JOIN (Jurnal Online Informatika)
p-ISSN: 2528-1682, e-ISSN: 2527-9165
Volume 8 Number 2 | December 2023: 252-260
DOI: 10.15575/join.v8i2.1193

Retweet Prediction Using Multi-Layer Perceptron Optimized by The Swarm Intelligence Algorithm

Jondri[1], Indwiarti[2], Dyas Puspandari[3]
[1]Data Science, Telkom University, Indonesia
[2,3]Informatics, Telkom University, Indonesia

Article Info

Article history:

Received September 22, 2023
Revised November 01, 2023
Accepted November 04, 2023
Published December 28, 2023

Keywords:

Twitter
User-based
Time-based
Content-based
Multi-layer perceptron
Swarm intelligence

ABSTRACT

Retweets are a way to spread information on Twitter. A tweet is affected by several features which determine whether a tweet will be retweeted or not. In this research, we discuss the features that influence the spread of a tweet. These features are user-based, time-based and content-based. User-based features are related to the user who tweeted, time-based features are related to when the tweet was uploaded, while content-based features are features related to the content of the tweet. The classifier used to predict whether a tweet will be retweeted is Multi Layer Perceptron (MLP) and MLP which is optimized by the swarm intelligence algorithm. In this research, data from Indonesian Twitter users with the hashtag FIFA U-20 was used. The results of this research show that the most influential feature in determining whether a tweet will be retweeted or not is the content-based feature. Furthermore, it was found that the MLP optimized with the swarm intelligence algorithm had better performance compared to the MLP.

Corresponding Author:

Jondri,
Sains Data, School of Computing, Telkom University
Jl. Telekomunikasi Terusan Buah Batu Bandung 40257
Email: jondri@telkomuniversity.ac.id

1. INTRODUCTION

Twitter is a social media that is widely used by people to interact with each other and share opinions about events. On Twitter, someone can upload tweets containing text, images, or videos with a maximum limit of 280 characters. Information on Twitter flows from followee to follower. Tweets uploaded by someone will appear on their followers' timelines. If the follower agrees with the content of the tweet, then he can retweet the tweet. This kind of action is called a retweet.

Many factors influence whether a tweet will be retweeted or not, including who is tweeting (user-based), the content of the tweet (content-based), when the tweet was uploaded (time-based), and the similarity of the content of previous tweets between followers and followees. The tweet uploader is the main actor in the dissemination of information on Twitter, therefore it is important to understand the influence of users on retweet predictions. The number of followers and followees is one of the factors that influence whether a tweet gets retweets or not[1][2][3][4]. Strong interaction between the uploader and his followers will influence his tweets to be retweeted by his followers [5][6]. A person's activeness in tweeting will influence the tweets they make to get retweets[7]. The similarity between the uploader and his followers, such as similar gender and similar location, also influences the possibility of the uploader's tweet being retweeted by his followers.[8]. Someone tends to retweet tweets whose content is similar to the tweets they usually upload [9] [8]. The content of the uploaded tweet influences whether a tweet will receive a retweet or not, such as the sentiment of the tweet [4], the arrangement of words used [10][11], the presence of words that reflect the emotions of the uploader or the presence of emoticons [12] [13]. When a tweet is uploaded will influence whether the tweet will get retweets or not, tweets uploaded at midnight will get fewer retweets than tweets uploaded in the afternoon[4].

The classification method used to predict whether a tweet will get a retweet or not, among other things, Bayesian Poisson Factorization (BPF) Model [9], Log-Linear Regression [10], Artificial Neural Network [14] [11] , Deep Neural Network [1] [3], Support Vector Machine(SVM) optimized by Cuckoo Search algorithm [5], XGBoost[2][13], Logistic Regression[8][11][7], MAKER-RIMER Prediction Model [12], SVM[11][7][4], Random Forest [11] [13] [7][4], Naive Bayes[11] [4], Probabilistic Matrix Factorization Method[6], Decision Tree[7], K-Nearest Neighbors (KNN) [7] .

In this research, we will discuss retweet predictions based on user-based, content-based and time-based features. The contribution of this research is to compare the influence of these 3 types of features and features that have a high correlation with data classes. The next contribution is that this research uses the MLP classification method with hyperparameters that are optimized using several Swarm Intelligence algorithms. These hyperparameters are the number of hidden layers, number of neurons in the hidden layer, type of activation function, solver, alpha value and initial learning rate value.

