LOGISTIC REGRESSION WITH MULTI-SWARM OPTIMIZATION TO PREDICT DOTA 2 GAME VICTORY

Raden Hasbie Finno Hadiyanto[1], Galih Hermawan[2]

[1,2] Program Studi Teknik Informatika, Universitas Komputer Indonesia
Jl. Dipati Ukur No.112-116, Bandung 40132
E-mail : rdhasbiefh@gmail.com[1], galih.hermawan@email.unikom.ac.id[2]

ABSTRACT

Defend of the Ancients 2 or DOTA 2 is a game where the game consists of 2 teams, namely DIRE and RADIANT, each of which consists of 5 players. In this game, victory is determined by which team can destroy the ancient opponent first. Several factors can influence the outcome of a game, one of which is the team's playing style. These factors can be used as a reference to predict which team will win the game. To make predictions, this study uses the Logistic Regression method. Before the prediction is made, the optimal solution in the form of the best weight will be used to calculate the Logistic Regression calculation. To get the optimal solution, this research uses Multi-Swarm Optimization method. By combining the two methods, the optimal solution is obtained by using 3 swarms, 15 particles, 40 iterations, the search space limit (-10.10), and the particle speed limit (-1.1), where the resulting accuracy is very good, that is 95.76% for the lowest accuracy and 97.51% for the highest accuracy. These results indicate that the method used can produce optimal solutions.

Keyword: Defend of The Ancients 2, Logistic Regression, Multi-Swarm Optimization, Prediction of Victory, Game

1. INTRODUCTION

1.1 Background

Defend of the Ancients 2 or better known as DOTA 2 is a game with the MOBA genre (Multiplayer Online Battle Arena) where the game consists of 2 teams, namely DIRE and RADIANT, each of which consists of 5 players with each player playing a different character game (heroes). The two teams will destroy and defend each other's bases. The victory will be determined by which team can destroy the ancient opponent first.

In this game, several factors can influence the outcome of a game such as character selection and team play style [1]. These factors can be used as a reference to predict which team will win the match based on match data that has been done before.

Several studies have been done to predict victory in the DOTA 2 game. Zhengyao Li, Dingyue Cui, and Chen Li has been researching to predict a victory in the DOTA 2 game by comparing several algorithms based on team play style and producing Logistic Regression (LR) as the best model with an accuracy rate of 59.713% [2]. In another study, Nicholas Kinkade and Kevin Lim chose to compare character selection factors and playing styles using LR which resulted in an accuracy rate of 63% in character selection factors, and 73% in team play style factors [1]. However, the LR method does not have an analytical solution that can be used in solving linear equations to get the value of the parameters $\beta 0, \ldots, \beta k$, so the numerical optimization method is used to solve the problem [3]. Subathra and Nedunchezhian in their research succeeded in increasing the accuracy of LR by 4.98% by combining it using Particle Swarm Optimization (PSO) in the case of alias classification [4]. While in McCaffrey's research using Multi-Swarm Optimization (MSO) which is a variant of PSO, McCaffrey concluded that MSO tends to produce more optimal results and can handle optimization problems better than PSO [5].

Based on this description, this research will use the Logistic Regression method with Multi-Swarm Optimization to predict victory in the DOTA 2 game.

1.2 Purpose

Based on the description that has been explained in the background before, the purpose of the objectives in this study are as follows:

1. To implement the MSO algorithm that is used to optimize the value of the parameters $\beta 0, \ldots, \beta k$ in the LR method.
2. To measure the accuracy of the application LR algorithm with MSO optimization in predicting DOTA 2 game victory.

2. RESEARCH CONTENT

The contents of this research contain explanations that covering research methods, The International, system architecture, normalization, Multi-Swarm Optimization methods, Logistic Regression methods, and test results.

2.1 Research Method

In this research, six stages are carried out, starting from the stage of problem formulation, data collection, method analysis which includes data normalization, MSO optimization, and LR prediction, application development, system testing and finally the conclusion. The following is a schematic of the method in this study can be seen in Figure 1.

