Optimization of Decision Tree Algorithm in Text Classification of Job Applicants Using Particle Swarm Optimization

Wikan Kuncara Jati[1], Kemas Muslim L.[2]
[1,2]School of Computing, Telkom University, Bandung, Indonesia
[1]wikankun@student.telkomuniversity.ac.id, [2]kemasmuslim@telkomuniversity.ac.id

2020 3rd International Conference on Information and Communications Technology (ICOIACT) | 978-1-7281-7356-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICOIACT50329.2020.9332101

***Abstract*—Job interview is one of the stages that must be passed by job applicants before getting a job. However, the manual interview process resulted in a large amount of cost and time of selection, so we need a system that can provide recommendations of which applicants are qualified. Currently there are many studies of text classification using the Naive Bayes method, K-Nearest Neighbor, Support Vector Machine, and Deep Learning. Therefore, algorithms that are used less frequently will be tested in this study, such as the Decision Tree. Then the Swarm Intelligence method, Particle Swarm Optimization, is implemented to improve the performance of the method. So that this research focuses on testing and comparing ordinary Decision Tree and Decision Tree methods that are optimized with Particle Swarm Optimization. From the test results, the accuracy of the optimized model increased by 7.1%, and the highest accuracy achieved was 74.3%.**

***Keywords*—job interview, job applicants, text classification, decision tree, particle swarm optimization.**

I. Introduction

One stage in the employee selection process is a job interview that must be passed by the job applicant before getting a job. The job interview is a very important stage because in this stage the interviewer can assess the personality of the job applicant. The purpose of the job interview is to assess the psychological aspects, behavior, leadership, commitment, and various other aspects that are included in the company's assessment. So job applicants who pass the interview stage are expected to have values that are in line with the company's work culture.

PT. Telkom is one of the largest telecommunications companies in Indonesia which routinely holds employee recruitment every year. Every time PT. Telkom recruits employees, job applicants who register reach tens of thousands of people. To get the best employees who have values in accordance with the work culture of PT. Telkom, it must be selected based on the suitability of the applicant's character with the work culture of PT. Telkom. Despite going through various processes such as administrative selection to psychological testing, the number of applicants who passed to the interview stage was still too much. So, to be able to make a selection of many people is needed a lot of experts as well. Resulting in very large costs and very long selection time duration.

To overcome the large amount of costs that must be incurred to hire experts and shorten the selection time, we need a system that can provide recommendations for job applicant qualifications. By analyzing the essay inputted by the applicant, the system can then classify the suitability of the applicant's character with the company's work culture. From the results of the classification, the system can provide a recommendation whether the applicant qualifies or not.

Over the last few years, text classification has been widely studied and rapid progress has been seen in this field, including Machine Learning approaches such as Bayesian classifier, Decision Tree and Random Forest, k-Nearest Neighbor (KNN), Support Vector Machine (SVM), Neural Networks Imitation, and Rocchio [1]. The algorithms are very popular because they produce very good accuracy. So this research will focus on other algorithms that are used less frequently.

Tree-based classification algorithms, such as the Decision Tree, are algorithms that have advantages in terms of simplicity, interpretability, and the ability to handle feature interactions [2]. However, this algorithm is very sensitive to small disturbances in data, and can easily overfit [3], and has problems with out-of-sample prediction [4]. According to Suyanto [5], to overcome the weaknesses of the Decision Tree algorithm, the Swarm Intelligence (SI) technique can be used to optimize the algorithm. Therefore, this research focuses on testing and comparing Decision Tree algorithms without Swarm Intelligence and Decision Tree algorithms that are optimized with Swarm Intelligence. The Swarm Intelligence algorithm used in this study is Particle Swarm Optimization because of its simple concept and easy implementation [6].

II. Related Works

I. Naim et.al. [7], proposed a research in prediction and analysis of job interview performance using interview videos from Massachusetts Institute of Technology (MIT). Their automated analysis includes facial expressions (smiles, head gestures), language (word counts, topic modeling), and prosodic information (pitch, intonation, pauses) of the interviewees. Their framework predicts the ratings for interview traits automatically. The traits predicted such as excitement, friendliness, and engagement with correlation coefficients of 0.73 or higher, and quantifies the relative importance of facial expressions, language, and prosody.

