICICoS), 2017 Crossref	69 words — 2%
3	Yoga Pristyanto, Akhmad Dahlan. "Hybrid Resampling for Imbalanced Class Handling on Web Phishing Classification Dataset", 2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), 2019 Crossref	64 words — 2%
4	scholarworks.lib.csusb.edu Internet	60 words — 2%
5	Sumarni Adi, Yoga Pristyanto, Andi Sunyoto. "The Best Features Selection Method and Relevance Variable for Web Phishing Classification", 2019 International	56 words — 2%

Conference on Information and Communications Technology (ICOIACT), 2019
Crossref

6	Yoga Pristyanto, Irfan Pratama, Anggit Ferdita Nugraha. "Data level approach for imbalanced class handling on educational data mining multiclass classification", 2018 International Conference on Information and Communications Technology (ICOIACT), 2018 Crossref	34 words — 1%
7	journal.ugm.ac.id Internet	19 words — 1%
8	Arun Kulkarni, Leonard L.. "Phishing Websites Detection using Machine Learning", International Journal of Advanced Computer Science and Applications, 2019 Crossref	14 words — < 1%
9	Yoga Pristyanto, Irfan Pratama. "Missing Values Estimation on Multivariate Dataset : Comparison of Three Type Methods Approach", 2019 International Conference on Information and Communications Technology (ICOIACT), 2019 Crossref	14 words — < 1%
10	Hadj Ahmed Bouarara. "chapter 7 Multi-Agents Machine Learning (MML) System for Plagiarism Detection", IGI Global, 2018 Crossref	12 words — < 1%
11	Mohammad Suyanto, Ferry Wahyu Wibowo. "Animation opportunities of intelligent multimedia systems in developing a creative economy park", 2018 International Conference on Information and Communications Technology (ICOIACT), 2018 Crossref	11 words — < 1%

12 issuu.com
Internet
10 words — < 1%

13 Mohammed Al-Janabi, Ed de Quincey, Peter Andras. "Using supervised machine learning algorithms to detect suspicious URLs in online social networks", Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017, 2017
Crossref
9 words — < 1%

14 Waleed Ali. "Phishing Website Detection based on Supervised Machine Learning with Wrapper Features Selection", International Journal of Advanced Computer Science and Applications, 2017
Crossref
9 words — < 1%

15 Yan Wei, Ni Ni, Dayou Liu, Huiling Chen, Mingjing Wang, Qiang Li, Xiaojun Cui, Haipeng Ye. "An Improved Grey Wolf Optimization Strategy Enhanced SVM and Its Application in Predicting the Second Major", Mathematical Problems in Engineering, 2017
Crossref
9 words — < 1%

16 engj.org
Internet
9 words — < 1%

17 mdpi.com
Internet
9 words — < 1%

18 pt.scribd.com
Internet
8 words — < 1%

19 www.hindawi.com
Internet
8 words — < 1%

20 Dwi Sunaryono, Siti Rochimah, Riyanarto Sarno, Shoffi Izza Sabilla, Irzal Ahmad Sabilla, Dewi
7 words — < 1%

Sekarini. "Geotagging for Mapping Distribution of Meatballs Containing Borax based on Electronic Nose Detection", 2020 International Seminar on Application for Technology of Information and Communication (iSemantic), 2020
Crossref

21 Hui Han. "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning", Lecture Notes in Computer Science, 2005 — 7 words — < 1%
Crossref

22 Ilham A.E. Zaeni, Dessy Rif'a Anzani, Dwi Sudarno Putra, Mazarina Devi, Laili Hidayati, Imam Sudjono. "Classifying the Parental Involvement on School From Home during Covid-19 using C4.5 Algorithm", 2020 4th International Conference on Vocational Education and Training (ICOVET), 2020 — 7 words — < 1%
Crossref

23 Moloud Abdar, U. Rajendra Acharya, Nizal Sarrafzadegan, Vladimir Makarenkov. "NE-nu-SVC: A New Nested Ensemble Clinical Decision Support System for Effective Diagnosis of Coronary Artery Disease", IEEE Access, 2019 — 7 words — < 1%
Crossref

EXCLUDE QUOTES	ON	EXCLUDE SOURCES	OFF
EXCLUDE BIBLIOGRAPHY	ON	EXCLUDE MATCHES	OFF