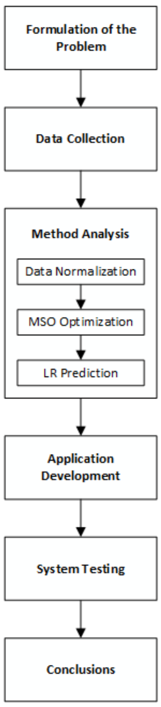

Figure 1. Research Method Scheme

2.2 The International

The International is one of the biggest tournaments in DOTA 2 games [6]. Each match result from the tournament will be used as a dataset in this research. The dataset used amounted to 1748 data, 409 data from The International 2015 tournament, 398 data from The International 2016 tournament, 540 data from The International 2017 tournament, and 401 data from The International 2018 tournament. The data used were obtained with the API provided by the *opendota* [7]. The data is obtained during the game progresses. The attributes that will be used can be seen in Table 1.

Table 1. Attributes on The International Data

Attribute	**Information**	**Attribute**	**Information**
KR	*Kill Radiant*	KD	*Kill Dire*
DR	*Dead Radiant*	DD	*Dead Dire*
AR	*Assist Radiant*	AD	*Assist Dire*
GPMR	*Gold per Minute Radiant*	GPMD	*Gold per Minute Dire*
XPMR	*Experience per Minute Radiant*	XPMD	*Experience per Minute Dire*
LHR	*Last Hit Radiant*	LHD	*Las Hit Dire*
DNR	*Denied Radiant*	DND	*Denied Dire*
GR	*Gold Radiant*	GD	*Gold Dire*

The attributes contained in Table 1 will then be represented as dimensions that will be used in the optimization process with MSO, and prediction with LR. The number of dimensions that will be used is 16 dimensions with one additional dimension, namely $\beta 0$ which will be used as the search for the best solution.

2.3 System Architecture

The application development that will be used to predict victory in the DOTA 2 game includes several processes. The first process is to normalize the input data using z-score normalization. After the input data is normalized, then the normalized data will be optimized in the MSO training process. After getting the optimal solution from the MSO training process, the process carried out next is to do LR testing, in this process, the prediction will be done using the LR method. For the whole process, it can be seen in the following block diagram in Figure 2.

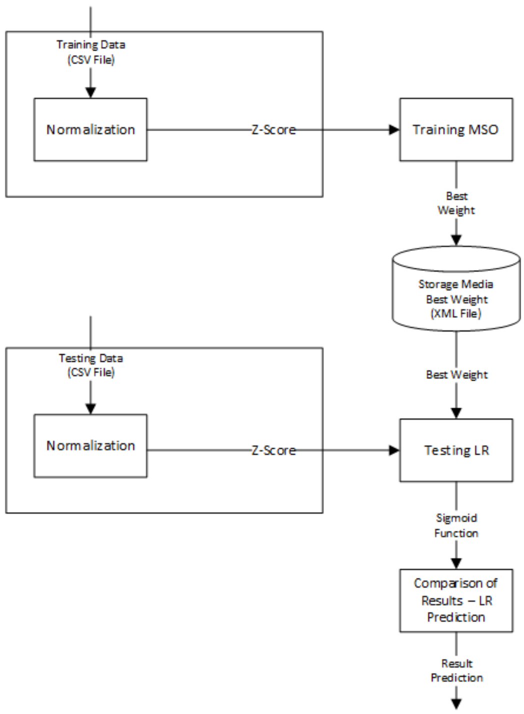

Figure 2. Block Diagram of the System

In the normalization process, the input data used is a CSV file format. The file contains data from the results of the match at The International tournament in the DOTA 2 game. The results of the normalization process will then go through the training process using MSO, this training aims to get the value of the best (optimal) solution. The optimal solution obtained, will then be saved into a file with XML format. The file that contains the best weight will then go through the testing stages using LR. At this test stage, the normalized test data will be processed to obtain prediction results using the optimal solution that has been obtained previously. The results of the predictions obtained will then be compared with the actual results on the actual data. Comparisons are made with the aim of getting the accuracy value of the method used.

2.4 Normalization

The data processing that is carried out to convert the initial data into another form that aims to maintain the range/scale of the data so that the data used is more appropriate for analysis is called the normalization process [8]. In this study, the method used to normalize is the z-score normalization. Calculations using the z-score normalization method can be done using the following equation [9]:

$$v' = \frac{v-\bar{x}}{\sigma_x} \tag{1}$$

There are several processes carried out in normalizing data. Following is a picture of the flow of the entire process carried out can be seen in Figure 3.

