Accredited Ranking SINTA 2
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 6 No. 3 (2022) 442 - 447 ISSN Media Electronic: 2580-0760

Prediction of Retweets Based on User, Content, and Time Features Using EUSBoost

Ghina Khoerunnisa[1], Jondri[2], Widi Astuti[3]
[1,2,3]School of Computing, Telkom University
[1]ghinak@student.telkomuniversity.ac.id, [2]jondri@telkomuniversity.ac.id, [3]widiwdu@telkomuniversity.ac.id

Abstract

Twitter is one of the popular microblogs that allow users to write posts. Retweeting is one of the mechanisms for the diffusion of information on Twitter. One way to understand the spread of information is to learn about retweet predictions. This study focuses on predicting retweets using Evolutionary Undersampling Boosting (EUSBoost) based on user, content, and time-based features. We also consider the vector of text as a predictive feature. Models with EUSBoost are able to outperform models using the AdaBoost method. The evaluation results show that the best model can achieve an AUC performance score of 77.21% and a GM score of 77.18%. While the Adaboost-based models achieved AUC scores ranging from 68% to 69% and GM scores ranging from 62% to 63%. In addition, we found that there was no significant difference between using numeric features only and combining numeric and text features.

Keywords: information diffusion, retweet prediction, EUSBoost

1. Introduction

The development of technology directs humans to a digital-based life. The presence of digital technology has a major impact on human life in various aspects. One of them is the aspect of giving and getting information. In this digital era, social networks are the main intermediary in the diffusion of information [1]. In social networks, humans or users are connected through social relationships, personal attachment, neighborhood, and other factors [2]. Many people spend hours online getting information about products [3], news, hobbies, and interests [4]. This has led to increased interactions on social networks. In addition, people on social media can also share information independently. Thus, the information can spread quickly over a wide area [5].

Microblogs such as Twitter are one of the media in the digital era that can disseminate information widely and quickly. On Twitter, the main mechanism for the diffusion of information is retweeting [6]. Retweeting is an activity to repost a tweet that looks like the original tweet. Not only limited to reposting but retweeting also includes quote retweets, which is retweeting with comments [7]. The more retweets of a tweet, the more widespread the information becomes. So, studying the spread of information through retweet prediction can assist in determining whether or not a tweet has the potential to go viral.

Our research on retweet prediction is based on several studies related as a reference. Research conducted by Bunyamin and Tunys by comparing various machine learning methods such as Passive Aggressive, Linear Support Vector Machine, Logistic Regression, Decision Tree, and Random Forest. The features used in this research are user and tweet-based. The results of the study indicate that the Random Forest model achieves the highest performance of the other learning methods considered. Then the use of user-based features and tweets outperforms user only and tweets only features, which means that these two features affect retweet predictions [8].

Other research on retweet prediction has also been carried out by Hoang and Mothe. This study uses a Random Forest algorithm and features based on user, content, and time. This study produces an evaluation score between 70% to 82% according to the data using the F-measure metric. This study also shows that some features have a greater influence on predicting retweets such as the number of followers, number of followees, and number of group users belong. In addition, time-based features are also highly correlated with retweet ability [9].

Accepted: 07-06-2022 | Received in revised: 24-06-2022 | Published: 30-06-2022

There are similar studies on retweet prediction that consider several implicit features, such as the research of Hoang and Mothe and the research of Daga, et al. Hoang and Mothe use the sentiment feature as an implicit feature, while Daga et al. implement information retrieval to predict retweets. Daga et al. compared two text vectorization methods, namely TF-IDF weighting and Doc2Vec with various learning algorithms such as Random Forest, Support Vector Machine, Neural Network, Logistic Regression, and Multinomial Naïve Bayes. The results show that all models provide 10% to 15% better accuracy using the TF-IDF method than Doc2Vec [4].

In this study, we aim to build a model that can predict which tweet will be retweeted and not retweeted. The features used are a combination of user-based, content-based, and time-based features from previous studies. In addition, we also consider the text as a feature by using TF-IDF weighting. The novelty of this study is we implementing a binary classification method, namely Evolutionary Undersampling Boosting. We consider our scenario to be a data imbalance problem because the number of instances in the class of not being retweeted greatly outnumbers the number of instances in the class of being retweeted.

