SEMARAK ILMU PUBLISHING
202103268166(003316878-P)

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Comparison of Pre-Defined Automatic Machine Learning (AutoML) for MBTI Personality Prediction of Twitter Users using Binary Classification Approach

Selvi Fitria Khoerunnisa[1], Farikhin[2], Bayu Surarso[2], Ahmad Ainun Herlambang[3], Retno Kusumaningrum[4,*]

[1] School of Postgraduate Studies, Universitas Diponegoro, Semarang Selatan, Semarang City, Central Java 50241, Indonesia
[2] Department of Mathematics, Faculty of Science and Mathematics, Universitas Diponegoro, Semarang Selatan, Semarang City, Central Java 50241, Indonesia
[3] Faculty of Engineering and Information Technology, University of Melbourne, Parkville VIC 3052, Australia
[4] Department of Informatics, Faculty of Science and Mathematics, Universitas Diponegoro, Semarang Selatan, Semarang City, Central Java 50241, Indonesia

ABSTRACT

The Myers-Briggs Type Indicator (MBTI) is a personality test that is globally accepted and used as a method for identifying personality. MBTI uses a four-factor linear model to characterize a person's behaviour patterns. This feature is often used to pursue career opportunities, make decisions, manage leadership, and deal with stress. In particular, MBTI personality prediction has been widely conducted and well performed using Recurrent Neural Network (RNN) based on Twitter data because it indirectly reveals most of a person's personality through their tweets. However, deep understanding is needed in building RNN-based solutions. Hence it will take a lot of time and resources to produce an excellent model architecture and the parameters used. Therefore, this study proposed the Auto Machine Learning (AutoML) method with a pre-defined search space to determine the correct model architecture and hyperparameters based on the results of data analysis. Thus, the search algorithm can exploit environments with suitable configurations in general. There are two pre-defined search spaces employed in this study, i.e. (i) two RNN algorithms, including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), and (ii) pre-trained Word2Ves as word embedding. In addition, this study compares the model's performance that employs preprocessing and raw data (without preprocessing). The first result shows that the preprocessing increases the F1-Score values for LSTM and GRU by 2.35% and 2.02%, respectively. Subsequently, the LSTM outperformed GRU by the values of F1-Score at 0.35% and accuracy at 0.76%. The implementation of LSTM with pre-processed data in pre-defined AutoML with Word2Vec as a word embedding technique can provide good performance on long and complex data sequences such as Twitter data for predicting its user personality.

Keywords:

Automatic machine learning; Personality prediction; MBTI; Binary classification

* *Corresponding author.*
E-mail address: retno@live.undip.ac.id

https://doi.org/10.37934/araset.62.1.106118

1. Introduction

Personality is the overall pattern of individual behaviour, emotional and cognitive which is a prominent feature in interacting with other individuals. Personality also refers to adaptive qualities and behaviours, including traits, interests, abilities, and emotional patterns [1]. Knowing personality can help pursue better career opportunities [2] and improve weaknesses that exist in an individual.

In addition, at work, a person can arrange his work most appropriately, including managing time, solving problems, approaching decision-making, and dealing with stress [3]. One of the most widely used instruments for understanding and predicting a person's personality is the Myers Brings Type Indicator (MBTI). MBTI is also widely discussed on social media because users usually share the results of the MBTI personal test on their social media. MBTI uses a four-factor linear model of four pairs of psychological preferences to predict personality [4]. The pair is E (Extrovert) or I (Introvert), N (Intuitive) or S (Sensing), F (Feeling) or T (Thinking), and J (Judging) or P (Perception).

The evolution in the technological sphere towards the participation of large numbers of users is fuelling the study of personality in the computing environment. In particular, Twitter has become a popular social networking medium for automatically predicting MBTI personality. Twitter is considered a medium for conveying ideas and interactions between individuals. It indirectly reflects their attitudes, behaviour, and personality [5]. Research about predicting MBTI personality has become a growing topic in natural language processing. Several studies have been conducted regarding the automatic prediction of personality. Research conducted by Marouf *et al.*, [6], Nisha *et al.*, [7], and Zumma *et al.*, [8] focuses on syntax and text lexicon and then uses shallow learning as a classifier. The use of deep learning, such as in research by Kumar *et al.*, [9], Majima and Markov [10], and Naik *et al.*, [11] has been able to overcome the problem of feature extraction in shallow learning, where deep learning can extract features automatically [12,13]. However, the process of building deep learning solutions requires depth theoretical knowledge of the appropriate model architecture, as well as its implementation in a computing environment. Experiments are often carried out using the trial-error method to determine appropriate model architecture using hyperparameters as an optimization, which causes long computing times with extensive resources. Those problems can be overcome using the Auto Machine Learning (AutoML) method.

AutoML automates model selection and hyperparameter tuning by encapsulating the process from raw data set to a trained model [14]. At first, AutoML will analyse the input data to build a search space and common hyperparameter values. Then, the search algorithm will determine the hyperparameters values that have high performance. One of the AutoML libraries for deep learning is AutoKeras [15]. Compared to other AutoML libraries such as Auto-PyTorch [16] for structured data tasks or AutoGluon [17], which includes a tree-based model, AutoKeras search space specifically optimized for raw data types and focused on deep learning, also fully compatible with the Tensorflow and Keras ecosystems. The search algorithm on AutoKeras requires initial configuration to identify the type of task to be completed. Configuration on model selection and hyperparameters settings in AutoKeras generally uses black-box optimization. However, instead of using black-box, searches can be implemented with pre-defined search algorithms from search space configured with parameters that are known to perform well in general. The goal is that the search algorithm will continue to exploit the better configuration environment. Based on the description above, this study will predict personality using the pre-defined AutoML model for Twitter users with a binary classification approach.