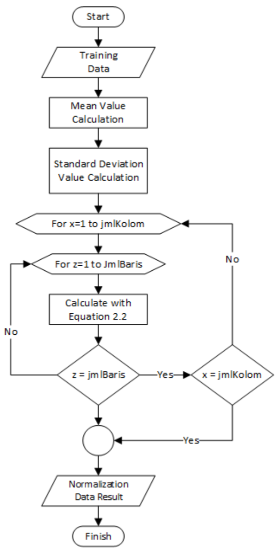

Figure 3. Flowchart of Data Normalization Process

In Figure 3, to calculate the value of the z-score, the mean and standard deviation of the data must be found first. To calculate the mean using the following equation [10]:

$$\bar{x} = \frac{\sum_{i=0}^{n} x_i}{n} \tag{2}$$

Whereas to calculate the value of the standard deviation the following equation is used [10]:

$$\sigma_x = \sqrt{\frac{\sum(x_i-\mu)^2}{N}} \tag{3}$$

If the calculation value of the standard deviation is only done on the sample data (not done on the whole data), then in the calculation the following equation is used [10]:

$$s_x = \sqrt{\frac{\sum(x_i-\bar{x})^2}{n-1}} \tag{4}$$

In this study, the process of data normalization is carried out to maintain the range/scale of each dimension in the dataset having the same weight.

2.5 Multi-Swarm Optimization

One method of optimization in machine learning that can be used to estimate a solution to a complex numerical problem is Multi-Swarm Optimization (MSO) [11]. MSO method is one of the variations are developed based on the method of Particle Swarm Optimization (PSO) [11]. In 1995, for the first time, PSO was introduced by Eberhart and Kennedy [12]. MSO is part of Swarm Intelligence. Swarm Intelligence is one of the techniques of artificial intelligence that is based on collective behavior and can regulate itself [12].

There are several parameters used in performing MSO optimization. The parameters used can be seen in Table 2 [5].

Table 2. MSO Parameters

Parameters	Value
Swarm	4
Particle	3
Maximum Iteration	5
Minimum Particle Position Limit	-10
Maximum Particle Position Limit	10
Minimum Particle Speed Limit	-1
Maximum Particle Speed Limit	1
Inertia Weight (*w*)	0.729
Cognitive Weight (*c1*)	1.49445
Social Weight (*c2*)	1.49445
Global Weight (*c3*)	0.3645

In this study, in carrying out the optimization process using MSO the following steps are used [11]:

1. Initialize random position and speed of particles.

   At this stage, the use of swarm and particle counts will be determined, as well as randomly giving values to the position and speed of the particles. To initialize random particle positions the following equation is used [11]:

$$x_i = minX + r_i(maxX - minX) \qquad (5)$$

Whereas to initialize random particle velocity the following equation is used [11]:

$$x_i = minX + r_i(maxX - minX) \qquad (6)$$

2. Evaluate the optimization of the function value of each particle.
   Evaluation of the function value on each particle is carried out to get the particle with the best function value. The method used to get the function value is to use MSE (Mean Squared Error). Following is the equation of MSE used [9]:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \check{Y}_i\right)^2 \qquad (7)$$

3. Determine *bestPartikel.*
   bestPartikel is the best position of each particle. This position is the position currently occupied by the particle.
4. Determine *bestSwarm.*
   bestSwarm is the best position of each swarm. This position is the position currently occupied by Swarm.
5. Determine *bestMulti-Swarm.*
   bestMulti-Swarm is the best position of every Multi-Swarm. This position is the position currently occupied by Multi-Swarm.
6. Update the particle speed (velocity).
   Velocity value calculation is done to determine the new position for each particle with the provisions :
   a. If v > 1, then v = 1
   b. If -1 <= v <= 1, then v = v
   c. If v < -1, then v = -1
   Following is the equation used to determine velocity values [5]:

$$v(t+1) = w.v(t) + (c_1.r_1)(p(t)-x(t)) + (c_2.r_2)(s(t)-x(t)) + (c_3.r_3)(g(t)-x(t)) \qquad (8)$$