2. Research Methods

2.1 Data Crawling

The dataset is obtained from crawling data on Twitter by utilizing the Twitter API. The crawled data is English tweet data from Twitter users. Akbar et al. said that the results of the analysis of social application trends in different communities resulted in Twitter, Youtube, and Facebook applications as the most popular applications from a business perspective [10]. Therefore, the keywords used are "e-commerce", "startup", and "marketing". The crawling process is carried out from January 31, 2022, to March 9, 2022. For data with tweets that are retweeted, it is labeled as 1 and those that are not retweeted are labeled as 0. The data obtained is then stored in a file with the format .csv.

The data crawling results consist of user objects, tweet objects, and entity objects, which will be used as predictive features. The feature is a necessary variable in the retweet prediction process. The features used in learning can affect the final prediction result. Some of the explicit and implicit features used in this study are user, content, and time-based features that have previously been used in Hoang and Mothe's research [9]. However, in this study, there are additional features such as verified account and also considering the text vectorization feature.

User-based features are features that are attached to the user. The time-based feature is a predictive feature related to the posting time of a tweet [9]. While content-based features are features that are attached to user tweets explicitly or implicitly. Implicit features are features on tweets that cannot be directly extracted into ready-to-use features. To obtain the implicit features an additional method is needed [7]. This feature shows that tweet content can help predict tweet popularity [11]. In this study, all of the features except text are numeric features. The list of features used to build the retweet prediction model is shown in Table 1.

Table 1. Features description and their origins

Feature.	Group	Description	Ref
Number of tweets	User	Number of tweets user posted	[8], [9], [12], [13]
Number of followers	User	Number of follower accounts	[8], [9], [12], [13]
Number of followees	User	Number of accounts followed	[8], [9], [12], [13]
Age of the account	User	Obtained from between the current year and the year the account was created	[8], [9], [13]
Number of tweets liked	User	Number of tweets a user liked	[9]
Average likes	User	Obtained by dividing the number of tweets liked by the age of the account	[9]
Average tweets	User	Obtained by dividing the number of tweets by the age of the account	[8], [9]
Username length	User	Length of username	[9]
Verified Account	User	Account verification status	[8], [12], [13]
Weekend	Time	Posting on the weekend	[9]
Morning	Time	Posting time between 04.00 AM to 09.59 AM	-
Afternoon	Time	Posting time between 10.00 AM to 02.59 PM	-
Evening	Time	Posting time between 03.00 PM to 06.59 AM	-
Night	Time	Posting time between 07.00 PM to 08.59 PM	-
Midnight	Time	Posting time between 09.00 PM to 11.59 PM	-
Dawn	Time	Posting time between 00.00 AM to 03.59 AM	-
Hashtag	Content	Tweet contains hashtag	[8], [9], [12]
URL	Content	Tweet contains URL	[8], [9], [12]
Tweet length	Content	Length of the tweet	[8], [9], [13]
Optimal length	Content	Number of characters in the tweet is between 70 to 100	[9]
Mention	Content	Tweet contains mention	[9]
Exclamation	Content	Tweet contains exclamation	[8], [9]
Picture	Content	Tweet contains picture	[9]
Video	Content	Tweet contains video	[9]
Sentiment	Content	Sentiment classes (positive, neutral, or negative)	[9]
Text	Content	Representation of tweet in vector shape	[4]

DOI: https://doi.org/10.29207/resti.v6i3.4125

2.2 Data Preprocessing

In the data preprocessing stage, the steps carried out consist of two stages, namely preprocessing data for numeric data and text data. In addition, there is also a labeling process for the sentiment feature. Data preprocessing for numeric data is cleaning the data from missing values and also duplicate data. We obtain the sentiment feature from the sentiment voting process of three people who have labeled text data with the sentiment. Meanwhile, for text data, data preprocessing carried out was to clean data from URLs, emojis, mentions, smileys, hashtags, and reserved words on Twitter such as "RT". Then cleaning the text data from punctuation marks, newlines, tabs, and numbers.

Users on Twitter usually make tweets with various abbreviations. So we also change the abbreviated words. Then the tweets are separated into words (tokens), remove stop words, and remove words that are not included in the English words dictionary. Then we do lemmatization which converts the token to its base word.