L. Chen et.al. [8], proposed an automated assessment of interview videos using Doc2Vec multimodal feature extraction

Authorized licensed use limited to: Prince of Songkla University provided by UniNet. Downloaded on July 15,2021 at 09:45:18 UTC from IEEE Xplore. Restrictions apply.

paradigm. A novel feature extraction method using ”visual words” automatically learn the video analysis output and proposes the Doc2Vec paradigm. The results show that this method provides an effective representation for the automated assessment of video interviews.

III. Literature Study

A. Pre-processing

Text documents prepared for text classification are represented by a large number of features. Data pre-processing is carried out at this stage to present the text document in clear word format. According to Korde [1], common data pre-processing processes are tokenization and stemming. Tokenization is the process of dividing text that can be in the form of sentences, paragraphs or documents, into certain tokens / parts. The references to tokenisation are often punctuation and spaces. Then stemming is by applying an algorithm that converts words into its canonical form. This step is the process of uniting tokens into their root form, for example ”eating” into ”eat”. So the Machine Learning algorithm can recognize the meaning of the word even though it has different forms. The next step is to delete stop words like ”the” and ”and”, so that there are no words that are not important. In addition, the elimination of stop words also affects the speed and accuracy of the Machine Learning algorithm.

B. Feature Extraction

After data is cleaned, a feature extraction method can be applied. The purpose of feature extraction is to change the format of documents into formats supported by the Machine Learning algorithm, namely by extracting semi-structured data forms (text) into structured (text features) [9]. The feature extraction technique used is Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF evaluates how relevant a word is to a document in an existing document. TF-IDF is calculated using formula (1).

$$tf_{t,d} \cdot \log\left(\frac{N}{df_t}\right) \tag{1}$$

Where:

$tf_{t,d}$	:	the number of times that term t occurs in document d
N	:	total number of documents in the corpus
df_t	:	number of documents where the term t appears

C. Decision Tree

Tree-based classification algorithms, such as Decision Tree, are algorithms that have advantages in terms of simplicity, interpretability, and the ability to handle feature interactions. The main idea of a Decision Tree is to create a tree based on attributes for categorized data points. Each internal node in the tree represents a test for an attribute (for example, if the value of a variable is more than 5), test results are represented by branches, class labels are represented by leaf nodes, and classification rules are represented by paths from root to leaf nodes. The Decision Tree algorithm used in this study is CART (Classification and Regression Trees). CART is used because according to S. Singh [10], CART can handle outliers well, which ID3 and C4.5 are susceptible to outliers.

D. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a well-known Swarm Intelligence algorithm. Its inspiration comes from the social behavior of flocks of birds flocking (bird flocking) or fish swimming in droves (fish schooling) that form certain formations without colliding with each other even though the distance between the animals are close together. This method searches for the optimal solution through agents, which are represented as particles, whose trajectories are adjusted by stochastic and deterministic components. Each particle is influenced by its best position and the best position in the group, but tends to move randomly. Following is the Particle Swarm Optimization algorithm in general.

```
particle initialization;
repeat
    for every particle do
        calculate fitness value;
        if new fitness is better than old fitness then
            renew fitness
        end
    end
    select particle with best fitness;
    save the index of particle with best fitness;
    for every particle do
        calculate the velocity using (2);
        renew the position using (3);
    end
until stop condition = True;
```

Algorithm 1: Particle Swarm Optimization

A particle *i* is represented in the position vector x_i and the velocity vector v_i. In the initialization process, the position vector is initialized with random numbers in the search scope. While the velocity vector is initialized with zero vector. At each iteration, the particle velocity and position change with equations 2 and 3 as follows [11].

$$v_{id} = \omega \cdot v_{id} + \phi_1 \cdot r \cdot (p_{id} - x_{id}) + \phi_2 \cdot r \cdot (p_{gd} - x_{gd}) \tag{2}$$

$$x_{id} = x_{id} + v_{id} \tag{3}$$

Where:

i	:	i-th particle
d	:	d-th dimension
ω	:	velocity reduction every generation (inertia)
ϕ_1	:	individual learning rate
ϕ_2	:	social learning rate
p	:	the best fitness value vector produced so far
g	:	the particle index with the best fitness in neighborhood topology
r	:	random floating point in the interval [0,1]

Authorized licensed use limited to: Prince of Songkla University provided by UniNet. Downloaded on July 15,2021 at 09:45:18 UTC from IEEE Xplore. Restrictions apply.