7. Update the position of the particles.
   After doing the new velocity calculation in equation (8), the next step is to do a calculation to determine the new particle position with the provisions :
   a. If t > 10, then t = 10
   b. If -10 <= t <= 10, then t = t
   c. If t < -10, then t = -10
   Following is the equation used to update the position of particles [11]:

$$x(t+1) = x(t) + v(t+1) \qquad (9)$$

8. Doing repetition from step 2 until the criteria are met, the criteria are said to be fulfilled if it has gotten a good enough function value or if it has reached the maximum iteration limit.

The following is the overall flow that is carried out in the optimization process using the MSO method can be seen in Figures 4 and 5.

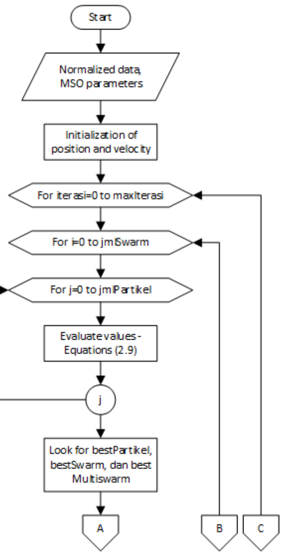

Figure 4. Flow Process of MSO Optimization

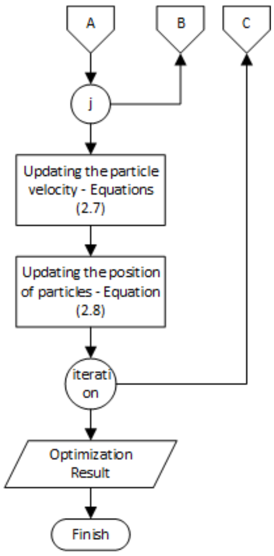

Figure 5. Flow Process of MSO Optimization - Continued

2.6 Logistic Regression

Logistic Regression (LR) is one of the algorithms in machine learning that is used to do classification [13]. This method was developed by a statistician named David Cox in 1958 [13]. The purpose of the LR classification is to create a model that can analyze a dataset that has one or more independent variables that will affect the results of the dependent variable, the results of the dependent variable can be 1/0, yes/no, and true/false [13].

In this study, the dependent variable used is won/lose. Following is the flow of the entire prediction process carried out using the LR method can be seen in Figure 6.

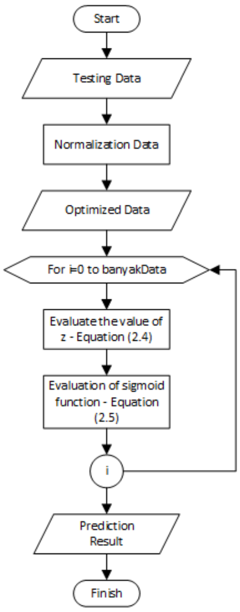

Figure 6. Flow Process of LR Prediction

In Figure 6, the dataset that has been through the stages of normalization and optimization, the dataset then will go into the next stages to evaluate the linear equation of z value and sigmoid function. Calculations in evaluating linear equations of z values can be calculated using equation [13]:

$$z = \alpha + \beta 1X1 + \ldots + \beta kXk \qquad (10)$$

Whereas the calculation of sigmoid function values can be done using equation [13]:

$$f(z) = 1/(1 + e\hat{}(-z)) \qquad (11)$$

The results of the calculation of sigmoid functions using equation (11), will produce values with a range (0, 1). The results will then be classified into two classes, namely win and lose with the provisions:

a. If $f(z) <= 0.5$, then $f(z)$ = *false* (lose)
b. if $f(z) >= 0.5$, then $f(z)$ = *true* (win)

The results of these predictions will then be compared with the actual results in the dataset to find out the accuracy values obtained from the method used.

2.7 Test Result

Tests are carried out through five stages, namely swarm number testing, particle number testing, iteration number testing, position limit testing, and speed limit testing. Each test will be carried out 10 times and the lowest average value will be taken from all the tests that have been done. After getting the best value of all parameters, then predictions will be made to the dataset using the best parameters.