2. THE COMPREHENSIVE THEORETICAL BASIS

2.1. Multi Layer Perceptron (MLP)

Multi-Layer Perceptron (MLP) is a learning method inspired by human brain neurons. In general, an MLP consists of an input layer, several hidden layers, and an output layer[15]. The number of neurons in the input layer depends on the number of features of the data, while the number of neurons in the output layer depends on the problem to be solved using MLP. If the problem is a regression problem, the output layer simply consists of one neuron, whereas if the problem to be solved is a classification problem, then the number of neurons depends on the number of classes of data and settings chosen. An MLP consists of 5 input neurons, 2 The hidden layer containing 3 neurons and the output layer containing 1 neuron can be seen in Figure 1.

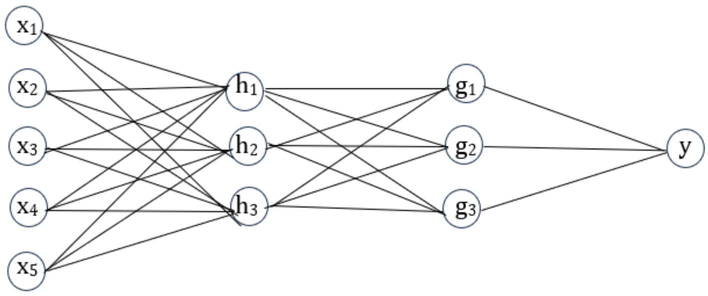


Figure 1. Multi-Layer Perceptron (MLP) with 2 hidden layers with 3 neurons in each layer

Input data x_1, x_2, x_3, x_4, x_5 will be multiplied by the weight then added, we get

$$v_j = \sum_{i=1}^{5} w_{ij}\, x_i + b_j \quad , \; j = 1,2,3 \qquad (1)$$

Where w_{ij} is the weight between $x_i, i = 1,2,3,4,5$ with $h_j, j = 1,2,3$ while $b_j, j = 1,2,3$ is the bias of j[th] neuron in the first hidden layer. The result of this sum will be input for the activation function f , we get

$$h_j = f(v_j)\,, j = 1,2,3 \qquad (2)$$

The output from the first hidden layer will be input for the neurons in the second hidden layer,

$$g_j = f(\sum_{i=1}^{3} w_{ij}^{1}\, h_j)\,, j = 1,2,3 \qquad (3)$$

where w_{ij}^{1} is the weight between the first hidden layer neurons and the second hidden layer neurons. Next, the output from the neurons in the second hidden layer will become input for the output neurons, so that we get

$$y = f(\sum_{i=1}^{3} w_i^{2}\, g_j)\,, j = 1,2,3 \qquad (4)$$

Where w_i^2 is the weight between the second hidden layer neurons and the output layer neurons. There are several activation functions that can be used, namely, threshold, linear, sigmoid, hyperbolic tangent, etc.

The number of neurons in the hidden layer depends on how the data is spread. If the data is separated linearly by a straight line, then there is no need for a hidden layer, as shown in Figure 2.a. If the data can be separated by d straight lines, then d neurons are needed in the hidden layer, as shown in Figure 2.b.

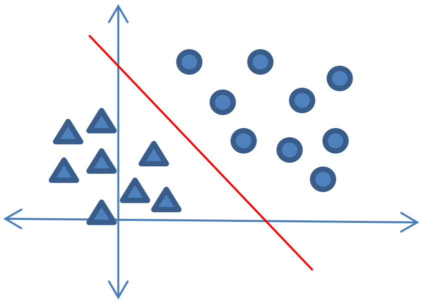


Figure 2.a. Data is linearly separated

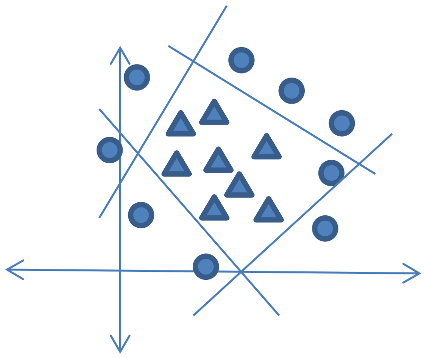


Figure 2.b The data is separated by 4 straight lines

If the number of attributes is only 2 or 3, then a plot can be made of the data so that the MLP architecture can be decided which is appropriate for the data. In real problems, the number of attributes of the data exceeds 3, so a plot cannot be made from the data. This requires trial and error to determine the number of hidden layers and the number of neurons in each hidden layer. Besides the number of hidden layers and the number of neurons in the hidden layer, there are other hyperparameters that influence the performance of an MLP. These hyperparameters are the activation function, optimizer, algorithm, and learning rate value [16]. The range of hyperparameter values can be seen in table 1.