2. Literature Review

2.1 MBTI Personality

Personality is the totality of thoughts, feelings, and behaviours, consciously and unconsciously. Knowing personality can help pursue better career opportunities and seek to identify and improve weaknesses. One of the most widely used personality tests today is the Myers-Briggs Type Indicator (MBTI) [18]. The MBTI is known for its accuracy and ease of use. MBTI describes personality based on what people have in mind.

As previously explained, the MBTI uses a four-factor linear model to characterize a person's behaviour patterns, namely E/I, N/S, F/T, and J/P. The first pair of psychological preferences, E/I, relates to sources of psychic energy. N/S relates to one's way of seeing the world, F/T relates to decision-making, and J/P relates to one's view of the world. The number of MBTI personality types is 16, a combination of 4 aspects [4]. For example, if someone's personality type is INFP, then it takes descriptive from "I," "N," "T," and "P." As explained before, the MBTI personality is obtained by combining four aspects. In computational science, this can be analysed using a binary classification approach by identifying each pair of aspects in each data. MBTI's pair of psychological preferences then underlies solving the problems in this study using the binary classification approach.

2.2 Related Work

Research on Personality prediction based on MBTI personality types through computational techniques can assist many sectors in understanding social media users. These studies will be briefly reviewed here. Zumma *et al*., [8] researched predicting the personality of Twitter users using the MBTI dataset, which consists of 16 type categories. The 16 categories are then classified into four personality traits from the input text. Namely E/I, N/S, F/T, J/P. This study used six different types of machine learning classifiers in the experiment. The model evaluation shows that Support Vector Machine (SVM) performs best.

The binary classification approach in predicting the personality of Twitter users is also applied to research by Kumar *et al*., [9]. The research data was taken directly using the Twitter API and labelled based on the shared MBTI personality type results. Although the MBTI results are confidential, many individuals openly share them in the media in various ways. The study used the XGBoost algorithm and reaped more than 80% yield. The use of deep learning in predicting the personality of Twitter users in the scope of Indonesian was carried out by Jeremy and Suhartono, [5]. The research predicts Big Five Personalities using RNN methods such as LSTM and GRU and has shown that the best evaluation result in the f1-score is 82.7% for the extroversion class. In contrast, the average for five personality classes in the f1-score is 69.76%. In addition, the resulting model was obtained based on Twitter data from 508 Twitter users and 100 tweets for each user.

3. Research Methodology

This section explains the detailed information about step by step used in this study, as illustrated in Figure 1. The study begins with the preparation of the dataset. The dataset will be collected and labelled manually. Then preprocessing is carried out to reduce noise and prepare the dataset.

The input data for creating the AutoML model consists of two types: data after preprocessing and raw data without initial preprocessing, considering that the AutoKeras library accepts input as raw data. Before the data is trained, the search space will be defined first and used in model selection and hyperparameter tuning in AutoKeras. Then the algorithm will exploit the configuration environment that has been defined. The best model chosen will be evaluated using a confusion matrix to determine performance. The pipeline for the text classification task on AutoKeras is similar to the pipeline in general. However, each step is simplified by automation and the hyperparameters used. If the user does not specify a required variable, AutoML will fill it in automatically. A detailed description of each stage can be seen in the following sub-sections.

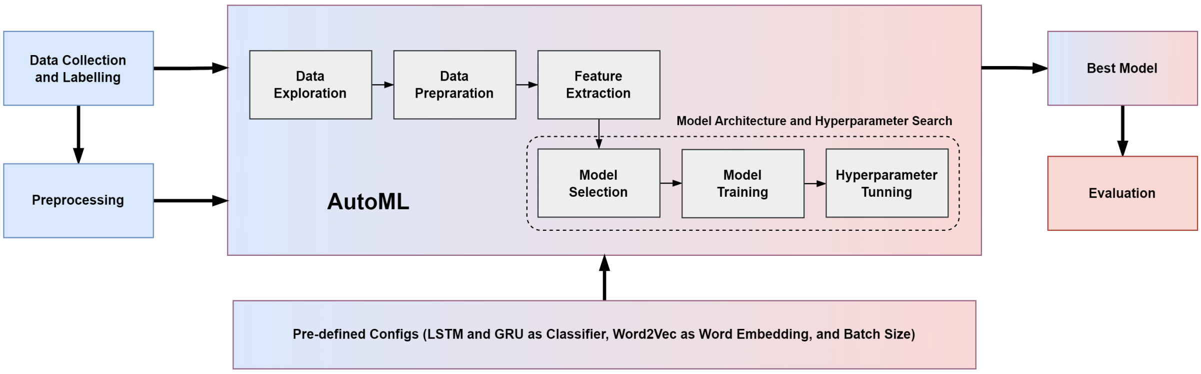

Fig. 1. Research methodology

3.1 Dataset Collection and Labelling

First, the data used in this study amounted to 5120 Indonesian Twitter users crawled using the Python library, namely tweepy from Twitter API. Each user consists of 200 tweets. The crawled dataset is a dataset with one personality type label. The data were manually labelled into 16 MBTI personality types based on the results of personality types shared by Twitter users. Following the binary classification approach, the labels in the dataset that have been collected are divided into four pairs of psychological preferences which consist of Introversion (I) or Extraversion (E), Intuition (N) or Sensing (S), Feeling (F) or Thinking (T), and Perceiving (P) or Judging (J).

3.2 Data Preprocessing

The purpose of preprocessing is to remove noise from the data so that the data is ready to be entered into the model. The following are preprocessing steps in this study as follows in Figure 2.