IV. Proposed Method

A. General Scheme

The general scheme proposed for this research is described as in figure 1.

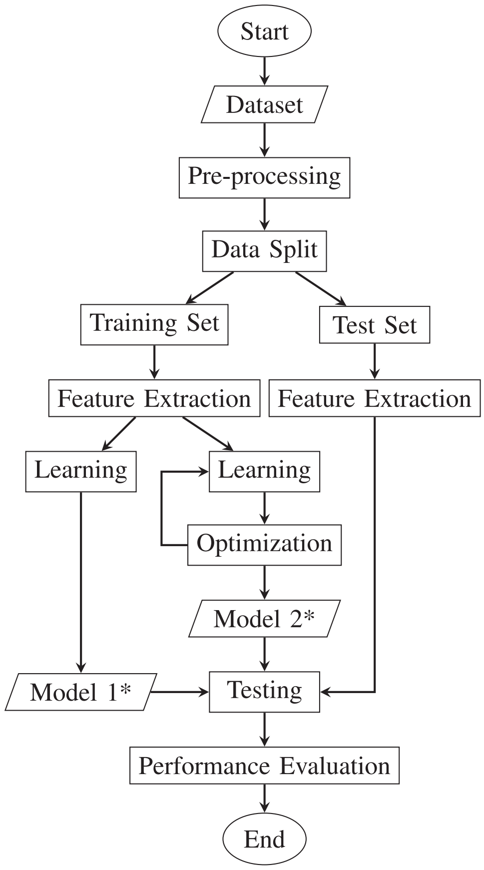

Fig. 1. General scheme

*Model 1 is a model that is trained with the ordinary Decision Tree algorithm, while model 2 is a model that is trained with the Decision Tree algorithm that is optimized with Particle Swarm Optimization.

B. Dataset

The original dataset used are interview transcript selection of PT. Telkom which has been given a weight according to 9 work culture values of PT. Telkom by professional psychologists. The nine values are: action, enthusiasm, focus, imagine, integrity, smart, solid, speed, and totality. There are 56 interview transcripts for each work culture value. Each record is categorized into three levels: 1 (low), 2 (medium), and 3 (high). The format of the original dataset is described as in table I

After the data is cleaned, the label is reduced. Data that has a score of 1 and 2 is merged and replaced with a score of 0, and the data with score of 3 replaced with a score of 1. So the data only has 2 labels that are 0 (low) and 1 (high). These data which format shown as in table II is ready to be transformed to a TF-IDF vector before being fed to the learning process. Finally, the dataset is in the form of the sparse matrix.

TABLE I
Original dataset format

name	interview	score
Aflah Naufal	biasanya Ketika saya mendapatkan ...	3
Anbi Alif Rahman	pertama saya akan membandingkan ...	2
Budi Pekerti	Oh biasanya kalau untuk menyusun ...	3
Ferdian Yulianto	menghadapi dua agenda penting ...	1

TABLE II
Dataset format after cleaned

name	interview	score
aflah naufal	biasa saya dapat acara sama sama penting ...	1
anbi alif rahman	pertama akan banding dua acara ...	0
budi pekerti	biasa kalau susun program mana kerja ...	1
ferdian yulianto	hadap agenda penting waktu sama ...	0

C. Pre-processing

Pre-processing stage is one of the major stage in Classification. The following is the proposed steps at this stage.

1) Read the dataset and remove empty records.
2) Tokenize the dataset,
3) Perform lowercase conversion,
4) Remove stop words from the dataset,
5) Stem the dataset so that it changes into canonical form.

D. Decision Tree Optimization

Decision Tree optimization is done by finding the best hyper-parameter value for each classification case with Particle Swarm Optimization. The parameters are coded into an array of integers representing parameters in the Decision Tree algorithm. The coded parameters is used in Particle Swarm Optimization as position vector. Both of position vector and velocity vector ranged from 0 to 99. Parameter encoding is explained in table III.