Table 1. List of hyperparameters and their value limits

Hyperparameter	Range of values
Number of hidden layers	{1,2,3,4,5}
Number of neurons in the hidden layer	$\{x \mid 10 \leq x \leq 50, x \in Z\}$
Activation function	{relu,tanh}
Solver	{adam, lbfgs}
Alpha Value	(0.00001, 0.0001)
Initial Learning rate	(0.001, 0.01)

There are thousands of possible combinations of these hyperparameters, making it impossible to manually select the best hyperparameter combination. To overcome this, in this research, a swarm intelligence algorithm was used to select the best combination of hyperparameters.

2.2. Swarm Intelligence

Swarm intelligence is an artificial intelligence algorithm that imitates the intelligence of a group of animals, such as ants, fireflies, bats, gray wolves and others. The working principle of this algorithm is to update the position of each individual so that their fitness is better. In general, the intelligent swarm algorithm is as follows [17] :

a. Initiation of initial position of particle/individual
   A number of N individuals are selected and their positions are determined randomly in the search space.
b. Evaluate the fitness value of the particle/individual
   At each particle/individual position, calculate the fitness value. The position of a point represents an MLP architecture. The fitness value of the individual in this position is the F1 score value from the testing data.
c. Position updates
   The position of the particle/individual is updated with certain rules so that the fitness value of the particle/individual gets better.
d. Determines the stop condition.
   The process is repeated until the stop condition is met. The stopping condition can be that the maximum iteration is reached or if the position of the individual/particle does not change much between two consecutive iterations.

The swarm intelligence algorithms used in this research are Artificial Bee Colony(ABC) [18], Bacterial Foraging Optimization(BFO) [19], Hybrid Bat Algorithm(HBA)[20], Cat Swarm

Optimization(CSO)[21], Cuckoo Search(CS)[22], Firefly Algorithm(FA) [23], Grey Wolf Optimizer(GWO)[24], Particle Swarm Optimization(PSO)[25]. These algorithms will be used in the hyperparameter tuning of MLP.

2.3. Retweet

Retweet is a facility provided by Twitter where someone can re-upload other people's tweets. The mechanism by which retweets occur starts from a followee uploading a tweet. This tweet will appear on the follower's timeline. If the follower agrees with the content of the tweet, then he will retweet the tweet. Not all tweets get many retweets. There are even tweets that don't get any retweets at all. There are many factors that influence a tweet to get retweets. In this research, the influence of the features of the tweet uploader (user based), the content of the tweet (content based), and when the tweet was made (time based) on the number of retweets obtained were studied. The complete features used can be seen in table 2[4].

Table 2. User based, content based, and time based features

Feature	Description	Data Type
User Based	Total tweets uploaded by users	
UB_1		numeric
UB_2	The number of followers of the user	numeric
UB_3	The number of followee of the user	numeric
UB_4	User account age since the account was created	numeric
UB_5	Number of user tweets liked	numeric
UB_6	Average likes received by users every day	numeric
UB_7	Average tweets uploaded by users every day	numeric
UB_8	Length of user name	Numeric
Content Based		
CB_1	Tweets containing location	Boolean
CB_2	Tweets containing the name of the organization	Boolean
CB_3	The tweet contains the name of the television show	Boolean
CB_4	The contents of tweets are grouped based on sentiment	{positif, netral, negative}
CB_5	Tweet contains video	Boolean
CB_6	Tweets contain images	Boolean
CB_7	Tweets contain uppercase letters	Boolean
CB_8	Tweets contain numbers	Boolean
CB_9	Tweets contain exclamation points	Boolean
CB_10	Tweets mention a user name	Boolean
CB_11	Tweet contains URL	Boolean
CB_12	Tweets contain hashtags	Boolean
CB_13	Tweets are between 70 and 100 characters long	Boolean
CB_14	Length of tweet body text	Numeric
Time Based		
TB_1	The tweet was made on a holiday	Boolean
TB_2	Tweets were made from 11 am - 1 pm	Boolean
TB_3	Tweets are made from 5 pm – 9 pm	Boolean
TB_4	The tweet was made over the weekend	Boolean

2.4. Handling Unbalanced Data and Determining System Performance

In many two-class classification problems there are cases where the data is not balanced. The data in one class dominates the other classes. This causes the proposed machine learning method to focus on learning on the majority class. To overcome this, a process was carried out to handle this imbalanced data. There are two methods that can be used, oversampling and undersampling. The oversampling method is a way to deal with unbalanced data by adding data to the minority class so that the amount of data in both classes is equal [26]. The undersampling method is a way to deal with unbalanced data by reducing the data in the majority class so that the data for both classes is balanced [9]. Classification performance is measured by calculating accuracy, precision, recall, and f1 score [27]. In the case of imbalanced data, it is not appropriate to measure performance using accuracy, because the results are influenced by the majority class [28].

