ICIC Express Letters
Volume **16**, Number **2**, February **2022**

ICIC International ©2022 ISSN 1881-803X
pp. 117–125

MACHINE LEARNING ALGORITHMS EXPLORATION FOR PREDICTING PERSONALITY FROM TEXT

Andry Chowanda[1], Derwin Suhartono[1], Esther Widhi Andangsari[2] and Kamal Zuhairi bin Zamli[3]

[1]Computer Science Department, School of Computer Science
[2]Psychology Department, Faculty of Humanities
Bina Nusantara University
Jl. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia
{ achowanda; dsuhartono; esther }@binus.edu

[3]Faculty of Computing
University Malaysia Pahang
Gambang, Kuantan 26300, Malaysia
kamalz@ump.edu.my

Received March 2021; accepted June 2021

Abstract. *This research explores several best machine learning algorithms to build a model for personality prediction from the text. Moreover, extensive feature sets were also explored to determine the best features to represent the dataset. The personality model implemented in this research was the Myers-Briggs Type Indicator (MBTI) model, where there is no much research done to automatically predict the MBTI personality type using machine learning. The dataset used in this research was from the (MBTI) Myers-Briggs Personality Type Dataset. Oversampling and undersampling techniques were applied to the dataset to make the dataset more balanced. The Artificial Neural Networks algorithm achieved the best result with the score of 76.3% and 77.5% for accuracy and F1 score, respectively.*

Keywords: Personality prediction, Machine learning, Features exploration, MBTI

1. **Introduction.** Personality characterizes the individuals' characteristics, patterns of thought, emotions and behaviour. Personality is unique to each individual and can be recognized by using specific tests conducted by experts. Recently, machine learning is implemented to build an automatic personality prediction from people's face, handwriting, text, and voice prosody. The prediction model can be used in many cases such as job application, marketing segmentation, and enhancing user experiences by displaying the user's preferences when interacting with the application or system. Several personality models and tools exist, such as the Big Five Personality, DISC Profile test, and the MBTI (Myers-Briggs Type Indicator). There is not much research in predicting personality from text based on the MBTI model by using machine learning. Hence, this research aims to explore several machine learning techniques to build a model that can predict the MBTI model of personality by using text modality. MBTI is a personality psychological instrument constructed by Myers and Briggs based on Jung theory [1]. It focuses on the character and dynamic of each personality type explained in it. This instrument does not use alone as a personality profiling assessment but also as a complementary or supporting instrument to explain other aspects, such as career profiling [2] and students' learning style [3]. Users need to understand the four dichotomies are constructed in MBTI. Everyone has these four dichotomies, as Jung explained in his theory [1]. The perception of activities is represented by Sensing (S) and Intuition (I); rational functions are represented by Thinking

DOI: 10.24507/icicel.16.02.117

(T) and Feeling (F); attitudes or internal energies are represented by Extroversion (E) and Introversion (I), and attitudes or outer world orientation are represented by Judging (J) and Perceiving (P) [1]. The MBTI instrument's uniqueness is on its four dichotomies, which become a reflection of an individual's psychological disposition [1]. This instrument measures the balancing scores of each dichotomy that give preference meaning of other personality type. As a self-report instrument, MBTI gives final judgment about the result to users' judgment themselves. Users have a part in which personality type is the best fit for them. The dataset used was the (MBTI) Myers-Briggs Personality Type Dataset. The research contributes to the explorations of machine learning algorithms and extensive feature sets, and hyper-parameters exploration. Five algorithms explored in this papers are: Generalized Linear Model, Fast-Large Margin, Gradient Boosted Tree, Support Vector Machine, and Artificial Neural Network. The result shows that the Artificial Neural Network algorithm achieved the best score of 0.763, 0.814, 0.775, 0.763, 0.237 and ± 1.1 for recall, precision, F1, accuracy, error, and standard deviation of the accuracy score, respectively. The rest of the paper is structured as follows. Recent related work related to personality prediction is described in the next section. Section 3 illustrates the algorithms design proposed in this research. The results are then discussed in Section 4. Finally, Section 5 concludes the results and demonstrates future direction of the research.

