Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12708

e-ISSN : 2541-2019
p-ISSN : 2541-044X

Retweet Prediction Based on User-Based, Content-Based, Time-Based Features Using ANN Classification Optimized with the Bat Algorithm

Muhammad Rafi Rahadian[1)*], Jondri[2)], Kemas Muslim L[3)]
[1)2)3)]School of Computing, Telkom University Bandung, Indonesia
[1)] rafirahadian.student@telkomuniversity.ac.id, [2)]jondri@telkomuniversity.ac.id,
[3)]kemasmuslim@telkomuniversity.ac.id

Submitted : July 17, 2023 | **Accepted** : July 19, 2023 | **Published** : July 20, 2023

Abstract: Twitter is one of the most popular social media platforms today for information dissemination. It is favored by the public due to its real-time information sharing capabilities. Twitter provides two important features for information dissemination: Tweets and Retweets. Tweets allow users to write messages that can be instantly shared. Each tweet can contain text, media such as images, videos, or URLs. Retweets allow users to repost someone else's tweet and distribute it to their own followers. The Retweet feature is considered an effective way to spread information, as a high number of retweets indicates that the information in the tweet is spreading quickly and widely. This research aims to predict retweets based on several features: User-Based Feature, Content-Based Feature, and Time-Based Feature. The classification method used is Artificial Neural Network, which is optimized using a Nature-Inspired Algorithm called Bat Algorithm. The evaluation results of this study show an accuracy of 86%, precision of 87.8%, recall of 93.6%, and F1-score of 90.6% without imbalance class handling. Under Undersampling condition, the accuracy is 80.8%, precision is 91.0%, recall is 81.4%, and F1-score is 85.9%. Under Oversampling condition, the accuracy is 82.4%, precision is 89.6%, recall is 85.6%, and F1-score is 87.5%. These results indicate that using user-based, content-based, and time-based features, applying Artificial Neural Network classification method, and optimizing hyperparameters using Bat Algorithm are effective in predicting retweets.

Keywords: Retweet; Artificial Neural Network; Bat Algorithm; Undersampling; Oversampling

INTRODUCTION

Twitter is a social media platform where users can share and discuss various topics, including opinions, interests, and perspectives on events ranging from education and sports to politics (Kushwaha et al., 2020). Twitter is one of the most popular platforms for information dissemination due to its ease of sharing real-time information (Suh et al., 2010). On Twitter, there is a feature called Tweet that allows users to write messages that can be instantly disseminated. With a simple interface and a limit of 280 characters per message, Twitter has become a system for obtaining real-time information (Daga et al., 2020). Additionally, there is a retweet feature that allows users to repost a tweet and share it with their followers. The retweet function provided by Twitter is considered a key mechanism for spreading information among users (Zhang et al., 2016). By understanding the reasons why users retweet a particular tweet or predicting whether a tweet will be retweeted, we can gain insights into how information is disseminated on social media networks (Firdaus et al., 2016).

On Twitter, there is a feature called Tweet that allows users to write messages that can be instantly disseminated. With a simple interface and a limit of 280 characters per message, Twitter has become a system for obtaining real-time information (Daga et al., 2020). Additionally, there is a retweet feature that allows users to repost a tweet and share it with their followers. The retweet function provided by Twitter is considered a key mechanism for spreading information among users (Zhang et al., 2016). By understanding the reasons why users retweet a particular tweet or predicting whether a tweet will be retweeted, we can gain insights into how information is disseminated on social media networks (Firdaus et al., 2016).

*Muhammad Rafi Rahadian

Hamidan Amarullah Purwaatmaja Ash-Shidiq EFSA conducted a study in 2021 that implemented a retweet prediction system using ANN classification method based on User-Based and Content-Based features. The research results showed that the use of user-based and content-based features can influence the occurrence of retweets. The best model was found to be the one that addressed class imbalance using the undersampling method with NearMiss from imblearn, resulting in an F1 Score of 86% (Efsa et al., 2021).