3. METHOD

Retweet prediction in this research was carried out in 4 stages, namely crawling Twitter data with the keywords FIFA U20 World Cup, getting features from tweets, tuning MLP hyperparameters, and analyzing the results as seen in figure 3.

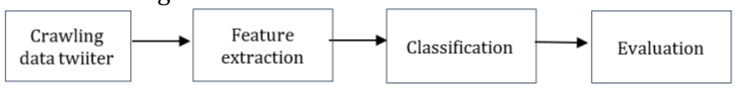


Figure 3. Diagram of the retweet prediction process

The first step taken was to crawl tweet data on the topic of the FIFA U20 World Cup. The dataset was taken from the netlytic.org website from May 30 2023 to June 8, 2023, and was limited to only using Indonesian. The data obtained was 870 tweets. These tweets are then grouped into two classes, namely class 0 (tweets do not get retweets) and class 1 (tweets get retweets). From this grouping, 582 class 0 data were obtained, and 288 class 1 data were obtained. After that, we look for user-based, content-based, and time-based features of each tweet. The classification process is carried out based on 3 scenarios in the dataset and 2 scenarios in the classifier. The scenarios in the dataset are unbalanced original data, undersampling data, and oversampling data. Meanwhile, the scenario in the classifier is to compare performance results using the default MLP Python library with MLP which is hyperparameter tuned using the Swarm Intelligence algorithm.

4. RESULT AND DISCUSSION

4.1. Analysis of Features that Influence Retweets

To see the features that influence retweets, experiments were carried out using all features, namely user-based features, content-based features, and time-based features. There are two classifiers used, namely the default MLP in the Sklearn Python library and MLP with hyperparameters optimized using swarm intelligence. MLP optimized with swarm intelligence used is MLP optimized with Artificial Bee Colony (MLP-ABC), MLP optimized with Bacterial Foraging Optimization (MLP-BFO), MLP optimized with Hybrid Bat Algorithm (MLP-HBA), MLP optimized with Cat Swarm Optimization (MLP-CSO), MLP optimized with Cuckoo Search (MLP-CS), MLP optimized with Firefly Algorithm (MLP-FA), MLP optimized with Gray Wolf Optimizer (MLP-GWO), and MLP which is optimized with Particle Swarm Optimization (MLP-PSO). In Table 3, the F1 score values for each of these classifiers are visible. The data used is original data.

Table 3. F1 score value using various combinations of features

Feature	MLP	MLP-ABC	MLP-BFO	MLP-HBA	MLP-CSO	MLP-CS	MLP-FA	MLP-GWO	MLP-PSO	Average
All Feature	0,54	0,66	0,7	0,65	0,71	**0,74**	0,7	0,71	0,64	0,67
User-Based	0,55	0,63	0,66	0,65	0,65	0,65	**0,75**	0,71	0,72	0,66
Time- Based	0,41	0,41	0,41	0,41	0,41	0,41	0,41	0,41	0,41	0,41
Content-based	0,67	0,74	0,74	0,72	0,69	**0,75**	0,69	0,74	0,72	**0,72**
Average	0,5425	0,61	0,6275	0,6075	0,615	0,6375	0,6375	**0,6425**	0,6225	

Based on table 3, it can be seen that the best feature used to predict whether a tweet will get a retweet or not is the content based feature with an average F1 score of 0.72. Meanwhile, the best classifier when using all features is MLP-CS with an F1 score of 0.74, MLP-FA when using user based features with an F1 score of 0.75, MLP-CS when using content based features with an F1 score of 0.75. Meanwhile, MLP-GWO is a classifier with the highest average F1 score for various feature combinations with an average F1 score of 0.6425.

4.2. The Effect of Undersampling and Oversampling

The distribution of the data used in this research is not balanced, where class 0 is more numerous than class 1, so undersampling and oversampling methods are used on the training data. The results of the F1 score for handling unbalanced data using the undersampling method can be seen in Table 4, while using the oversampling method can be seen in Table 5.