2.3 TF-IDF Weighting

As shown in Table 1, we consider the text to be a predictive feature. For data in the form of text, before entering into modeling, it is necessary to convert text data into numeric data. This is done because modeling in machine learning cannot process raw text [4]. One technique that is often used to convert text to numeric is to use TF-IDF weighting [14].

TF-IDF assigns a weight to each word with two criteria, namely TF (Term Frequency) and IDF (Inverse Document Frequency). TF shows the frequency of occurrence of a word in a sentence [14]. Then IDF counts the occurrence of a word in all documents or the logarithm of the division between the number of documents and the number of documents containing a word. IDF helps in eliminating the words that the majority appear because they will not have many contributions [4]. In the end, TF-IDF is the result of multiplication between TF and IDF scores.

2.5 Retweet Prediction Model

Evolutionary Undersampling Boosting (EUSBoost) is a boosting method inspired by Random Undersampling Boosting (RUSBoost) and similar boosting algorithms [15]. RUSBoost and similar boosting algorithms introduce an undersampling process in each iteration of the AdaBoost.M2 algorithm, except that EUSBoost uses Evolutionary Undersampling (EUS) instead of Random Undersampling (RUS) or SMOTE [16].

In EUS, each chromosome is a binary vector that represents the presence or absence of data. To reduce the search space, EUS only considers the majority class which means all data in the minority class is always introduced to a new data set [15]. The fitness function used to determine the ranking of the chromosomes can be seen in Equation 1. This function considers the balancing between minority and majority classes [16].

$$fitness_{eus} = \begin{cases} GM - \left|1 - \frac{n^+}{N^-} \times P\right|, if\ N^- > 0 \\ GM - P, if\ N^- = 0 \end{cases} \quad (1)$$

n^+ is the number of minority class data (retweeted), N^- indicates the number of majority class data (not retweeted). P is the factor used to balance the two classes, usually using a value of 0.2 and GM is the performance measure on EUS. EUS utilizes the CHC genetic algorithm with a HUX inclusion probability of 0.25 [16].

EUSBoost is an EUS embedded in the AdaBoost.M2 method. In EUSBoost, before conducting the training process, EUS will return a new dataset consisting of all minority class data and selected data from the majority class. Then calculate the weights for the new dataset and perform training. Data that is not included in the new dataset is retained but has no weight, so it is ignored [17].

One of the problems with EUSBoost is that finding an accurate basic classification can lead to a loss of diversity in the resulting model [16]. So there is a modification to the fitness function that considers diversity. The modified fitness function equation is shown in Equations (2)–(3).

$$\beta = \frac{N-t-1}{N} \quad (2)$$

$$fitness_{eus_H} = fitness_{eus} \times \frac{1.0}{\beta} \times \frac{10.0}{IR} + H \times \beta \quad (3)$$

Where $fitness_{eus}$ is the fitness function in Equation 1. β is the weight change factor in each iteration. N is the number of data and t is a running iteration. IR is the imbalance ratio obtained from the division between the number of minority class data and the number of majority class data. H is the minimum Hamming distance between candidate chromosomes and all previously generated chromosomes. In the first iteration ($t = 0$), EUS uses the fitness function in Equation 1 because there is no vector to compare with the candidate chromosome solution.

There are two scenarios of data features that are carried out, namely numeric features only and combined features between numeric features and text features. We developed two models by implementing EUSBoost in each scenario. To compare the performance of EUSBoost, we also developed two models with the same two scenarios by utilizing the AdaBoost algorithm. The list of learning algorithms used can be seen in Tabel 2, while the parameters used can be seen in Tabel 3.

DOI: https://doi.org/10.29207/resti.v6i3.4125

Table 2. The resume of the models used in experiments

Abbr.	Algorithm	Feature Scenario
EUS1	EUSBoost	Numeric + Text
ADB1	AdaBoost	Numeric + Text
EUS2	EUSBoost	Numeric only
ADB2	AdaBoost	Numeric only

Table 3. Parameters used in each algorithm

Algorithm	Parameters
EUSBoost	Population size = 30, Generation = 50, No of estimators = 200, Evaluation measure = GM, Distance function = Euclidean, P = 0.2
AdaBoost	No of estimators = 200