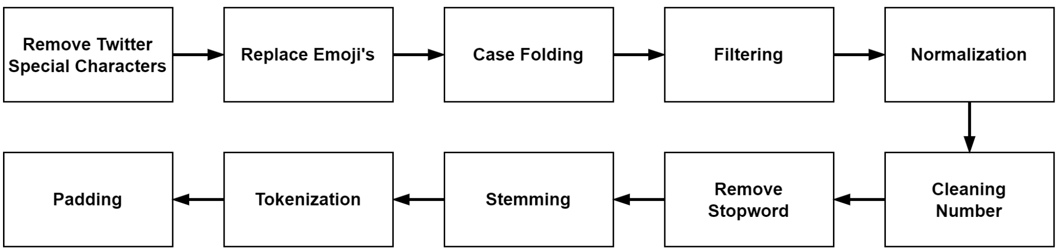

Fig. 2. Preprocessing steps in this study

Based on Figure 2, this study involves several preprocessing steps. The first step is removing Twitter's special characters, including URLs, mentions, and hashtags, in a tweet. Convert expression symbol based on the expression dictionary from Cahyaningtyas *et al.*, [19]. Case folding converts all text letters to lowercase. Then, the dataset will be clean of special characters except for letters, numbers, and hyphens, because that characters will be used in normalization. Normalization is changing slang words into standard forms based on the Colloquial Indonesian Lexicon Dictionary by

Salsabila *et al.*, [20]. After the normalization number is removed, remove stopwords, where stopwords are common words and rarely represent a particular class.

Stemming is hanging affixed words into essential words. Tokenization divides sentence into words based on spaces, tabs, or enter. Lastly, the padding process of equalizing the length is because deep learning requires the same input length. This study will then compare the performance results of the model trained with the dataset that has been pre-processed and the dataset without preprocessing. The goal is to see if AutoML can generate good-performing models even without preprocessing.

3.3 Automatic Machine Learning (AutoML)

Machine learning methods differ in solving different problems. This problem causes many resources needed to determine a suitable architectural model and optimal hyperparameters according to the problem being solved. AutoML can overcome this problem by specifying an auto-learning configuration within a time limit. AutoML enables accelerating machine learning research, improving performance, and implementing efficiently. One of the AutoML methods in the scope of deep learning is AutoKeras. AutoKeras is a library developed based on Tensorflow and Keras. AutoKeras implements based on search space and search algorithms. AutoKeras input data is raw data, then produces output as a trained model.

In general, the search algorithm in AutoKeras was applied as black-box optimization, which performs random searches based on the input data analysis results. However, instead of doing this method, AutoKeras can also implement a specific pre-defined search space in the search algorithm [15]. The goal is to take advantage of prior knowledge, perform a search, and exploit the scope of a suitable configuration. Specific pre-defined in AutoKeras are implemented into building blocks that connect input data to neural network blocks. Figure 3 shows the representation of the AutoKeras building blocks used in this study for the search space configuration.

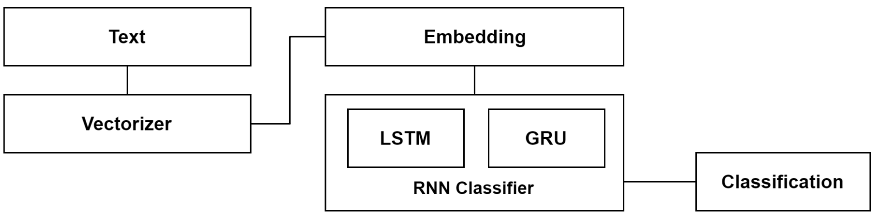

Fig. 3. The architecture of pre-defined AutoML used in this study

Based on Figure 3, the input text will be defined first. Text will be vectorized and weighted in the embedding process using pre-trained Word2Vec. The embedding results will be classified using the RNN block, which consists of LSTM and GRU. AutoKeras, in determining the optimized model, the search is carried out in iterations, and the user determines the number of iterations. Figure 4 shows the AutoML process in AutoKeras. After the data is initialized, fitting is an iterative process in three steps, according to [21].

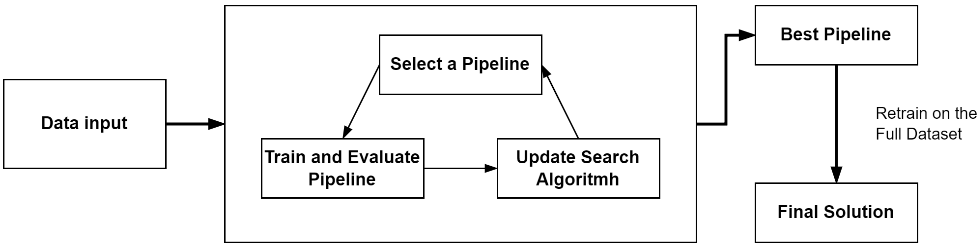

Fig. 4. AutoML process of the AutoKeras

First, AutoML will select a pipeline, classifier, and previously defined word embedding that will be selected from the search space based on the search algorithm from AutoML. The pipeline is then trained and evaluated to obtain its classification accuracy. The search algorithm will be updated based on the performance of the previously explored pipeline to become more efficient. After reaching the maximum number of iterations, the best pipeline will be found and retrained on the full dataset to get the final solution.