Table 4. F1 Score value by applying the undersampling method to the dat

Fitur	MLP	MLP-ABC	MLP-BFO	MLP-HBA	MLP-CSO	MLP-CS	MLP-FA	MLP-GWO	MLP-PSO	Average
All Feature	0,49	0,66	0,58	**0,67**	0,64	**0,67**	0,59	0,64	0,63	0,62
User-Based	0,48	0,64	0,63	0,57	0,64	**0,65**	0,61	0,64	0,64	0,61

Fitur	MLP	MLP-ABC	MLP-BFO	MLP-HBA	MLP-CSO	MLP-CS	MLP-FA	MLP-GWO	MLP-PSO	Average
Time-Based	0,54	0,53	0,54	0,55	0,54	0,56	0,54	0,53	0,54	0,54
Content- based	0,66	**0,72**	0,67	0,65	0,66	0,7	0,65	0,7	0,6	**0,67**
Average	0,5425	0,6375	0,605	0,61	0,62	**0,645**	0,5975	0,6275	0,6025	

Based on Table 4, it can be seen that after undersampling, the best feature used to predict whether a tweet will get a retweet or not is the content-based feature with an average F1 score of 0.67. In fact, the average F1 score when using content-based is better than using all features. Meanwhile, the best classifier if using all features is MLP-CS and MLP-HBA with an F1 score of 0.67, MLP-CS if using user-based features with an F1 score of 0.65, MLP-ABC if using content-based features with an F1 score of 0 .72. Meantime, MLP-CS is a classifier with the highest average F1 score for various feature combinations with an average F1 score of 0.645.

Table 5 The F1 score value is by applying the oversampling method to the data

Fitur	MLP	MLP-ABC	MLP-BFO	MLP-HBA	MLP-CSO	MLP-CS	MLP-FA	MLP-GWO	MLP-PSO	Average
All Feature	0,53	0,65	0,69	0,67	0,65	0,69	0,66	0,63	**0,71**	0,65
User- Based	0,54	0,62	0,66	0,63	0,66	0,59	0,68	0,64	**0,69**	0,63
Time-Based	0,53	0,53	0,53	0,53	0,54	0,53	0,53	0,53	0,54	0,53
Content-based	0,65	0,7	0,7	0,68	0,7	0,67	0,66	**0,71**	**0,71**	**0,687**
Average	0,5625	0,625	0,645	0,6275	0,6375	0,62	0,6325	0,6275	**0,6625**	

Based on Table 5, it can be seen that after oversampling, the best feature used to predict whether a tweet will get a retweet or not is the content-based feature with an average F1 score of 0.687. In fact, the average F1 score when using content-based is better than using all features with an F1 score of 0.65. Meanwhile the best classifier if using all features is MLP-PSO with an f1 score of 0.71, MLP-PSO if using user-based features with an f1 score of 0.69, MLP-GWO and MLP-PSO if using content-based features with an f1 score of 0 .71. Meantime, MLP-PSO is a classifier with the highest average F1 score for various feature combinations with an average F1 score of 0.6625.

4.3. Analyze the effect of Hyperparameter Tuning on MLP

The average F1 score value of the default MLP classifier and the MLP whose architecture was optimized using swarm intelligence can be seen in Figure 4.