TABLE III
Optimized Decision Tree Parameters

No	Parameters	Value
1	Criterion	'gini', 'entropy'
2	Splitter	'best', 'random'
3	Maximum Depth	integer
4	Minimum Sample Split	integer
5	Minimum Sample Leaf	integer
6	Maximum Features	None, 'sqrt', 'log2'

Then the Particle Swarm Optimization algorithm is run in conjunction with the learning process to find the best value of these parameters. The fitness function used is the average value of accuracy and F1-score. This aims to minimize overfit when the data is too balance or too imbalance. After the optimization process is complete, the learning process is also finished and produces the best parameters that have been found. For comparison, the default (not optimized) Decision Tree have default parameters as shown in the table IV.

Authorized licensed use limited to: Prince of Songkla University provided by UniNet. Downloaded on July 15,2021 at 09:45:18 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
DEFAULT DECISION TREE PARAMETERS

No	Parameters	Value
1	Criterion	'gini'
2	Splitter	'best'
3	Maximum Depth	None (pure leaf)
4	Minimum Sample Split	2
5	Minimum Sample Leaf	1
6	Maximum Features	None (n features)

V. EXPERIMENTAL RESULT AND ANALYSIS

System testing is done by comparing the accuracy and performance of the Decision Tree algorithm that is not optimized and the Decision Tree algorithm that is optimized with Particle Swarm Optimization.

The test is carried out on three different scenarios by changing the proportion variable of the data train : data test used is as follows:

1) 80% data train : 20% data test
2) 70% data train : 30% data test
3) 60% data train : 40% data test

In addition, the hyper-parameters used in Particle Swarm Optimization are as follows:

- Number of agents = 20
- Stop condition = 15 generations
- ω = 0.5
- ϕ_1 = 0.5
- ϕ_2 = 0.5

The accuracy and F1-score of classification is as follows in table V. In addition, table VI, VII, and VIII is the decoded parameters of Decision Tree as result of optimization done by Particle Swarm Optimization.

TABLE V
EXPERIMENT RESULT

Scenario	Decision Tree		Decision Tree + PSO	
	Accuracy	F1-score	Accuracy	F1-score
1	66.4	67.0	71.4	71.7
2	62.8	64.3	68.1	68.1
3	63.3	64.3	74.3	71.7

TABLE VI
OPTIMIZED DECISION TREE PARAMETERS ON SCENARIO 1

Work Culture	Parameter Number					
	1	2	3	4	5	6
Integrity	gini	random	37	2	1	sqrt
Enthusiasm	entropy	best	58	6	1	log2
Totality	gini	best	17	7	1	sqrt
Solid	entropy	best	76	7	1	sqrt
Speed	gini	best	28	6	1	log2
Smart	gini	best	23	3	2	None
Imagine	entropy	random	26	3	2	None
Focus	entropy	best	97	2	2	sqrt
Action	gini	best	84	5	2	None

TABLE VII
OPTIMIZED DECISION TREE PARAMETERS ON SCENARIO 2

Work Culture	Parameter Number					
	1	2	3	4	5	6
Integrity	gini	random	33	4	1	None
Enthusiasm	entropy	random	63	3	2	None
Totality	gini	best	75	2	2	None
Solid	gini	random	85	6	1	None
Speed	gini	best	33	4	1	None
Smart	gini	best	16	3	1	log2
Imagine	entropy	random	81	2	1	log2
Focus	entropy	random	43	3	1	None
Action	gini	best	63	3	1	log2

TABLE VIII
OPTIMIZED DECISION TREE PARAMETERS ON SCENARIO 3

Work Culture	Parameter Number					
	1	2	3	4	5	6
Integrity	entropy	best	42	3	1	log2
Enthusiasm	gini	best	33	5	1	None
Totality	gini	best	59	6	1	None
Solid	gini	best	35	6	1	None
Speed	entropy	random	46	5	1	None
Smart	entropy	best	38	2	1	sqrt
Imagine	gini	random	23	6	1	None
Focus	gini	random	23	2	1	log2
Action	entropy	random	55	2	1	sqrt

There also time cost metric used in this experiment. Time cost is the length of time required by the Particle Swarm Optimization to optimize the Decision Tree in a dataset in each work culture. The average time cost for optimization in each scenario is as in table IX.