The Hill climbing algorithm by Elsken *et al.*, [22] is implemented by AutoKeras in the search algorithm to determine the best pipeline. *Hill climbing* is an algorithm that continues to move continuously and will end if it finds the peak of the solution. As shown in Algorithm 1, the search process in AutoKeras starts with configuring the number of evaluations in the search algorithm (t) and m the number of pre-defined configs in the search space. The pre-defined value of the search space is used to select the best hyperparameters. The parameters will be converted into sub-modules arranged hierarchically. Only one sub-module is selected in each mutation, and then the hyperparameter value is resampled. Submodules can be a single hyperparameter, a layer, or a complete model. Evaluation is carried out on each selected sub-module. The best evaluation will be stored and continue until the total number of evaluations ends.

Algorithm 1 The Search algorithm of AutoKeras
for $i \leftarrow 1$ to t **do**
if $i \leq m$ **then**
eval (ith pre-defined hp)
else
eval(mutate(get_best_hp())

3.4 Pre-Trained Word2Vec

Word embedding is a word mapping technique based on existing dictionaries for those containing real numbers. It is because the neural network cannot accept the input string. The words are converted into vector space [5]. This process will convert strings into vectors that can be input for deep-learning architectural models. In this study, the word embedding method is Word2Vec. Word2Vec can learn the representation of the group of words and then group similar words with the same vector. Because of that Word2Vec can efficiently utilize and train more data. Word2Vec models in this study used pre-trained word2vec developed by Google and were trained using the Google News dataset with a word count of over 11 billion words.

3.5 Long Short-Term Memory (LSTM)

The classification method that will be used to build the personality prediction model is LSTM. LSTM is a deep learning method developed from the RNN architecture. RNN has a long-term dependency problem and missing gradient [23], but LSTM can solve that problem through the presence of memory cells and gate units (input gate i_t, forget gate f_t, and output gate o_t). Figure 5 displays the LSTM network architecture.

LSTM has long-term memory and can process sequence data using a gate vector at each position to control the passage of information along the sequence. The input gate is the gate that determines the input value to be activated in the cell state. Forget gates determine whether some information needs to be removed from the cell. Meanwhile, the output gate is a gate that determines output based on input and memory cells for each timestamp t.

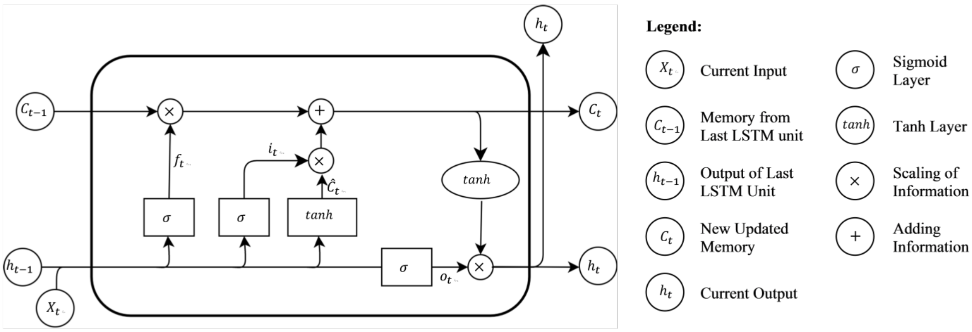

Fig. 5. LSTM architecture

3.6 Gated Recurrent Unit (GRU)

GRU is a modification of the RNN network. The architecture of GRU is different from LSTM. GRU simplifies LSTM but retains the ability to remember extended timestamps [24]. Several parameters in the LSTM are omitted, such as merging the input gate and forget gate to become an update z_t. Another important gate in GRU is the reset gate r_t, which helps determine how much the state value of the previous time stamp affects the current timestamp output value. The structure of the GRU algorithm can be seen in Figure 6 below. To calculate the output, GRU does a weighted sum between the activation input from the current timestamp and the output state from the previous timestamp.

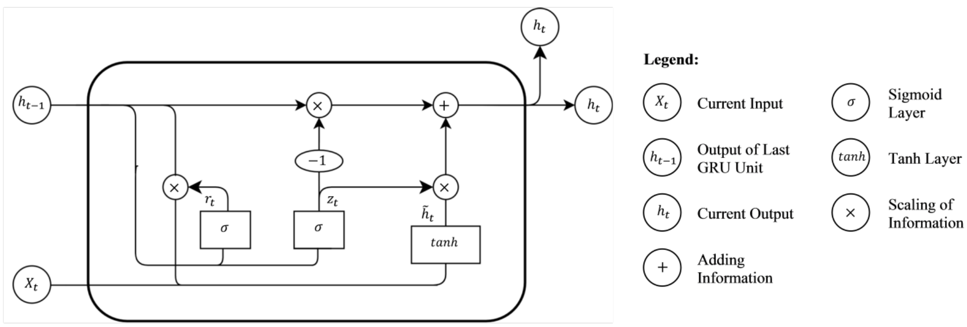

Fig. 6. GRU architecture

3.7 Training and Evaluation

As previously defined, the training and testing process uses AutoML with the AutoKeras library with pre-defined search space for model building and hyperparameters, according to Figure 3. The input data consists of two, tweet data that have been pre-processed and pure tweet data from crawling without preprocessing. Furthermore, the data is divided into 90% development and 10% testing data. Then, development data is divided into 90% training and 10% validation data.

The data will be trained and evaluated for four label pairs E/I, N/S, F/T, and J/P. The training process will use a small scale (20 epochs) and the maximum number of trials is 4. Furthermore, batch size hyperparameters with values (16, 32, 64, 128) are also used for optimization. The last stage is carried out to measure the performance result of the model. Calculation performs using a confusion matrix. Its matrix describes the performance of machine learning.