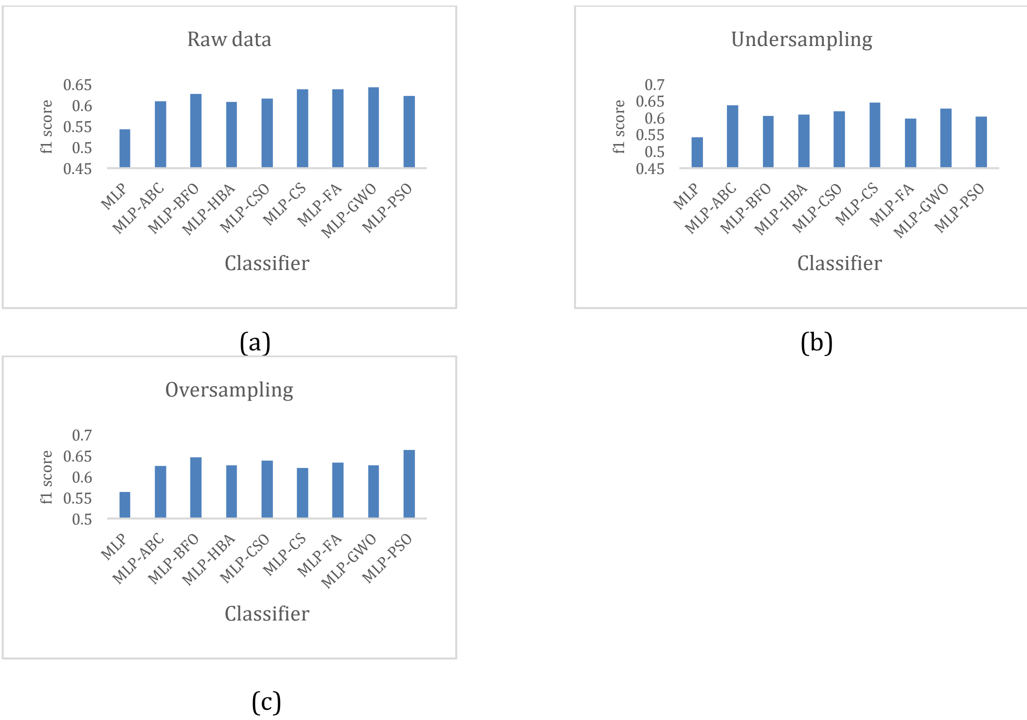
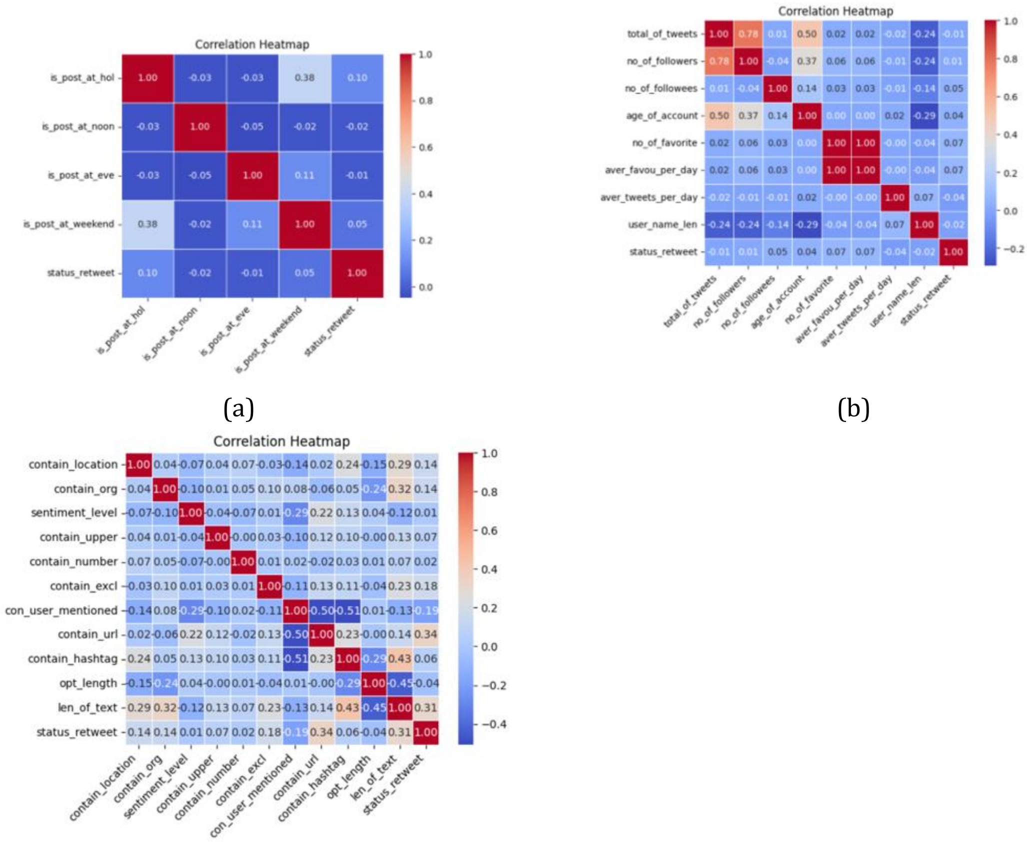


Figure 4. Comparison of the average F1 score of the default MLP classifier with MLP optimized with the Swarm Intelligence algorithm for cases of a) raw data, b) undersampling data, c) oversampling data.

From Figure 4, it can be seen that the average F1 score of the default MLP is always smaller than all MLP classifiers optimized with the Swarm Intelligence algorithm. Both for raw data and for data that has been processed using undersampling and oversampling methods.

4.4. Correlation Analysis between Features and Retweet Classes

The correlation value between attributes and output (retweet status) can be seen in Figure 5. Figure 5a shows the correlation between time-based feature attributes and output, Figure 5b shows the correlation between user-based feature attributes and output, while Figure 5b shows the correlation between user-based feature attributes and output. Figure 5c illustrates the correlation between content-based feature attributes and output. The features with the highest correlation with output (retweet status) are contain_url, len_of_text, con_user_mentioned, contain_excl, contain_location, and contain_org.