2.6 Evaluation Metrics

To evaluate the performance of the EUSBoost model using only numeric features and a combination of numeric and text features two evaluation metrics were used. The metrics are Geometric Mean (GM) and Area Under the ROC Curve (AUC). GM is the result of calculating the geometric mean of sensitivity and specificity [18]. The plot that visualizes the balance achieved between the True Positive rate and the False Positive rate can be seen from the ROC curve [19]. AUC is a scalar measure that is widely used as an evaluation metric because it is independent of imbalanced data. The formula of GM can be seen in Equations (4)–(6) and AUC can be seen on Equations (4)–(8).

$$sensitivity = \frac{TP}{TP+FN} \quad (4)$$

$$specificity = \frac{TN}{FP+TN} \quad (5)$$

$$GM = \sqrt{sensitivity\ x\ specificity} \quad (6)$$

$$fall\ out = \frac{FP}{FP+TN} \quad (7)$$

$$AUC = \frac{1 + sensitivity - fall\ out}{2} \quad (8)$$

3. Results and Discussions

3.1 Dataset

In this study, the crawled dataset consisted of 162340 rows. From these datasets, we found a lot of missing values and duplicate data. The amount of data after the data cleaning process from missing values and duplicates is 59802 rows. To reduce sentiment labeling time, we use 15% data sampling from the total number of clean data. So, the amount of data used for sentiment labeling and text data preprocessing is 8969 lines. After that, the three students carried out sentiment labeling and then took a vote on the three sentiment results as the final result of the sentiment label. Then we do data preprocessing for text data from cleaning to lemmatization.

The dataset is separated into train data and test data with a ratio of 70:30. The comparison of the number of retweeted and not retweeted data in the data train and data test is shown in Figure 1 and Figure 2. In the data train, there are 5321 rows of data that are not retweeted and 957 rows of data that are retweeted. Then in the test data, there are 2288 rows of data that are not retweeted and 403 rows of data that are retweeted. In the train and test data, the difference in the number of retweeted and not retweeted data is both quite large. Therefore, this research is included in the case of imbalance. For imbalanced data, undersampling or oversampling is necessary. However, we do not undersample or oversample the data because EUSBoost has already implemented undersampling in their algorithm. The last data preprocessing is converting text data into numeric vectors using TF-IDF weighting. To shorten the modeling time, we use PCA by reducing the dimensions of the feature text.

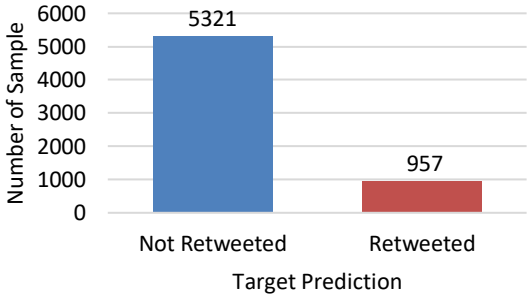

Figure 1. The number of samples in the data train

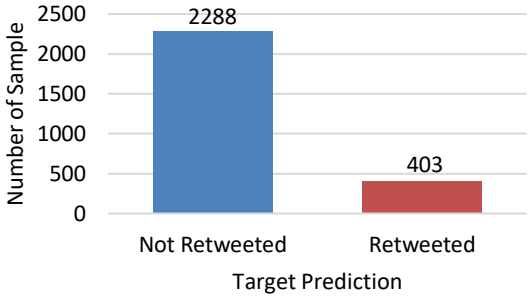

Figure 2. The number of samples in the data test

3.2 Experimental Result

The EUSBoost algorithm was built by deriving the AdaBoost classifier from the sklearn library using the python programming language. In this study, four experiments were conducted, namely, EUSBoost using only numeric features; EUSBoost with a combination of numeric and text; AdaBoost with only numeric data; and AdaBoost with a combination of numeric and text. When building a model with the EUSBoost algorithm it takes about 19 to 20 hours of runtime, while AdaBoost only takes seconds. This happens because in EUSBoost there is an undersampling process using a genetic algorithm that calculates fitness for all individuals.

DOI: https://doi.org/10.29207/resti.v6i3.4125

The results of the evaluation of the experiments carried out on the AUC and GM values are presented in Table 4. We found that the AUC and GM scores of EUS1 and EUS2 were not significantly different while the ADB1 and ADB2 models showed a difference in scores. In this study, we consider the AUC score as a measure to determine the best model, so EUS1 is the best model with a score of 77.21%. Meanwhile, the model with the lowest AUC score is ADB1, with a score of 68.43%.