The confusion matrix consists of rows and columns that provide information about the predicted class. Some metrics to evaluate the performance of a classification model are accuracy, precision, recall, and f1-score. The formula for these metrics can be seen in Eq. (1), Eq. (2), Eq. (3) and Eq. (4), respectively. TP represents true positive, TN is true negative, FP is false positive, and FN is false negative.

$$Accuracy = \frac{TP+TN}{TP+FN+FN+TN} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

$$F1 - Score = \frac{2 \times Recall \times Precision}{Recall+Precision} \tag{4}$$

4. Result and Discussion

As explained before, this study will compare the performance of customized AutoML in handling input data in the preprocessing stage and raw data without initial preprocessing. This study uses a binary approach to predict MBTI personality. The four binary classes are E/I, N/S, F/T, and J/P. Since this study evaluates four different batch size values for each type of binary classification, the results shown in this section are the average value of the four performances produced by each batch size. The experimental results for LSTM and GRU for each binary classification can be visually seen in Figure 7 and Figure 8, respectively.

Based on Figure 7, the classification models built using LSTM with preprocessing stage generally provide better performance for the F1-Score and Precision values in the three binary classifier categories, namely E/I, N/S, and F/T. However, the opposite condition is found in the J/P binary classifier for F1-Score and Precision values. In addition, the J/P binary classifier with Preprocessing provides better performance for Recall values. In this study, "E," "N," "F," and "J" was assigned as positive classes, while "I," "S," "T," and "P" was assigned as negative classes. Therefore, the high Recall value for J/P binary classifier means the classifier has a good ability to classify the "J" class and rarely misclassifies a person with a "J" personality as someone with a "P" personality. The performance of the binary classifier on the J/P dimension tends to have a different pattern from the other dimensions and the lowest F1-Score value for the model built based on pre-processed data. It is caused by signals that can be used to identify J/P dimensions on Twitter that are very difficult to find, which aligns with previous research by Stajner and Seren [18].

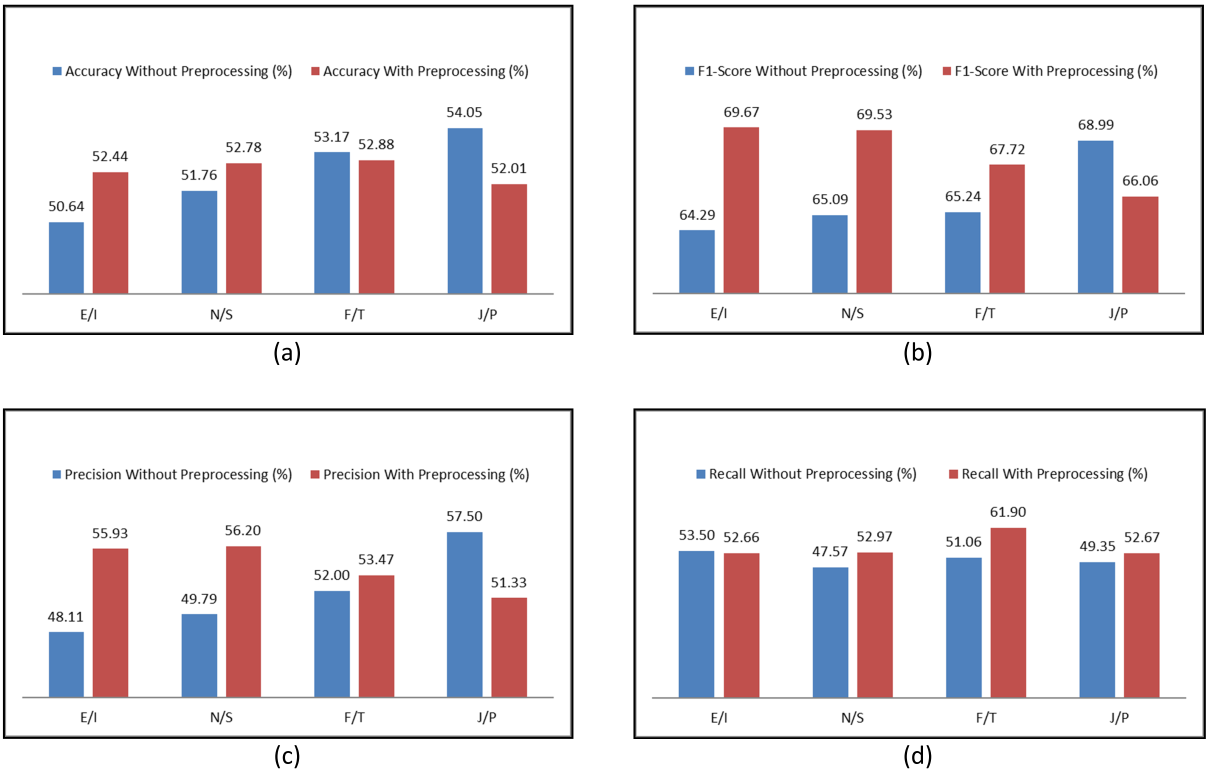

Fig. 7. Evaluation result of LSTM (a) Accuracy, (b) F1-Score, (c) Precision, (d) Recall

On the other hand, this condition does not appear in the implementation of the GRU, as depicted in Figure 8. The F1-Score value for all binary classifier models gives the best results when the classifier model is built based on pre-processed data. It is caused by, as mentioned before, GRU having a simpler architecture in terms of fewer parameters and fewer gates compared to LSTM [24]. These conditions made GRU less powerful and less adaptable to complex data without preprocessing. Nonetheless, GRU has the advantage of a more efficient processing time. However, based on the Precision value, it can be seen that the preprocessing gives a performance decrease of 0.87% in the E/I binary classifier. It is because linguistic signals in tweets made by people with dominant "E" personalities use intensifiers and exclamation points to express their feelings [18]. Those marks will be removed at the preprocessing stage, so there is no discriminating information between dimensions "E" and "I." It increases errors in classifying class I as class E.