2.7.1 Swarm Number Testing

Swarm testing is carried out to find out the number of swarms that need to be used to get the optimal (best) function and weight values. The function value here is the error value or error value, where the best error value is the smallest value. The number of swarms used in this test is 1, 2, 3, 4, and 5. The results of the number of swarm tests that have been carried out can be seen in Table 3.

2.7.2 Particle Number Testing

Testing the number of particles carried out to determine the number of particles that need to be used to get the optimal value of the function and weight (best). The function value here is the error value or error value, where the best error value is the smallest value. The number of particles used in this test is 3, 6, 9, 12, and 15. The results of the number of particles tested can be seen in Table 4.

Table 3. Swarm Number Testing

Total Swarm	Best Value Function of-										Average Function Value
	1	2	3	4	5	6	7	8	9	10	
1	0.0643	0.2788	0.2628	0.1439	0.0424	0.1478	0.5500	0.0304	0.2103	0.2958	0.2026
2	0.2018	0.4288	0.0384	0.0530	0.0445	0.0526	0.0298	0.1238	0.0399	0.0767	0.1089
3	0.0510	0.0224	0.0395	0.0547	0.0392	0.0237	0.0323	0.0696	0.0344	0.0304	0.0397
4	0.0371	0.0333	0.0239	0.1164	0.0347	0.0172	0.0307	0.0250	0.0729	0.0592	0.0450
5	0.0289	0.0260	0.0402	0.0423	0.0534	0.0412	0.0341	0.0267	0.1248	0.0588	0.0476

Table 4. Particle Number Testing

Total Partikel	Best Value Function of-										Average Function Value
	1	2	3	4	5	6	7	8	9	10	
3	0.0383	0.0568	0.1614	0.0496	0.0791	0.0411	0.1309	0.0756	0.1285	0.0425	0.0804
6	0.0267	0.0413	0.0552	0.0320	0.0234	0.0402	0.0320	0.0212	0.0186	0.0219	0.0312
9	0.0226	0.0252	0.0310	0.0287	0.0262	0.0448	0.0190	0.0380	0.0190	0.0277	0.0282
12	0.0233	0.0243	0.0292	0.0258	0.0212	0.0238	0.0234	0.0230	0.0303	0.0236	0.0248
15	0.0237	0.0145	0.0225	0.0242	0.0203	0.0310	0.0184	0.0195	0.0247	0.0296	0.0228

2.7.3 Iteration Number Testing

Testing the number of iterations is done to find out the number of iterations that need to be used to get the optimal (best) function and weight values. The function value here is the error value or error value, where the best error value is the smallest value. The number of iterations used in this test is 10, 20, 30, 40, and 50. The results of testing the number of iterations that have been done can be seen in Table 5.

Table 5. Iteration Number Testing

Total Iteration	Best Value Function of-										Average Function Value
	1	2	3	4	5	6	7	8	9	10	
10	0.0202	0.0302	0.0169	0.0188	0.0196	0.01602	0.01545	0.02813	0.03466	0.01872	0.02187
20	0.0107	0.0100	0.0170	0.0144	0.0129	0.01256	0.01064	0.01197	0.01202	0.00947	0.01216
30	0.0097	0.0098	0.0106	0.0094	0.0121	0.00984	0.00948	0.00993	0.00912	0.01035	0.01004
40	0.0097	0.0086	0.0084	0.0086	0.0088	0.00916	0.00886	0.00884	0.00957	0.01027	0.00907
50	0.0093	0.0085	0.0089	0.0093	0.0095	0.00906	0.00832	0.00970	0.01004	0.00850	0.00911

2.7.4 Position Limit Testing

Position limit testing is done to find out the position limit that needs to be used to get the optimal (best) function and weight value. The function value here is the error value or error value, where the best error value is the smallest value. The position limits used in this test are (-5, 5), (-10, 10), (-15.15), (-20, 20), and (-25, 25). The results of the position limit test that have been made can be seen in Table 6.