Table 4. The resume of experimental results

Metric	EUS1	EUS2	ADB1	ADB2
AUC	**77.21**	75.93	68.43	69.01
GM	**77.18**	75.91	62.11	63.04
TN	**1722**	1703	2223	2221
FN	**84**	91	243	238
TP	**319**	312	160	165
FP	**566**	585	65	67

The comparison of AUC scores for each experiment is shown in Figure 3. The comparison results show that the model with EUSBoost is superior to AdaBoost for each feature scenario, namely numeric features only and combined numeric and text features. From the experiment, it is also known that the EUSBoost model with numeric and text features has a slightly higher score than with numeric features only. Whereas the AdaBoost model is the opposite, the model with combined numeric and text features is slightly smaller than the model with only numeric features.

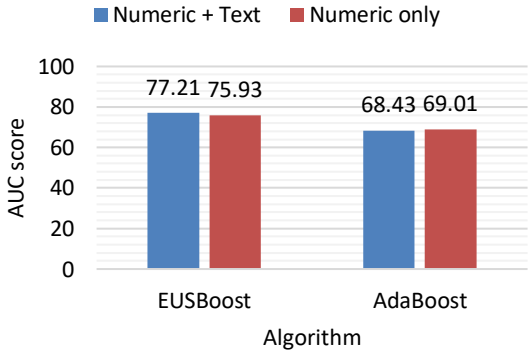

Figure 3. Comparison of the AUC score for each experiment

The experimental results scores for each model appear to be different. To see if the differences between each model are significant or not, we make a comparison by calculating the statistical significance of the p-value. Assuming the value of α is 0.05. The comparison of the four models with their p-values is shown in Table 5.

Based on the p-value of EUS1 and EUS2, we cannot reject the null hypothesis so that EUS1 and EUS2 are not significantly different. Likewise, with the ADB1 and ADB2 models. This shows that there is no difference between the model with numeric features and the model with combined numeric and text features. Meanwhile, the comparison of each EUSBoost model with the AdaBoost model resulted in a p-value less than 0.05. This means we can reject the null hypothesis hence EUSBoost models are significantly different from AdaBoost models.

Table 5. The comparison between experiments with statistical significance *p*-value ('==' means is not significantly different)

Hypothesis	*p*-value(AUC)	Inference
EUS1 == EUS2	0.289687	Accept
EUS1 == ADB1	0.001164	Reject
EUS1 == ADB2	0.001999	Reject
EUS2 == ADB1	0.004957	Reject
EUS2 == ADB2	0.008024	Reject
ADB1 == ADB2	0.431326	Accept

4. Conclusion

In this study, we aim to develop a predictive model of whether a tweet will be retweeted or not. We reused some user-based, content-based, and time-based features from the previous study. Besides that, we also consider the text as a predictive feature. We developed a binary classification machine learning model using the EUSBoost classifier. We found that our EUSBoost model performs better than the AdaBoost model. According to the experimental result, our best model is model EUS1 which is the EUSBoost using combined numeric and text features with an AUC score of 77.21%. In addition, our experiments show that there is no significant difference between the models with numerical features only and the models with the combination of numeric and text features so the difference does not have a major effect on the models.

For future work, we would like to develop and evaluate EUSBoost with a bigger dataset and a bigger number of estimators. We also would like to apply another method for converting text to numerical vectors such as Word2Vec or Doc2Vec.

Reference

[1] B. Arafah and M. Hasyim, "Social Media as a Gateway to Information: Digital Literacy on Current Issues in Social Media," *Webology*, vol. 19, no. 1, pp. 2491–2503, Jan. 2022, doi: 10.14704/web/v19i1/web19167.

[2] P. Kumar and A. Sinha, "Information diffusion modeling and analysis for socially interacting networks," *Social Network Analysis and Mining*, vol. 11, no. 1, Dec. 2021, doi: 10.1007/s13278-020-00719-7.

[3] Y. K. Dwivedi *et al.*, "Setting the future of digital and social media marketing research: Perspectives and research propositions," *International Journal of Information Management*, vol. 59, Aug. 2021, doi: 10.1016/j.ijinfomgt.2020.102168.

[4] I. Daga, A. Gupta, R. Vardhan, and P. Mukherjee, "Prediction of likes and retweets using text information retrieval," in *Procedia Computer Science*, 2020, vol. 168, pp. 123–128. doi: 10.1016/j.procs.2020.02.273.