2. **Related Work.** The number of research in personality prediction with machine learning has been increasing over these decades. One coined the general term in personality prediction, perception, and synthesis using machine learning as personality computing [4]. This research focuses only on personality prediction using machine learning. Predicting personality using machine learning can be done using several modalities or features such as text, visual (or video), and voice prosody [4-10]. Several algorithms can be used to predict personality using machine learning. The choices of algorithms depend on the modalities or features used in training. Traditional machine learning algorithms are still used in this research area, as they still provide excellent results depending on the type of datasets [4, 9]. Algorithms such as Support Vector Machine and Gradient Boosted Tree are still providing high accuracy for predicting personality from text [4,9]. Recently, deep learning techniques are also applied to predicting personality from text, visual, and voice prosody features. Convolutional neural network architecture is best to handle visual features, while recurrent neural network architectures such as long-short term memory and gated recurrent unit are best to handle sequential or temporal features (e.g., sequential text, visual, and voice prosody) [4-10]. Several features extraction techniques can be used in a text modality, namely N-Grams, Linguistic Inquiry and Word Count (LIWC), Machine Reading Comprehension (MRC), Term Frequency-Inverse Document Frequency (TF-IDF), Part-of-Speech (POS), and the combination of the techniques [4, 5, 7, 11]. More advanced features extraction techniques are currently implemented to represent the text features, for example, Word2Vec, Glove, FastText, and Bidirectional Encoder Representations from Transformers (BERT) [7, 8, 10, 12]. Features representation model such as BERT is claimed to have a high generalization to be implemented in major natural language processing problems [12, 13]. Finally, most of the research used the Big Five (OCEAN) personality type as the classes [14]. This research aims to explore several machine learning algorithms to model a personality prediction from the text by using MBTI (Myers-Briggs Type Indicator) personality model [1-3].

3. **Machine Learning Algorithm.** This research aims to explore various machine learning techniques to learn a model that can predict personality from the text. The dataset used in this research was (MBTI) Myers-Briggs Personality Type Dataset. The dataset consists of 16 distinct personalities as the classes. Table 1 demonstrates the dataset profile. There are a total of 7818 data with imbalanced classes problem. Columns "Class"

indicates the MBTI personality type. Columns "Actual" and "% @Class A" show the actual size and the percentage of each class, respectively. As the dataset is highly imbalanced, both under-sampling and over-sampling techniques were implemented to the data. The best formation of the data is shown in Table 1. Columns "Sampling", "% Changes" and "% @Class S" demonstrate the size of data after sampling methods applied, how significant the percentage changes, and the percentage of each class, respectively. Columns "Train" and "Test" indicate the number of data used in each class's training and testing phase, respectively. In summary, from a total of 7818, only 4200 data were used in the experiment with 85% : 15% train and test split.

Table 1. The dataset profile

Class	**Actual**	**% @Class A**	**Sampling**	**% Changes**	**% @Class S**	**Train**	**Test**
INTP	1177	15.1%	300	−75%	7%	255	45
INFJ	1333	17.1%	300	−77%	7%	255	45
INTJ	972	12.4%	300	−69%	7%	255	45
ENFJ	168	2.1%	300	79%	7%	255	45
INFP	1660	21.2%	300	−82%	7%	255	45
ENTJ	213	2.7%	300	41%	7%	255	45
ENFP	609	7.8%	300	−51%	7%	255	45
ISTP	292	3.7%	300	3%	7%	255	45
ISFJ	153	2.0%	300	96%	7%	255	45
ENTP	611	7.8%	300	−51%	7%	255	45
ISTJ	179	2.3%	300	68%	7%	255	45
ISFP	252	3.2%	300	19%	7%	255	45
ESTP	78	1.0%	150	92%	4%	128	22
ESFP	46	0.6%	150	226%	4%	128	22
ESTJ	35	0.5%	150	329%	4%	128	22
ESFJ	40	0.5%	150	275%	4%	128	22
TOTAL	**7818**	**100%**	**4200**		**100%**	**3572**	**628**