In this study, the focus will be on predicting the likelihood of retweets for tweets containing the keyword 'world cup' obtained from the Twitter API using user-based, content-based, and time-based features using a classification method. The classification method used in this study is Artificial Neural Network (ANN), which is inspired by the way the human biological neural system, consisting of up to 60 trillion interconnected neurons, performs pattern recognition and decision-making (Farizawani et al., 2020). The hyperparameters of the ANN will be optimized using Bat Algorithm, which is an algorithm that mimics the foraging behavior of bats called Echolocation. It is expected that using this approach, the system can achieve a high level of accuracy in predicting retweets.

LITERATURE REVIEW

Related Studies

Previous research conducted a retweet prediction system by employing the classification technique of Artificial Neural Network (ANN) with both User-Based and Content-Based features. The findings indicated that the selected features had an impact on retweet activity, leading to the conclusion that the ANN classification method achieved an accuracy level of 86%(Efsa et al., 2021).

Another research was undertaken to analyze the factors affecting the probability of a tweet being retweeted by others. In this research, they amass a dataset consisting of 74 million tweets and discovered that certain aspects, like the number of followers, followings, and the age of the twitter account, played a significant role in determining the retweet rate of tweets (Suh et al., 2010).

There is another subsequent research that has made predictions for likes and retweets on a tweet using several machine learning classification methods such as SVM, Naïve Bayes, Logistic Regression, Random Forest, and Neural Network. These methods were processed using text processing techniques including bag-of-words (TFIDF) and word embeddings (Doc2Vec). The authors obtained results indicating that the bag-of-words technique showed better performance, ranging from 10% to 15% improvement, across all machine learning methods used. (Daga et al., 2020).

Retweet

Retweet is a functionality present on Twitter that enables users to duplicate or repost information. When users discover an engaging tweet authored by someone else and wish to share it with their own followers, they have the option to retweet the tweet by clicking the retweet button provided or by copying the message and often including indicators like RT or Via followed by the original author's username in the format @username (Suh et al., 2010). Retweet is seen as an effective way to spread information, and the count of retweets serves as a measure of a tweet's popularity. The higher the number of retweets a tweet receives, the faster the information within that tweet is disseminated (Kwak et al., 2010).

Retweet Features

Researchers used three different types of features: User-based Features, Content-based Features, and Time-based Features, obtained from the study conducted by Thi Bich Ngoc Hoang et al. (Hoang & Mothe, 2018). The features used in the research are categorized as follows, as shown in Table 1.

Table 1 Retweet Features

Feature Type	Feature Name	Data Type	Description
User Based Features	Total_of_tweets	Numerical	Represents the total number of tweets posted by the user with a numeric data type.
	No_of_favourite	Numerical	It indicates the number of tweets liked by the user with numerical data type.
	No_of_followees	Numerical	This denotes the number of people followed by the user with a numerical data type.
	No_of_follower	Numerical	This refers to the number of people following the user with a numerical data type.
	Age_of_account	Numerical	It represents the number of days since the user's account was created, expressed as a numerical data type.

*Muhammad Rafi Rahadian

	Username_len	Numerical	This denotes the length of the user's username, represented as a numerical data type.
	Aver_tweets_per_day	Numerical	It signifies the average number of tweets posted by the user per day, represented as a numerical data type.
Content Based Features	has_uppercase	Boolean	Indicates whether the tweet contains all capital letters.
	has_number	Boolean	ndicates whether the tweet includes any numerical digits.
	has_exclamation	Boolean	Indicates whether the tweet contains an exclamation mark.
	has_url	Boolean	Indicates whether the tweet includes a URL.
	has_hashtag	Boolean	Indicates whether the tweet contains a hashtag.
	opt_length	Boolean	Indicates whether the content length falls within the range of 70 to 100 characters.
	text_length	Numerical	Represents the length of the tweet's content as a numerical value.
	has_image	Boolean	Indicates whether the tweet contains an image.
	has_video	Boolean	Indicates whether the tweet contains a video.
	has_rt	Boolean	Indicates whether the tweet includes the abbreviation "RT"
	contain_user_mentioned	Boolean	Indicates whether the tweet mentions other users.
Time Based Features	is_is_posted_in_noon	Boolean	Indicates whether the tweet was posted during the midday hours, around 11 am to 1 pm.
	is_posted_in_eve	Boolean	Indicates whether the tweet was posted in the evening, approximately between 6 pm to 9 pm.
	is_posted_on_weekend	Boolean	Indicates whether the tweet was posted on a weekend.
	is_posted_on_holiday	Boolean	Indicates whether the tweet was posted on a holiday.