(a) (b)

(c)

Figure 5. Correlation between output and features a) time-based, b) user-based, c) content-based

The features that have the highest correlation with this output are then used as features for retweet prediction. The classifier used is the classifier with the highest average value for raw data (MLP-GWO), undersampling data (MLP-CS), and oversampling data (MLP-PSO). The F1 score values for these three classifiers can be seen in Table 6.

Table 6. The F1 score uses 6 features with the largest correlation with the output

Fitur	MLP-CS	MLP-GWO	MLP-PSO
6 Best features	0,74	0,67	0,66

The best F1 score value is obtained when using the MLP classifier with hyperparameter tuning using the Cuckoo Search algorithm (MLP-CS).

From Figure 4, it can be seen that the performance of the MLP optimized with swarm intelligence is better than the default MLP. These results are consistent for raw data, data that has undergone oversampling and undersampling processes. However, there are two problems that arise when using MLP optimized with swarm intelligence in retweet prediction. The first problem is that the hyperparameters obtained are not global optimum solutions, but local optimum values. This can be seen from the hyperparameter results obtained which will be different if the experiment is carried out repeatedly. The second problem is in terms of program execution time, MLP optimized with swarm intelligence requires much longer execution time compared to the default MLP. These two problems can be material for further research studies.

5. CONCLUSION

Hyperparameter tuning on the MLP was proven to improve the performance of the MLP. This is indicated by the F1 score value of the MLP classifier optimized with the swarm intelligence algorithm being greater than the default MLP F1 score. Furthermore, the feature that has the most influence on whether a tweet gets a retweet or not is the content-based feature. The content-based feature is related to the content of the tweet.

ACKNOWLEDGEMENTS

This research was funded by the Telkom University Center for Research and Community Service (PPM) through the Basic Research scheme. We would like to thank the Chancellor of Telkom University for his support for this research.

REFERENCES

[1] L. Wang, Y. Zhang, J. Yuan, K. Hu, and S. Cao, "FEBDNN: fusion embedding-based deep neural network for user retweeting behavior prediction on social networks," *Neural Comput Appl*, vol. 34, no. 16, pp. 13219–13235, Aug. 2022, doi: 10.1007/s00521-022-07174-9.

[2] S. Wang, Z. Li, Y. Wang, and Q. Zhang, "Machine learning methods to predict social media disaster rumor refuters," *Int J Environ Res Public Health*, vol. 16, no. 8, Apr. 2019, doi: 10.3390/ijerph16081452.

[3] J. Wang and Y. Yang, "Tweet Retweet Prediction Based on Deep Multitask Learning," *Neural Process Lett*, vol. 54, no. 1, pp. 523–536, Feb. 2022, doi: 10.1007/s11063-021-10642-3.

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

[5] T. Dai, Y. Xiao, X. Liang, Q. Li, and T. Li, "ICS-SVM: A user retweet prediction method for hot topics based on improved SVM," *Digital Communications and Networks*, vol. 8, no. 2, pp. 186–193, Apr. 2022, doi: 10.1016/j.dcan.2021.07.003.

[6] B. Jiang, Z. Lu, N. Li, J. Wu, and Z. Jiang, "Retweet prediction using social-aware probabilistic matrix factorization," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 2018, pp. 316–327. doi: 10.1007/978-3-319-93698-7_24.

[7] S. Sharma and V. Gupta, "Role of twitter user profile features in retweet prediction for big data streams," *Multimed Tools Appl*, vol. 81, no. 19, pp. 27309–27338, Aug. 2022, doi: 10.1007/s11042-022-12815-1.

[8] X. Tang, Q. Miao, Y. Quan, J. Tang, and K. Deng, "Predicting individual retweet behavior by user similarity: A multi-task learning approach," *Knowl Based Syst*, vol. 89, pp. 681–688, Nov. 2015, doi: 10.1016/j.knosys.2015.09.008.

[9] S. Wang, C. Li, Z. Wang, H. Chen, and K. Zheng, "BPF++: A Unified Factorization model for predicting retweet behaviors," *Inf Sci (N Y)*, vol. 515, pp. 218–232, Apr. 2020, doi: 10.1016/j.ins.2019.12.017.

[10] N. Y. Jalali and P. Papatla, "Composing tweets to increase retweets," *International Journal of Research in Marketing*, vol. 36, no. 4, pp. 647–668, Dec. 2019, doi: 10.1016/j.ijresmar.2019.05.001.