Several pre-processing techniques were implemented to the data to enhance the quality of the dataset. The techniques used in this experiment were tokenization, noise removal, stop-words filtering, stemming, filtering tokens based on length, case transformation, and extracting text features from the data. Irrelevant and not meaningful text (e.g., header, JSON tag, HTML tag) were removed before the tokenisation process took place. Tokenisation is a process to convert sentences into a small block of words (tokens). The next step is stop-words filtering, where common words in the text were filtered and removed from the text. Tokens with more than 20 characters were removed in this experiment to increase the training speed. Next, the stemming process ensures only the root of the words saved in the dataset. Finally, essential features were extracted by using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization technique.

Five machine learning algorithms were explored in this research. Table 2 demonstrates the algorithms implemented in this experiment. The hyper-parameters and feature sets for each algorithm were also explored to find the dataset's optimal solutions. Generalized Linear Model uses 700 features, Fast-Large Margin implements 800, Artificial Neural Network and Gradient Boosted Tree use 1000 features, and Support Vector Machine implements 750 features. Table 3 illustrates the architecture and hyper-parameters settings for Artificial Neural Network. The architecture consists of six layers, where one layer for input layer (1000 neurons with RELU activation function), one layer for the classification layer (16 neurons with Softmax activation function), and four hidden layers (200 neurons

Table 2. Automatic feature sets exploration & selection

Algorithm	**Evaluated feature sets**	**Features used**
Generalized Linear Model	114	700
Fast-Large Margin	114	800
Artificial Neural Network	114	1000
Gradient Boosted Tree	114	1000
Support Vector Machine	114	750

Table 3. Artificial Neural Network hyper-parameters setting

Layer	**Unit**	**Type**	**Dropout**	**L1**	**L2**
1	1000	Rectifier	0%	0.000010	0
2	200	Rectifier	0%	0.000010	0
3	200	Rectifier	0%	0.000010	0
4	200	Rectifier	0%	0.000010	0
5	200	Rectifier	0%	0.000010	0
6	16	Softmax	0%	0.000010	0

each with RELU activation function). No dropout is implemented in the architecture. However, the L1 regularization was implemented to deal with the overfitting problem.

4. **Results and Discussion.** In this experiment, five algorithms were explored, together with 4,200 data (3,572 for training, 628 for testing). The hyper-parameters and feature sets for each algorithm were also explored to find the dataset's optimal solutions. The best hyper-parameters and features extraction then were implemented (see Table 2). During the training process, some of the hyper-parameters combinations for each algorithm were also explored and evaluated in Fast-Large Margin, Gradient Boosted Tree, and Support Vector Machine algorithms. The hyper-parameters and architecture settings for Artificial Neural Network are illustrated in Table 3.

Table 7 demonstrates the overall results for all algorithms. The Generalized Linear Model algorithm uses 700 best features to train the model. In this algorithm, there was no additional hyper-parameters exploration. Generalized Linear Model provides the classification accuracy score of 0.713 with a standard deviation value of ± 1.0, the classification error value of 0.287, recall score of 0.717, the precision score of 0.753, and F1 score of 0.723. Tables 4-6 demonstrate the exploration of the hyper-parameters combination for each algorithm. Table 4 illustrates the combination of hyper-parameters exploration for Fast-Large Margin. From a total of 114 feature sets explored (each has 1,000 features), only 800 best features were used in this experiment. Moreover, the combination of the parameter C, where it indicates the error term's penalty parameter, was explored. In this

Table 4. Fast-Large Margin hyper-parameters exploration

C	**Error**
0.001	0.742
0.01	0.742
0.1	0.513
1	0.318
10	0.297
100	**0.290**
1000	0.311