Artificial Neural Network

Artificial Neural Network is a technique or approach to information processing inspired by the functioning of the biological neural system, particularly in the human brain, in processing information. ANN is designed in a similar way to the human brain, with interconnected neuron nodes (Dastres & Soori, 2021). As shown in Figure 1, the fundamental structure of an Artificial Neural Network (ANN) comprises three layers: the Input Layer, the Hidden Layer, and the Output Layer. The Input Layer is linked to the Hidden Layer, which, in turn, is connected to the Output Layer. The Input Layer receives raw data as input to the neural network. The Hidden Layer's activation is influenced by both the Input Layer's activity and the weights connecting the Input Layer and the Hidden Layer. Similarly, the Hidden Layer's activity affects the output of the neural network through the weights between the Hidden Layer and the Output Layer. (Wasukar, 2014). Artificial Neural Network has advantages such as the capability to handle and process information even when it is not complete or lacking certain details and tolerate errors in data (Mijwel, 2018) .

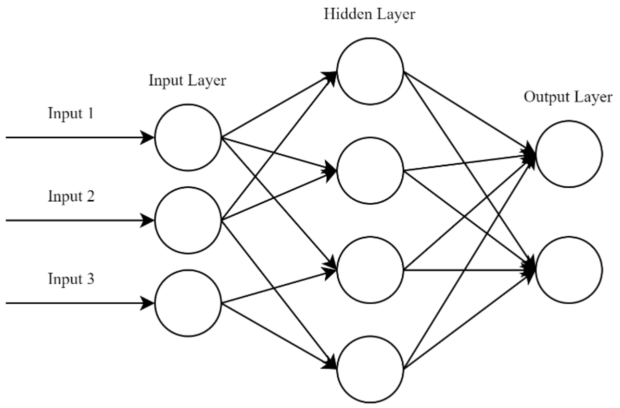

Fig. 1 Artificial Neural Network Structure

*Muhammad Rafi Rahadian

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12708
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Bat Algorithm

Is a Nature Inspired Algorithm which is an algorithm inspired by nature. The Bat Algorithm was first developed in 2010 by Xin-She Yang (Yang, 2010). The Bat Algorithm has a working concept inspired by the foraging behavior of micro bats, namely echolocation. Echolocation is a kind of sonar mechanism: bats, especially microbats, create loud, short sound pulses and find out the distance to an object by using the echoes that are played back to their ears (Zebari et al., 2020). The Bat Algorithm was developed with three key ideal rules. Firstly, all bats utilize echolocation to sense and estimate distances, allowing them to distinguish between food or prey. Secondly, each bat randomly flies at specific positions and frequencies. The loudness level and pulse rate of their echolocation calls can vary depending on their proximity to the target. When the Pulse Rate increases and the echolocation noise level decreases, it indicates that the bat is close to the target (Yang, 2010). There are mathematical formulas for updating the location x^t and velocity v^t as follows:

$$f_i = f_{min} + (f_{max} - f_{min})\beta, \quad (1)$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_*)f_i, \quad (2)$$

$$x_i^t = x_i^{t-1} + v_i^t, \quad (3)$$

Where $\beta \in [0,1]$ is a random vector drawn from a uniform distribution. Furthermore, the loudness and pulse emission rates may undergo variations during iterations. The equations for updating them are as follows:

$$A_i^{t+1} = \alpha A_i^t, \quad (4)$$

$$r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)], \quad (5)$$

Where $0 < \alpha < 1$ dan $\gamma > 0$ are constants (Yang, 2010). The Bat Algorithm was chosen as the optimization algorithm because the Bat Algorithm is a promising metaheuristic algorithm because its efficiency in dealing with various optimization problems in various fields has been proven to be good (Shehab et al., 2022) .