TABLE IX
TIME COST RESULT

Work Culture	Time Cost (in seconds)		
	Scenario 1	Scenario 2	Scenario 3
Integrity	0.647887	0.612733	0.616355
Enthusiasm	0.646270	0.627782	0.634817
Totality	0.614832	0.591423	0.588431
Solid	0.638300	0.642780	0.634285
Speed	0.627825	0.648804	0.627827
Smart	0.644266	0.624822	0.621317
Imagine	0.629326	0.618339	0.634802
Focus	0.660711	0.653228	0.635276
Action	0.640783	0.632806	0.629856
Average time cost	**0.638911**	**0.628080**	**0.624774**
Total time cost	**5.750199**	**5.652717**	**5.622966**

VI. CONCLUSION

Based on the results of testing on the three scenarios performed, it can be seen that both accuracy and F1-scores on models that use a Decision Tree that is optimized with Particle Swarm Optimization are better than an ordinary Decision Tree. The average accuracy increased by 7.1%, and the average F1-score rose by 5.3%. In general, by finding the best value of hyper-parameters, the accuracy and performance of the Decision Tree algorithm is better. This is important because hyper-parameters control the overall behavior of the Decision Tree model.

Authorized licensed use limited to: Prince of Songkla University provided by UniNet. Downloaded on July 15,2021 at 09:45:18 UTC from IEEE Xplore. Restrictions apply.

The time cost result table shows that the total time required to optimize the process of learning from the dataset using Decision Tree is about 5.7 seconds. With a relatively small number of datasets, the time cost is also relatively short.

Based on the results of tests conducted, we obtain some conclusion. The average accuracy and F1-score of a Decision Tree model optimized with Particle Swarm Optimization is better than an ordinary Decision Tree. The highest value of accuracy and F1-score achieved was 74.3% and 71.7%. In other words, hyper-parameters play important roles in determining the performance of an algorithm.

REFERENCES

[1] V. Korde and C. N. Mahender, "Text classification and classifier: A survey," International Journal of Artificial Intelligence & Applications, vol. 3, no. 2, pp. 85–99, 2012.

[2] I. Hmeidi, M. Al-Ayyoub, N. A. Abdulla, A. A. Almodawar, R. Abooraig, and N. A. Mahyoub, "Automatic arabic text categorisation: A comprehensive comparative study," Journal of Information Science, vol. 41, no. 1, p. 114, 2015.

[3] C. Giovanelli, X. Liu, S. Sierla, V. Vyatkin, and R. Ichise, "Towards an aggregator that exploits big data to bid on frequency containment reserve market," Annual Conference of the IEEE Industrial Electronics Society (IECON 2017), vol. 43, pp. 7514–7519, 2017.

[4] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," Information, vol. 10, no. 4, p. 150, 2019.

[5] Suyanto, Swarm intelligence: Komputasi modern untuk optimasi dan big data mining. Penerbit Informatika, 2017.

[6] K. Y. Lee and J. Park, "Application of particle swarm optimization to economic dispatch problem: Advantages and disadvantages," in 2006 IEEE PES Power Systems Conference and Exposition, pp. 188–192, 2006.

[7] I. Naim, M. I. Tanveer, D. Gildea and M. E. Hoque, "Automated Analysis and Prediction of Job Interview Performance," in IEEE Transactions on Affective Computing, vol. 9, no. 2, pp. 191-204, 1 April-June 2018.

[8] L. Chen, G. Feng, C. W. Leong, B. Lehman, M. Martin-Raugh, H. Kell, and S. Y. Yoon, "Automated scoring of interview videos using Doc2Vec multimodal feature extraction paradigm,". Proceedings of the 18th ACM International Conference on Multimodal Interaction, pp. 161-168, 2016.

[9] Z. Wang, X. Cui, L. Gao, and et al., "A hybrid model of sentimental entity recognition on mobile social media," Eurasip Journal on Wireless Communications and Networking, 2016.

[10] S. Singh and M. Giri, "Comparative Study Id3, Cart And C4.5 Decision Tree Algorithm: A Survey," International Journal of Advanced Information Science and Technology (IJAIST), vol. 3, no. 7, pp. 47-52, 2014.

[11] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39–43, 1995.

Authorized licensed use limited to: Prince of Songkla University provided by UniNet. Downloaded on July 15,2021 at 09:45:18 UTC from IEEE Xplore. Restrictions apply.