[5] Y. Özkent, "Social media usage to share information in communication journals: An analysis of social media activity and article citations," *PLoS ONE*, vol. 17, no. 2 February, Feb. 2022, doi: 10.1371/journal.pone.0263725.

[6] S. N. Firdaus, C. Ding, and A. Sadeghian, "Retweet: A popular information diffusion mechanism – A survey paper," *Online Social Networks and Media*, vol. 6, pp. 26–40, Jun. 2018, doi: 10.1016/j.osnem.2018.04.001.

DOI: https://doi.org/10.29207/resti.v6i3.4125

[7] S. N. Firdaus, C. Ding, and A. Sadeghian, "Retweet Prediction based on Topic, Emotion and Personality," *Online Social Networks and Media*, vol. 25, Sep. 2021, doi: 10.1016/j.osnem.2021.100165.

[8] H. Bunyamin and T. Tunys, "A Comparison of Retweet Prediction Approaches: The Superiority of Random Forest Learning Method," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 14, no. 3, p. 1052, Sep. 2016, doi: 10.12928/telkomnika.v14i3.3150.

[9] T. B. N. Hoang and J. Mothe, "Predicting information diffusion on Twitter – Analysis of predictive features," *Journal of Computational Science*, vol. 28, pp. 257–264, Sep. 2018, doi: 10.1016/j.jocs.2017.10.010.

[10] Z. Akbar, J. Liu, and Z. Latif, "Mining social applications network from business perspective using modularity maximization for community detection," *Social Network Analysis and Mining*, vol. 11, no. 1, p. 115, 2021, doi: 10.1007/s13278-021-00798-0.

[11] M. G. Silva, M. A. Domínguez, and P. G. Celayes, "Analyzing the retweeting behavior of influencers to predict popular tweets, with and without considering their content," in *Communications in Computer and Information Science*, 2019, vol. 898, pp. 75–90. doi: 10.1007/978-3-030-11680-4_9.

[12] M. Wang, W. Zuo, and Y. Wang, "A multidimensional nonnegative matrix factorization model for retweeting behavior prediction," *Mathematical Problems in Engineering*, vol. 2015, 2015, doi: 10.1155/2015/936397.

[13] K. Lytvyniuk, R. Sharma, and A. Jurek-Loughrey, "Predicting Information Diffusion in Online Social Platforms: A Twitter Case Study," in *Studies in Computational Intelligence*, 2019, vol. 812, pp. 405–417. doi: 10.1007/978-3-030-05411-3_33.

[14] D. Kim, D. Seo, S. Cho, and P. Kang, "Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec," *Information Sciences*, vol. 477, pp. 15–29, Mar. 2019, doi: 10.1016/j.ins.2018.10.006.

[15] M. Galar, A. Fernández, E. Barrenechea, and F. Herrera, "EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling," *Pattern Recognition*, vol. 46, no. 12, pp. 3460–3471, Dec. 2013, doi: 10.1016/j.patcog.2013.05.006.

[16] B. Krawczyk, M. Galar, Ł. Jeleń, and F. Herrera, "Evolutionary undersampling boosting for imbalanced classification of breast cancer malignancy," *Applied Soft Computing Journal*, vol. 38, pp. 714–726, Jan. 2016, doi: 10.1016/j.asoc.2015.08.060.

[17] D. Ameta, "Ensemble Classifier Approach in Breast Cancer Detection and Malignancy Grading - A Review," *International Journal of Managing Public Sector Information and Communication Technologies*, vol. 8, no. 1, pp. 17–26, Mar. 2017, doi: 10.5121/ijmpict.2017.8102.

[18] W. Lee and K. Seo, "Downsampling for Binary Classification with a Highly Imbalanced Dataset Using Active Learning," *Big Data Research*, vol. 28, p. 100314, May 2022, doi: 10.1016/j.bdr.2022.100314.

[19] J. Zhao, J. Jin, S. Chen, R. Zhang, B. Yu, and Q. Liu, "A weighted hybrid ensemble method for classifying imbalanced data," *Knowledge-Based Systems*, vol. 203, Sep. 2020, doi: 10.1016/j.knosys.2020.106087.

DOI: https://doi.org/10.29207/resti.v6i3.4125