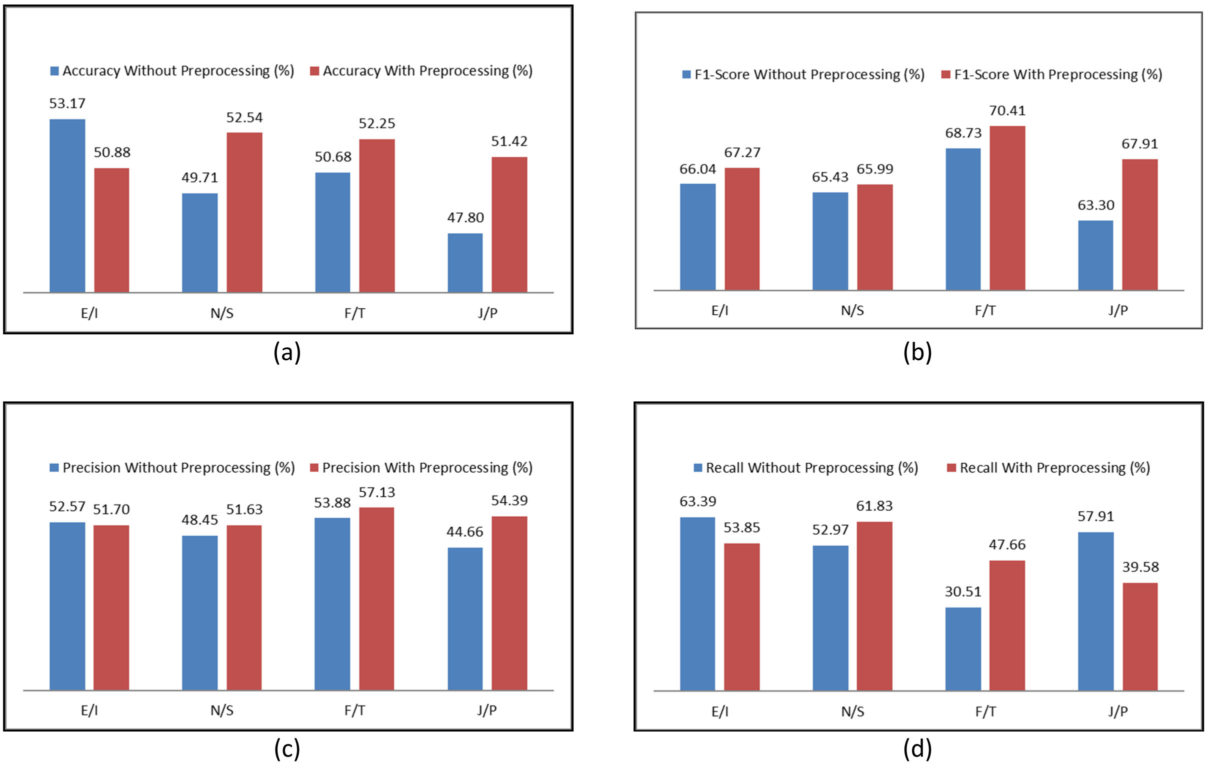

Fig. 8. Evaluation result of GRU (a) Accuracy, (b) F1-Score, (c) Precision, (d) Recall

Furthermore, Table 1 and Table 2 present the average values of all binary classifiers for both models developed based on LSTM and GRU, respectively. The two tables are intended to provide a general understanding of the effect of applying the preprocessing stage for the binary classifier approach using LSTM and GRU for personality detection.

Table 1
Average of LSTM experimental result

Label	Accuracy (%)		F1-Score (%)		Precision (%)		Recall (%)	
	Preprocessing		Preprocessing		Preprocessing		Preprocessing	
	Without	With	Without	With	Without	With	Without	With
E/I	50.64	52.44	64.29	69.67	48.11	55.93	53.50	52.66
N/S	51.76	52.78	65.09	69.53	49.79	56.20	47.57	52.97
F/T	53.17	52.88	65.24	67.72	52.00	53.47	51.06	61.90
J/P	54.05	52.01	68.99	66.06	57.50	51.33	49.35	52.67
Avg	52.41	<u>52.53</u>	65.90	<u>68.25</u>	51.85	<u>54.23</u>	50.37	<u>55.05</u>

Based on Table 1, implementing the preprocessing stage in forming a binary classifier model using LSTM generally performs better than not implementing the preprocessing stage. The increase in the average accuracy value, F1-Score, Precision, and Recall, is 0.12%, 2.35%, 2.38%, and 4.68%, respectively. The pre-processed input is cleaner and has a more consistent data structure, making it easier for the classifier to classify or result in good performance. Furthermore, the magnitude of the increase in F1-Score and Precision has an almost similar range. However, a significant increase occurred in the Recall value, which reached a value of 4.68%.

Table 2
Average of GRU experimental result

Label	Accuracy (%)		F1-Score (%)		Precision (%)		Recall (%)	
	Preprocessing		Preprocessing		Preprocessing		Preprocessing	
	Without	With	Without	With	Without	With	Without	With
E/I	53.17	50.88	66.04	67.27	52.57	51.70	63.39	53.85
N/S	49.71	52.54	65.43	65.99	48.45	51.63	52.97	61.83
F/T	50.68	52.25	68.73	70.41	53.88	57.13	30.51	47.66
J/P	47.80	51.42	63.30	67.91	44.66	54.39	57.91	39.58
Avg	50.34	51.77	65.88	67.90	49.89	53.71	51.20	50.73

In contrast, the results of Table 2 show that the application of the preprocessing stage in forming a binary classifier using GRU only gives the best performance on accuracy, F1-Score, and Precision values. The magnitude of the increase in accuracy, F1-Score, and Precision values compared to the model that was not pre-processed reached 1.43%, 2.02%, and 3.82%, respectively. However, the opposite result appears on Recall. The average value of Recall impairment is 0.47%. A significant decrease in recall of E/I and J/P dimensions causes the impairment of average recall for all dimensions.