RandomUnderSampler

RandomUnderSampler is a function available in the Imbalanced-learn library used to address class imbalance in a dataset. Class imbalance occurs when the number of samples in each class is unbalanced or disproportionate. This function works by reducing the number of samples from the majority class (the class with more samples) to equalize the sample count with the minority class (the class with fewer samples) (Lemaître et al., 2017).

SMOTE

SMOTE, which stands for Synthetic Minority Oversampling Technique, is a groundbreaking technique in the research community for addressing imbalanced classification problems. The fundamental concept of SMOTE involves oversampling by generating synthetic instances in the feature space, using the instance itself and its K-nearest neighbors. This approach is designed to prevent overfitting and aid the classifier in effectively identifying decision boundaries between different classes. By creating synthetic instances based on the existing data, SMOTE helps to balance the class distribution and improve the performance of classifiers in handling imbalanced datasets. (Pradipta et al., 2021).

METHOD

In this study, the researchers developed a system that uses artificial neural network methods to predict retweets on Twitter data with the keyword 'world cup' using features from user-based, content-based, and time-based aspects. The system will be optimized using the Bat Algorithm. The design of the system built for this research can be seen in Figure 2.

*Muhammad Rafi Rahadian

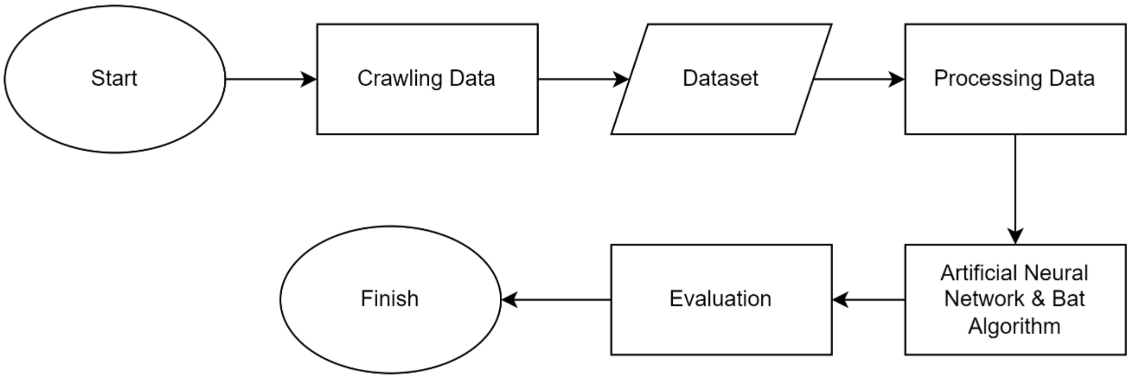

Fig. 2 Artificial Neural Network Structure

Crawling Data

Data tweets were obtained through a crawling process on the Twitter application using the Netlytic platform. The result was 2500 English-language tweets that used the keyword "world cup" within the time range starting from July 9, 2023, to July 16, 2023.

SMOTE

The tweet data obtained through the crawling process already has several features within it. The researchers made some changes, which involved removing irrelevant features and adding some necessary features according to the User Based Feature, Content Based Feature, and Time Based Feature requirements, as shown in Table 2.

Table 2. User Based, Content Based, Time Based Features Used

Feature Code	Features Name	Data Type
F1	No_of_favourite	Numeric
F2	Total_of_tweets	Numeric
F3	No_of_followees	Numeric
F4	No_of_follower	Numeric
F5	age_of_account	Numeric
F6	username_len	Numeric
F7	Aver_tweets_per_day	Numeric
F8	has_uppercase	Boolean
F9	has_number	Boolean
F10	has_exclamation	Boolean
F11	has_url	Boolean
F12	has_hashtag	Boolean
F13	opt_length	Boolean
F14	text_length	Numeric
F15	has_image	Boolean
F16	has_video	Boolean
F17	has_RT	Boolean
F18	Contain_user_mentioned	Boolean
F19	is_posted_in_noon	Boolean
F20	is_posted_in_eve	Boolean
F21	is_posted_on_weekend	Boolean
F22	is_posted_on_holiday	Boolean

Data Processing

After obtaining the data and determining the features, data processing is performed to prepare it for use in the model. One of the steps taken is to check for class imbalance in the acquired data. Class 0 indicates tweets that are not retweeted, while class 1 indicates retweeted tweets. In the obtained data, there is an imbalance in the classes, where class 0 has 1828 data, while class 1 only has 672 data. The result of checking this class imbalance can be seen in the visualization displayed in Figure 3. The solution for handling class imbalance is to divide the testing scenario into 3 scenarios: the first scenario is without handling class imbalance, the second scenario is to perform undersampling using the RandomUnderSampler method, and the third scenario is to perform oversampling using the SMOTE method.