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

[12] L. Rivadeneira, J. B. Yang, and M. López-Ibáñez, "Predicting tweet impact using a novel evidential reasoning prediction method," *Expert Syst Appl*, vol. 169, May 2021, doi: 10.1016/j.eswa.2020.114400.

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

[14] Y. Liu, J. Zhao, and Y. Xiao, "C-RBFNN: A user retweet behavior prediction method for hotspot topics based on improved RBF neural network," *Neurocomputing*, vol. 275, pp. 733–746, Jan. 2018, doi: 10.1016/j.neucom.2017.09.015.

[15] H. X. Xiao, X. Liu, R. Yu, B. Yao, F. Zhang, and Y. Q. Wang, "Contributions of internal climate variability in driving global and ocean temperature variations using multi-layer perceptron neural network," *Advances in Climate Change Research*, vol. 13, no. 4, pp. 459–472, Aug. 2022, doi: 10.1016/j.accre.2022.06.001.

[16] M. Stankovic *et al.*, "Tuning Multi-Layer Perceptron by Hybridized Arithmetic Optimization Algorithm for Healthcare 4.0," *Procedia Comput Sci*, vol. 215, pp. 51–60, 2022, doi: 10.1016/j.procs.2022.12.006.

[17] Y. Miao, Y. Yao, X. Hong, L. Xiong, F. Zhang, and W. Chen, "Research on optimal control of HVAC system using swarm intelligence algorithms." [Online]. Available: https://ssrn.com/abstract=4372684

[18] A. Banharnsakun, "A new approach for solving the minimum vertex cover problem using artificial bee colony algorithm," *Decision Analytics Journal*, vol. 6, p. 100175, Mar. 2023, doi: 10.1016/j.dajour.2023.100175.

[19] Y. Long *et al.*, "Local Path Planning with Multiple Constraints for USV Based on Improved Bacterial Foraging Optimization Algorithm," *J Mar Sci Eng*, vol. 11, no. 3, p. 489, Feb. 2023, doi: 10.3390/jmse11030489.

[20] Q. Liu, L. Wu, W. Xiao, F. Wang, and L. Zhang, "A novel hybrid bat algorithm for solving continuous optimization problems," *Appl Soft Comput*, vol. 73, pp. 67–82, Dec. 2018, doi: 10.1016/j.asoc.2018.08.012.

[21] L. Wang and S. Han, "Improved cat swarm optimization for parameter estimation of mixed additive and multiplicative random error model," *Geod Geodyn*, vol. 14, no. 4, pp. 385–391, Jul. 2023, doi: 10.1016/j.geog.2022.10.003.

[22] P. Ong and Z. Zainuddin, "An optimized wavelet neural networks using cuckoo search algorithm for function approximation and chaotic time series prediction," *Decision Analytics Journal*, vol. 6, p. 100188, Mar. 2023, doi: 10.1016/j.dajour.2023.100188.
[23] M. Ghasemi, S. kadkhoda Mohammadi, M. Zare, S. Mirjalili, M. Gil, and R. Hemmati, "A new firefly algorithm with improved global exploration and convergence with application to engineering optimization," *Decision Analytics Journal*, vol. 5, p. 100125, Dec. 2022, doi: 10.1016/j.dajour.2022.100125.
[24] Z. Li, "A local opposition-learning golden-sine grey wolf optimization algorithm for feature selection in data classification," *Appl Soft Comput*, vol. 142, p. 110319, Jul. 2023, doi: 10.1016/j.asoc.2023.110319.
[25] Y. Zhang and X. Hou, "Application of video image processing in sports action recognition based on particle swarm optimization algorithm," *Prev Med (Baltim)*, vol. 173, p. 107592, Aug. 2023, doi: 10.1016/j.ypmed.2023.107592.
[26] V. R. S. Nastiti, Z. Sari, and B. Chintia Eka Merita, "The Implementation of Restricted Boltzmann Machine in Choosing a Specialization for Informatics Students," *Jurnal Online Informatika*, vol. 8, no. 1, pp. 36–43, Jun. 2023, doi: 10.15575/join.v8i1.917.
[27] O. G. Ajayi and J. Ashi, "Effect of varying training epochs of a Faster Region-Based Convolutional Neural Network on the Accuracy of an Automatic Weed Classification Scheme," *Smart Agricultural Technology*, vol. 3, p. 100128, Feb. 2023, doi: 10.1016/j.atech.2022.100128.
[28] P. Thölke *et al.*, "Class imbalance should not throw you off balance: Choosing the right classifiers and performance metrics for brain decoding with imbalanced data," *Neuroimage*, vol. 277, p. 120253, Aug. 2023, doi: 10.1016/j.neuroimage.2023.120253.