Furthermore, the performance comparison between the classification models built using LSTM gives better results than GRU. LSTM produces F1-Score and Accuracy values of 68.25% and 52.53%, respectively. In comparison, GRU produces F1-Score values and accuracy of 67.9% and 51.77%. The increase in the value of the F1-Score is 0.35%, while the increase in accuracy is 0.76%. As mentioned, this study collected 200 tweets for each user, with 5120 Twitter users as the dataset. Therefore, the data is quite complex to classify by the generated classifier model. LSTM is a model that can deal with complex and lengthy sequences datasets since it has more parameters and gates to capture complex patterns in the data. Compared to the previous study by Jeremy and Suhartono [5], which achieved an average F1-score of 69.76%, this study has a lower average F1-score of 68.24%. However, previous research [5] used the OCEAN model, which in the case of a multi-class classification task, has a smaller number of classes when compared to the MBTI model applied to this study.

Subsequently, the performance of each classification algorithm (LTSM and GRU) will be compared. Figure 9 compares the performance of the LSTM and GRU at each batch size. It can be seen that LSTM with preprocessing results in the best F1-Score for binary classification with personality dimensions E/I and N/S for all batch values, whereas the best F1-Score for binary classification with dimensions F/T and J/P obtained by implementing GRU without preprocessing and LSTM without preprocessing, respectively.

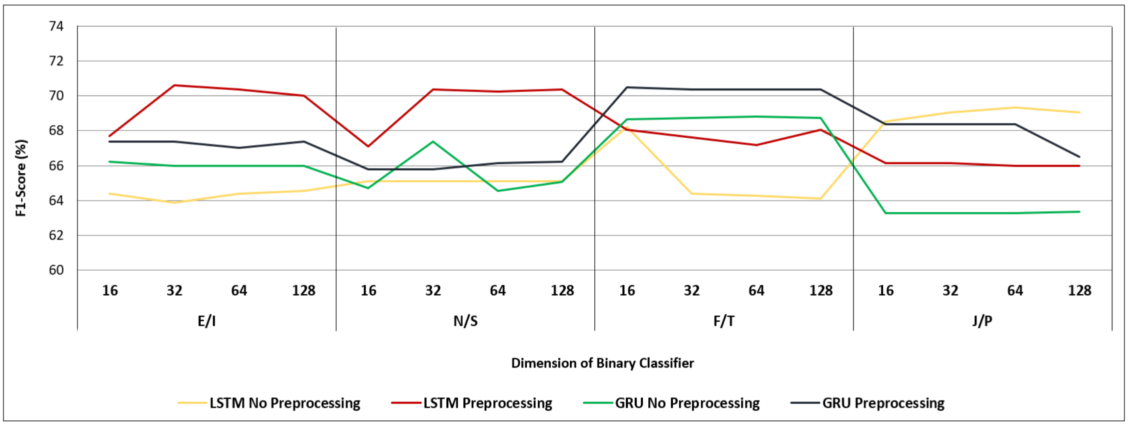

Fig. 9. Comparison of F1-Score for LSTM and GRU

5. Conclusions

This article proposes implementing AutoML with pre-defined algorithms and parameters to predict the MBTI personality model of Twitter users. Since the AutoML library implemented in this study is AutoKeras, which receives raw input and pre-processed data, the first experiment compares the performance when preprocessing is implemented (with preprocessing) or not implemented (without preprocessing). The second experiment was conducted using two pre-defined algorithms, LSTM and GRU. This study applies a binary classifier for each MBTI personality dimension, namely E/I, N/S, F/T, and J/P. The results show that applying the preprocessing stage can improve classifier performance in the form of F1-Score values for LSTM and GRU by 2.35% and 2.02%, respectively. The results of comparing the model's performance built based on LSTM outperformed GRU by 0.35% for the F1-Score and 0.76% for the accuracy value. The best model is obtained from the dataset that has been pre-processed. In conclusion, empirical results prove that preprocessing can improve classifier performance on complex data, such as detecting the MBTI personality model. The LSTM algorithm can perform better than GRU in this study because LSTM can handle complex and lengthy data sequences in line with its parameters and gates.

Acknowledgement
This study was funded by a grant from Directorate of Research, Technology and Community, Ministry of Education, Culture, Research, and Technology, Indonesia (PTM Grant 449A-70/UN7.D2/PP/VI/2023, 20 June 2023).