*Muhammad Rafi Rahadian

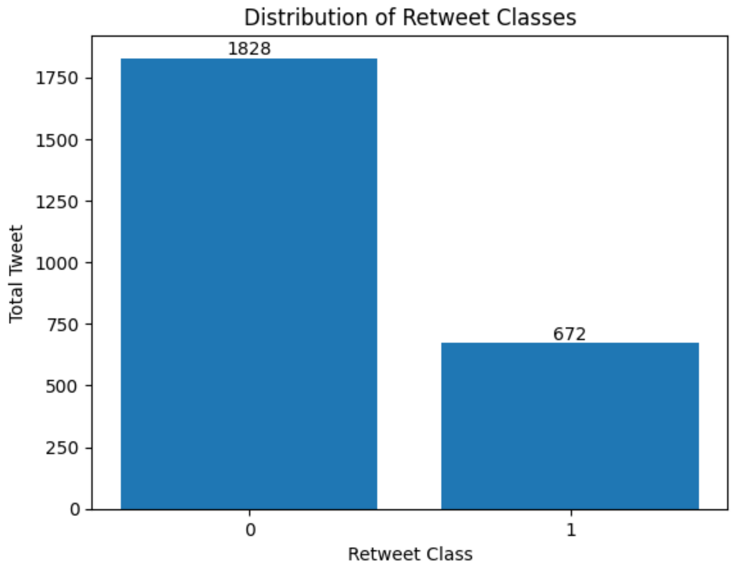

Fig. 3 Retweet Class Distribution Visualization

RESULT

This study uses a dataset consisting of 2500 collected data. The dataset includes three main features: User-Based Feature, Content-Based Feature, and Time-Based Feature. Evaluation is conducted to identify the best combination of methods and techniques that yield the highest performance in three different testing scenarios.

Testing Scenarios

In the context of an imbalanced class distribution, the researcher implemented solutions to address this issue. The solutions involved resampling the data to achieve class balance. The resampling methods used by the researcher were undersampling (randomly removing samples from the majority class) and oversampling (replicating samples from the minority class). Therefore, the researcher divided the testing scenarios into three scenarios.

Testing Scenario 1

In test scenario 1, the dataset used is the dataset that is still in an imbalanced class condition.

Testing Scenario 2

In test scenario 2, the dataset used is a dataset that has performed imbalanced class handling with the undersampling method using the RandomUnderSampler. After the undersampling technique is carried out, the retweet class distribution becomes 533 for each of the majority and minority classes as shown in Figure 4.

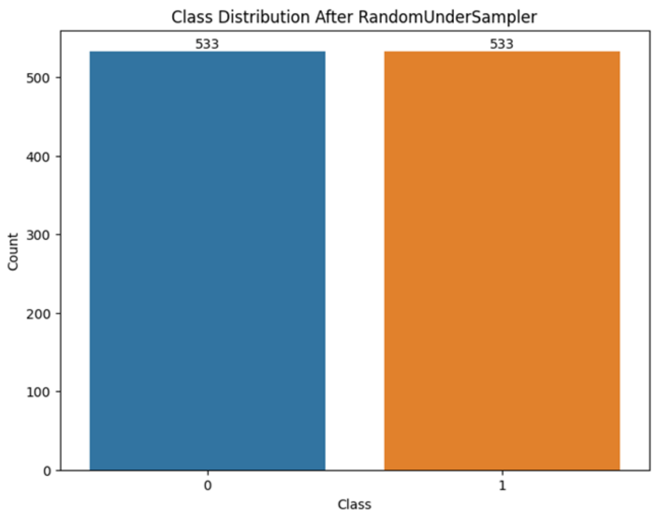

Fig. 4. Visualization of Retweet Class Distribution After Undersampling Process

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12708
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Testing Scenario 3

In test scenario 3, the dataset used is a dataset that has performed Imbalanced Class Handling with the oversampling technique using SMOTE. After the oversampling technique was carried out, the retweet class distribution became 1467 for each of the majority and minority classes as shown in Figure 5.