Table 6. Position Limit Testing

P.Min	P.max	Best Value Function of-										Average Function Value
		1	2	3	4	5	6	7	8	9	10	
-5	5	0.0108	0.0100	0.0103	0.0094	0.0097	0.0090	0.0093	0.0101	0.0094	0.0097	0.0098
-10	10	0.0088	0.0086	0.0087	0.0093	0.0089	0.0108	0.0097	0.0121	0.0095	0.0095	0.0096
-15	15	0.0091	0.0094	0.0113	0.0085	0.0083	0.0093	0.0129	0.0106	0.0115	0.0094	0.0100
-20	20	0.0099	0.0125	0.0099	0.0105	0.0117	0.0151	0.0114	0.0088	0.0104	0.0116	0.0112
-25	25	0.0175	0.0115	0.0272	0.0105	0.0157	0.0215	0.0100	0.0150	0.0097	0.0127	0.0151

2.7.5 Speed Limit Testing

Speed limit testing is done to find out the speed limit that needs to be used to get the optimal (best) function and weight value. The function value here is the error value or error value, where the best error value is the smallest value. The speed limits used in this test are (-1, 1), (-2, 2), (-3, 3), (-4, 4), and (-5, 5). The results of the speed limit test that have been carried out can be seen in Table 7.

Table 7. Speed Limit Testing

K.Min	K.max	Best Value Function of-										Average Function Value
		1	2	3	4	5	6	7	8	9	10	
-1	1	0.0086	0.0099	0.0107	0.0089	0.0087	0.0087	0.0092	0.0087	0.0089	0.0091	0.0091
-2	2	0.0096	0.0090	0.0085	0.0098	0.0092	0.0094	0.0089	0.0091	0.0120	0.0097	0.0095
-3	3	0.0091	0.0088	0.0095	0.0100	0.0097	0.0093	0.0093	0.0093	0.0112	0.0098	0.0096
-4	4	0.0097	0.0096	0.0169	0.0102	0.0152	0.0107	0.0145	0.0099	0.0102	0.0089	0.0116
-5	5	0.0132	0.0134	0.0138	0.0096	0.0108	0.0122	0.0091	0.0115	0.0097	0.0108	0.0114

2.7.6 Conclusion Testing Methods

From the overall tests conducted, the following are the best parameters obtained to produce optimal function values and weights. Here are the best parameters obtained during the testing process :

a. Particle position limit : (-10, 10)
b. Particle speed limit : (-1, 1)
c. Number of swarms : 3
d. Number of particles : 15
e. Number of iterations : 40
f. Weight of inertia (*w*) : 0.729
g. Cognitive weight const (*c1*) : 1.49445
h. Social weight const (*c2*) : 1.49445
i. Global weight const (*c3*) : 0.3645

Initialization of the position and speed of particles carried out randomly will produce different optimization results even though using the same parameters.

Therefore, by using the best parameters obtained from the test, then optimization will be carried out 10 times by using the best parameters obtained. The results of the optimization can be seen in Table 8.

Table 8. Optimization Result

	Best Value Function of-									
	1	2	3	4	5	6	7	8	9	10
Function Value	0.0101	0.0083	0.0092	0.0095	0.0092	0.0092	0.0132	0.0105	0.0093	0.0087
Accuracy	96.01%	96.01%	96.01%	97.01%	95.76%	96.26%	95.76%	97.26%	97.51%	96.51%

From the optimization results obtained through 10 tests conducted, the lowest accuracy value is 95.76% and the highest accuracy value is 97.51%. Following are the best weights obtained at the lowest accuracy values can be seen in Tables 9 and 10.

Table 9. Best Weight of Lowest Optimization Results

b0	KR	DR	AR	GPMR	XPMR	LHR	DNR	GR
2.8343	1.3366	-5.7329	-0.8344	8.4690	3.2384	-4.8872	-0.2276	1.0626

Table 10. Best Weight of Lowest Optimization Results - Continued

KD	DD	AD	GPMD	XPMD	LHD	DND	GD
2.8031	2.0400	2.0813	-9.1808	-3.3933	3.8595	0.8637	-1.3780

By using the best weights, the lowest optimization results in Tables 9 and 10, we get the results of the accuracy of the predicted victory of the match with the results of the comparison as follows in Table 11.

Table 11. Lowest Comparison Results

Total Data	True Comparison Total	False Comparison Total	Accuracy
401	384	17	95.76%

While the best weight obtained at the highest accuracy value can be seen in Tables 12 and 13.