References

[1] Cummings, J.A. and Sanders, L. “Introduction to Psychology.” *Saskatoon, SK, Canada: University of Saskatchewan Open Press.* (2019). https://openpress.usask.ca/introductiontopsychology
[2] Keirsey, David. "Please understand me II: Temperament, character, intelligence." *(No Title)* (1998).
[3] Sperandeo, Raffaele, Giovanni Messina, Daniela Iennaco, Francesco Sessa, Vincenzo Russo, Rita Polito, Vincenzo Monda *et al.,* "What does personality mean in the context of mental health? A topic modeling approach based on abstracts published in PubMed over the last 5 years." *Frontiers in psychiatry* 10 (2020): 938. https://doi.org/10.3389/fpsyt.2019.00938
[4] Briggs, Isabel Myers, Isabel Briggs Myers, and Peter B. Myers. *Gifts differing: Understanding personality type*. Davies-Black, 1995.
[5] Jeremy, Nicholaus Hendrik, and Derwin Suhartono. "Automatic personality prediction from Indonesian user on twitter using word embedding and neural networks." *Procedia Computer Science* 179 (2021): 416-422. https://doi.org/10.1016/j.procs.2021.01.024
[6] Al Marouf, Ahmed, Md Kamrul Hasan, and Hasan Mahmud. "Comparative analysis of feature selection algorithms for computational personality prediction from social media." *IEEE Transactions on Computational Social Systems* 7, no. 3 (2020): 587-599. https://doi.org/10.1109/TCSS.2020.2966910
[7] Kusumaningrum, Retno, Iffa Zainan Nisa, Rizka Putri Nawangsari, and Adi Wibowo. "Sentiment analysis of Indonesian hotel reviews: from classical machine learning to deep learning." (2021). https://doi.org/10.26555/ijain.v7i3.737
[8] Zumma, Md Thoufiq, Jerin Akther Munia, Dipankar Halder, and Md Sadekur Rahman. "Personality prediction from twitter dataset using machine learning." In *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1-5. IEEE, 2022. https://doi.org/10.1109/ICCCNT54827.2022.9984495
[9] Agarwal, Navanshu, Lokesh Chouhan, Ishita Parmar, Sheirsh Saxena, Ridam Arora, Shikhin Gupta, and Himanshu Dhiman. "Personality prediction and classification using twitter data." In *Social Networking and Computational Intelligence: Proceedings of SCI-2018*, pp. 707-716. Springer Singapore, 2020. https://doi.org/10.1007/978-981-15-2071-6_59
[10] Majima, Seiyu, and Konstantin Markov. "Personality prediction from social media posts using text embedding and statistical features." In *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*, pp. 235-240. IEEE, 2022. https://doi.org/10.15439/2022F133

[11] Nisha, Kulsum Akter, Umme Kulsum, Saifur Rahman, Md Farhad Hossain, Partha Chakraborty, and Tanupriya Choudhury. "A comparative analysis of machine learning approaches in personality prediction using MBTI." In *Computational Intelligence in Pattern Recognition: Proceedings of CIPR 2021*, pp. 13-23. Springer Singapore, 2022. https://doi.org/10.1007/978-981-16-2543-5_2
[12] Ghazalli, Shuwaibatul Aslamiah, Hazlina Selamat, Nurulaqilla Khamis, and Mohammad Fadzli Haniff. "Short Review on Palm Oil Fresh Fruit Bunches Ripeness and Classification Technique." *Journal of Advanced Research in Applied Mechanics* 106, no. 1 (2023): 37-47. https://doi.org/10.37934/aram.106.1.3747
[13] Khan, Umair, William Pao, Nabihah Sallih, and Farruk Hassan. "Identification of horizontal gas-liquid two-phase flow regime using deep learning." *CFD Letters* 14, no. 10 (2022): 68-78. https://doi.org/10.37934/cfdl.14.10.6878
[14] Karras, Aristeidis, Christos Karras, Nikolaos Schizas, Markos Avlonitis, and Spyros Sioutas. "Automl with bayesian optimizations for big data management." *Information* 14, no. 4 (2023): 223. https://doi.org/10.3390/info14040223
[15] Jin, Haifeng, François Chollet, Qingquan Song, and Xia Hu. "Autokeras: An automl library for deep learning." *Journal of machine Learning research* 24, no. 6 (2023): 1-6.
[16] Zimmer, Lucas, Marius Lindauer, and Frank Hutter. "Auto-pytorch: Multi-fidelity metalearning for efficient and robust autodl." *IEEE transactions on pattern analysis and machine intelligence* 43, no. 9 (2021): 3079-3090. https://doi.org/10.1109/TPAMI.2021.3067763
[17] Erickson, Nick, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. "Autogluon-tabular: Robust and accurate automl for structured data." *arXiv preprint arXiv:2003.06505* (2020).
[18] Štajner, Sanja, and Seren Yenikent. "Why is MBTI personality detection from texts a difficult task?." In *Proceedings of the 16th conference of the European chapter of the association for computational linguistics: main volume*, pp. 3580-3589. 2021. https://doi.org/10.18653/v1/2021.eacl-main.312
[19] Cahyaningtyas, Risma Mustika, Retno Kusumaningrum, and Djalal Er Riyanto. "Emotion detection of tweets in Indonesian language using LDA and expression symbol conversion." In *2017 1st International Conference on Informatics and Computational Sciences (ICICoS)*, pp. 253-258. IEEE, 2017. https://doi.org/10.1109/ICICOS.2017.8276371
[20] Salsabila, Nikmatun Aliyah, Yosef Ardhito Winatmoko, Ali Akbar Septiandri, and Ade Jamal. "Colloquial indonesian lexicon." In *2018 International Conference on Asian Language Processing (IALP)*, pp. 226-229. IEEE, 2018.
[21] Song, Qingquan, Haifeng Jin, and Xia Hu. *Automated machine learning in action*. Simon and Schuster, 2022.
[22] Elsken, Thomas, Jan-Hendrik Metzen, and Frank Hutter. "Simple and efficient architecture search for convolutional neural networks." *arXiv preprint arXiv:1711.04528* (2017).
[23] Shobana, J., and M. Murali. "An efficient sentiment analysis methodology based on long short-term memory networks." *Complex & Intelligent Systems* 7, no. 5 (2021): 2485-2501. https://doi.org/10.1007/s40747-021-00436-4
[24] Ekman, Magnus. *Learning deep learning: Theory and practice of neural networks, computer vision, natural language processing, and transformers using TensorFlow*. Addison-Wesley Professional, 2021.