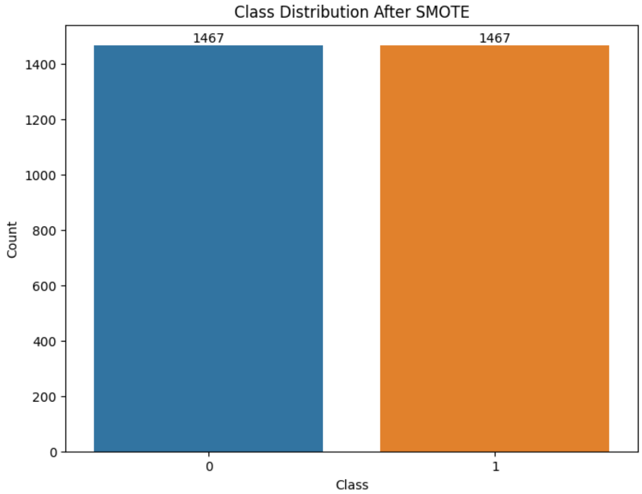

Fig. 5. Visualization of Retweet Class Distribution After Oversampling Process

Results of Scenario 1 Testing

In the first test, the dataset with class imbalance was used. The data was divided into an 80:20 ratio for testing purposes. The objective of this test was to find accuracy, precision, recall, and F1-score values for the dataset using user-based, content-based, and time-based features. The test was repeated 5 times for each ANN model, including the default model and the best model optimized using the Bat Algorithm. This approach was performed repeatedly to ensure consistent and well-tested results.

Table. 3. Results of Scenario 1

	Default Model	Best Model
Accuracy	84.2%	86.0%
Precision	86.7%	87.8%
Recall	92.2%	93.6%
F1-Score	89.4%	90.6%

Table 3 shows the results obtained from the testing of Scenario 1 for the default ANN model with an 80:20 dataset split. The results were quite good, with an accuracy of 84.2%, precision of 86.7%, recall of 92.2%, and F1-score of 89.4%.

Table 4. Best parameter of Scenario 1

Best Parameter	
Hidden_layer_sizes	(28, 28, 28, 28)
Activation	*tanh*
Solver	*adam*
Alpha	1.85e-05
Learning_rate_init	0.0018

Table 4 presents the results of Hyperparameter tuning obtained for finding the best ANN model optimized with the Bat Algorithm. The "hidden_layer_sizes" parameter determines the number of hidden layers and the number of neurons in each hidden layer. In Scenario 1, there are 4 hidden layers, each with 28 neurons. The "activation" parameter determines the activation function used in each neuron. In this case, the 'tanh' activation function is utilized. The "solver" used is 'adam'. The "alpha" value is 1.85e-05. In this model, a small alpha value is chosen. A small alpha value indicates slower weight adjustment. The "learning_rate_init" is 0.0018. By utilizing this Hyperparameter tuning, improved results were obtained compared to the default model of Scenario 1. The accuracy increased to 86%, precision increased to 87.8%, recall increased to 93.6%, and the F1-score increased to 90.6%.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12708
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Results of Scenario 2 Testing

In the second test, the dataset with imbalanced classes was handled using the undersampling method with RandomUnderSampler. The data was divided into an 80:20 ratio for testing purposes. The test was repeated 5 times for each ANN model, including the default model and the best model optimized using the Bat Algorithm. This approach was performed repeatedly to ensure consistent and well-tested results.

Table 5. Results of Scenario 2 with Undersampling using RandomUnderSampler

	Default Model	Best Model
Accuracy	79.2%	80.8%
Precision	89.5%	91.0%
Recall	80.6%	81.4%
F1-Score	84.8%	85.9%

Table 5 presents the results obtained from the testing of Scenario 2 for the default ANN model with an 80:20 dataset split. The results show an accuracy of 79.2%, precision of 89.5%, recall of 80.6%, and F1-score of 84.8%.