Table 12. Best Weight of Highest Optimization Results

b0	KR	DR	AR	GPMR	XPMR	LHR	DNR	GR
0.5840	1.1124	-2.6377	0.2268	4.0970	1.2219	-3.2434	-0.1159	4.8480

Table 13. Best Weight of Highest Optimization Results - Continued

KD	DD	AD	GPMD	XPMD	LHD	DND	GD
2.8031	2.0400	2.0813	-9.1808	-3.3933	3.8595	0.8637	-1.3780

By using the best weighting, the highest optimization results in Tables 12 and 13, the results obtained are accurate predictions of victory matches with the results of the comparison as follows in Table 14.

Table 14. Highest Comparison Results

Total Data	True Comparison Total	False Comparison Total	Accuracy
401	391	10	97.51%

From the comparison of the results on the test data with the prediction results, by using the best weight the lowest optimization results, the same comparison results were obtained as many as 384 data with the results of different comparisons of 17 data and an accuracy rate of 95.76%. Whereas by using the best weight of the highest optimization results, the same comparison results were obtained with 391 data with different comparison results of 10 data and an accuracy rate of 97.51%. These results indicate that the method used can produce optimal solutions.

3. CLOSING

Based on the results of research that has been done, it can be concluded that the application of the Logistic Regression (LR) method with the optimization of Multi-Swarm Optimization (MSO) in predicting victory in the DOTA 2 game in this system can produce excellent weighting through the optimization process, the weight is then used in the process of predicting matches using the Logistic Regression method and produces the lowest accuracy rate of 95.76% with the highest accuracy of 97.51%.

This research can be developed by increasing the number of features that affect the prediction process, one of which is an item that greatly influences the running condition of the game during the game. Adding some features such as prediction feature based on character selection in the game, or based on the players who play the DOTA 2 game can also be done since data access in various kinds of matches is easily available.

BIBLIOGRAPHY

[1] N. Kinkade and K. Lim, “DOTA 2 Win Prediction”, Technical Report, University of California, San Diego, 2015.

[2] Z. Li, D. Cui, and C. Li, “Dota2 Outcome Prediction”, Technical Report, University of California, San Diego, 2017.

[3] V. Robles, C. Bielza, P. Larranaga, S. Gonzalez, and L. Ohno-Machado, “Optimizing logistic regression coefficients for discrimination and calibration using estimation of distribution algorithms”, *TOP - An Official Journal of the Spanish Society of Statistics and Operations Research*, vol. 16, pp. 345, 2008.

[4] M. Subathra and R. Nedunchezhian, “An Improved Alias Classification using Logistic Regression with Particle Swarm Optimization”, *Indian J. Sci. Technol.*, vol. 8, no. 28, pp. 1–5, 2016.

[5] J. McCaffrey, “Test Run - Logistic Regression Classification with Multi-Swarm Optimization”, *Microsoft Developer Network*, Jan-2015.

[6] Valve Corporation, *The International - DOTA 2 CHAMPIONSHIP*, http://www.dota2.com/international/overview/, 15 September 2018 14.57

[7] A. Cui, H. Chung, and N. Hanson-Holtry, *Open Source DOTA 2 Data Platform*, https://www.opendota.com/, 15 September 2018 15.20

[8] C. C. Aggarwal, *Data Mining - The Textbook*. New York: Springer, 2015.

[9] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2 ed. San Francisco: Morgan Kauffman, 2006.
[10] R. A. Gordon, Applied Statistics for the Social and Health Sciences. New York: Routledge, 2012.
[11] J. McCaffrey, ‘Test Run - Multi-Swarm Optimization’, *Microsoft Developer Network*, Feb-2013.
[12] G. Hermawan, “Implementasi Algoritma *Particle Swarm Optimization* Untuk Penentuan Posisi Strategis *Agent* Pada Simulasi Robot Sepak Bola Dua Dimensi”, Jurnal Ilmiah Komputer dan Informatika (KOMPUTA), pp. 63-70, 2012.
[13] D. G. Kleinbaum dan M. Klein, *Logistic Regression - A Self-Learning Text*, 3 ed. New York: Springer, 2010.