Table 6. Best Parameter of Scenario 2

Best Parameter	
Hidden_layer_sizes	(16, 16, 16, 16)
Activation	*tanh*
Solver	*adam*
Alpha	8.80e-05
Learning_rate_init	0.0013

In Scenario 2, the best model with hyperparameter tuning had 4 hidden layers, each with 16 neurons. The activation function used was 'relu', and the solver used was 'adam'. The alpha value was 8.80e-05, and the learning_rate_init (initial learning rate used during model training) was 0.0013. By utilizing hyperparameter tuning, the results improved compared to the default model in Scenario 2. The accuracy increased to 80.8%, precision increased to 91.0%, recall increased to 81.4%, and the F1-score increased to 85.9%.

Results of Scenario 3 Testing

In the third test, the dataset with imbalanced classes was handled using the oversampling method with SMOTE. The data was divided into an 80:20 ratio for testing purposes. The test was repeated 5 times for each ANN model, including the default model and the best model optimized using the Bat Algorithm. This approach was performed repeatedly to ensure consistent and well-tested results.

Table 7. Results of Scenario 3 with Oversampling using SMOTE

	Default Model	Best Model
Accuracy	82.2%	82.4%
Precision	90.7%	89.6%
Recall	83.9%	85.6%
F1-Score	87.2%	87.5%

Table 7 presents the results obtained from testing scenario 1 for the default ANN model with a dataset split of 80:20, yielding an accuracy of 82.2%, precision of 90.7%, recall of 83.9%, and an F1-score of 87.2%.

Table 8. Best Parameter of Scenario 3

Best Parameter	
Hidden_layer_sizes	(44, 44, 44, 44)
Activation	*relu*
Solver	*adam*
Alpha	9.05e-05
Learning_rate_init	0.0062

In scenario 3, there are 4 hidden layers, each consisting of 44 neurons. The activation function used is 'relu'. The solver used is 'adam'. The alpha value used is 9.05e-05. The learning_rate_init, which is the initial learning rate used during model training, has a value of 0.0062. By using this Hyperparameter tuning, improved results were obtained compared to the default model in scenario 3. The accuracy achieved is 82.4%, precision is 89.6%, recall is 85.6%, and the F1-score for the best model is 87.5%.

DISCUSSIONS

Based on the results obtained from all scenarios, it was found that using the same split data obtained quite good results. For the default ANN model without an imbalance class handler, it gets quite good results by achieving 79.2% accuracy, 89.5% precision, 80.6% recall, and 84.8% F1-score. The best model that uses the ANN model is optimized by the Bat Algorithm by using the best parameters that have been found previously to experience an increase in results. Accuracy increased to 86%, Precision increased to 87.8%, Recall increased to 93.6%, and F1-score increased to 90.6%. When the imbalance-class handler is carried out using the undersampling method using the RandomUnderSampler and oversampling using SMOTE, the results are quite good for the default model and the best model, but because the F1-score is the metric used when there is an imbalance class, the f1-score in the undersampling scenario and the oversampling turned out to get results that were no greater than the results before carrying out the imbalance class handler. One of the factors that might influence this is the loss of information. When the imbalance-class handler is carried out using the undersampling method using the RandomUnderSampler and oversampling using SMOTE, the results are quite good for the default model and the best model, but because the F1-score is the metric used when there is an imbalance class, the f1-score in the undersampling scenario and the oversampling turned out to get results that were no greater than the results before carrying out the imbalance class handler. One of the factors that might influence this is the loss of information.

CONCLUSION

In this study, the researcher can conclude that the use of user-based features and content-based features can influence retweet activity on Twitter. By applying the Artificial Neural Network (ANN) classification method and optimizing hyperparameters using the Bat Algorithm, the research found that the best model, which used the ANN optimized with the Bat Algorithm and implemented hyperparameter tuning, consistently outperformed the default model in three testing scenarios with the same data split. The study also discovered that in Scenario 1, where no imbalance class handling was used, the ANN model achieved an F1-Score of 90.6%, which was the highest result compared to the models using undersampling with RandomUnderSampler (F1-Score: 85.9%) and oversampling with SMOTE (F1-Score: 87.5%). Although undersampling and oversampling also yielded decent results, they remained below the model without imbalance class handling. Therefore, it can be concluded that the discovered hyperparameter tuning had a significant impact on the final results, while the use of undersampling and oversampling did not provide a significant influence in the context of this research. For future research, it is recommended to explore other optimization algorithms that can further improve the model's performance.

REFERENCES

Daga, I., Gupta, A., Vardhan, R., & Mukherjee, P. (2020). Prediction of likes and retweets using text information retrieval. *Procedia Computer Science*, *168*, 123–128.

Dastres, R., & Soori, M. (2021). Artificial Neural Network Systems. *International Journal of Imaging and Robotics (IJIR)*, *21*(2), 13–25.

Efsa, H. A. P. A.-S., Jondri, J., & Lhaksmana, K. M. (2021). Prediksi Retweet Menggunakan Fitur Berbasis Pengguna Dan Fitur Berbasis Konten Dengan Metode Klasifikasi Ann. *EProceedings of Engineering*, *8*(5).

Farizawani, A. G., Puteh, M., Marina, Y., & Rivaie, A. (2020). A review of artificial neural network learning rule based on multiple variant of conjugate gradient approaches. *Journal of Physics: Conference Series*, *1529*(2), 022040.

Firdaus, S. N., Ding, C., & Sadeghian, A. (2016). Retweet prediction considering user's difference as an author and retweeter. *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 852–859.

Hoang, T. B. N., & Mothe, J. (2018). Predicting information diffusion on Twitter–Analysis of predictive features. *Journal of Computational Science*, *28*, 257–264.

Kushwaha, A. K., Kar, A. K., & Vigneswara Ilavarasan, P. (2020). Predicting information diffusion on twitter a deep learning neural network model using custom weighted word features. Responsible design, implementation and use of information and communication technology, 456–468. In *Lecture Notes in Computer Science* (Vol. 12066, pp. 456–468).

Kwak, H., Lee, C., Park, H., & Moon, S. (2010). What is Twitter, a social network or a news media? *Proceedings of the 19th International Conference on World Wide Web*, 591–600.

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *The Journal of Machine Learning Research*, *18*(1), 559–563.

Mijwel, M. M. (2018). Artificial neural networks advantages and disadvantages. *Retrieved from LinkedIn Https//Www. Linkedin. Com/Pulse/Artificial-Neuralnet Work*.

*Muhammad Rafi Rahadian

Pradipta, G. A., Wardoyo, R., Musdholifah, A., Sanjaya, I. N. H., & Ismail, M. (2021). SMOTE for handling imbalanced data problem: A review. *2021 Sixth International Conference on Informatics and Computing (ICIC)*, 1–8.

Shehab, M., Abu-Hashem, M. A., Shambour, M. K. Y., Alsalibi, A. I., Alomari, O. A., Gupta, J. N. D., Alsoud, A. R., Abuhaija, B., & Abualigah, L. (2022). A Comprehensive Review of Bat Inspired Algorithm: Variants, Applications, and Hybridization. *Archives of Computational Methods in Engineering*, 1–33.

Suh, B., Hong, L., Pirolli, P., & Chi, E. H. (2010). Want to be retweeted? large scale analytics on factors impacting retweet in twitter network. *2010 IEEE Second International Conference on Social Computing*, 177–184.

Wasukar, A. R. (2014). Artificial neural network–an important asset for future computing. *International Journal For Research In Emerging Science And Technology*, *1*.

Yang, X.-S. (2010). A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)* (pp. 65–74). Springer.

Zebari, A. Y., Almufti, S. M., & Abdulrahman, C. M. (2020). Bat algorithm (BA): review, applications and modifications. *Int J Sci World*, *8*(1), 1–7.

Zhang, Q., Gong, Y., Wu, J., Huang, H., & Huang, X. (2016). Retweet prediction with attention-based deep neural network. *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 75–84.