Table 5. Gradient Boosted Tree hyper-parameters exploration

Number of trees	**Max depth**	**LR**	**Error rate**
30	2	0.001	0.365
90	2	0.001	0.360
150	2	0.001	0.357
30	4	0.001	0.414
90	4	0.001	0.395
150	4	0.001	0.393
30	7	0.001	0.418
90	7	0.001	0.411
150	7	0.001	0.402
30	2	0.010	0.343
90	2	0.010	0.313
150	2	0.010	0.294
30	4	0.010	0.378
90	4	0.010	0.338
150	4	0.010	0.319
30	7	0.010	0.387
90	7	0.010	0.342
150	7	0.010	0.328
30	2	0.100	0.279
90	**2**	**0.100**	**0.270**
150	2	0.100	0.270
30	4	0.100	0.287
90	4	0.100	0.280
150	4	0.100	0.276
30	7	0.100	0.315
90	7	0.100	0.307
150	7	0.100	0.296

Table 6. Support Vector Machine hyper-parameters exploration

Gamma	**C**	**Error**
0.005	10	0.742
0.050	10	0.742
0.500	10	0.513
5.000	10	0.318
0.005	100	0.297
0.050	**100**	**0.290**
0.500	100	0.278
5.000	100	0.289
0.005	1000	0.318
0.050	1000	0.388
0.500	1000	0.492
5.000	1000	0.513

research, the parameter C was explored in 7 different settings from 0.001 to 1,000 multiplied by 10 in every phase. The best classification error was achieved by the parameter C of 100 with a classification error of 0.29. With the best hyper-parameter and features,

TABLE 7. Overview results

Algorithm	Recall	Precision	F1	Accuracy	Error	STD Acc
Generalized Linear Model	0.717	0.753	0.723	0.713	0.287	**±1.0**
Fast-Large Margin	0.72	0.723	0.733	0.71	0.29	±2.0
Artificial Neural Network	**0.763**	**0.814**	**0.775**	**0.763**	**0.237**	±1.1
Gradient Boosted Tree	0.73	0.766	0.743	0.73	0.27	±1.9
Support Vector Machine	0.75	0.779	0.755	0.742	0.258	±1.1

Fast-Large Margin achieved the classification accuracy score of 0.71 with a standard deviation value of ±2.0, recall score of 0.72, the precision score of 0.723, and F1 score of 0.733.

The dataset was also trained using six layers of Artificial Neutral Network with L1 regularization. The best 1,000 features extracted from the dataset (in 114 feature sets) were implemented as the input. The features then were processed in four RELU layers of 200 units. The Artificial Neural Network provides the best classification accuracy score of 0.763 with a standard deviation value of ±1.1, the classification error value of 0.237, recall score of 0.763, the precision score of 0.814, and F1 score of 0.775. The Artificial Neutral Network algorithm offers the best results compared to the other algorithms. The next algorithm explored in this research was Gradient Boosted Tree. Like Artificial Neural Network, the Gradient Boosted Tree implements the best 1,000 features from 114 feature sets extracted from the dataset. In addition, 27 combinations of hyper-parameters were also explored during training phases (see Table 5). The hyper-parameters explored in the training phases were the number of trees $(30, 90, 150)$, the maximum value of depth for the trees $(2, 4, 7)$, and the learning rate $(0.001, 0.01, 0, 1)$. The best combination achieved was 90 trees with a maximum depth of two and a learning rate value of 0.1. This combination provides the best classification error rate of 0.27, classification accuracy score of 0.73 with a standard deviation value of ±1.9, recall score of 0.73, the precision score of 0.766, and F1 score of 0.743 (see Table 7). Finally, a Support Vector Machine was also implemented to train the model with the best 750 features used in training (extracted from 114 feature sets from the dataset). Table 6 illustrates the exploration of hyper-parameters in Support Vector Machine. There were 12 combinations of Gamma (γ) and the cost parameter C. The Gamma values were set from 0.005 to 5 with the multiplication of 10 for each iteration, while the parameter C was set from 10 to 1,000 with the multiplication of 10 for each iteration. Support Vector Machine algorithm achieved the best classification accuracy score of 0.742 with a standard deviation value of ±1.1, the classification error value of 0.258, recall score of 0.75, the precision score of 0.779, and F1 score of 0.755 (see Table 7).

Tables 8 and 9 demonstrate the detailed results for each class in all training algorithms proposed in this research. This dataset's overall best algorithm was the Artificial Neural Network (classification accuracy score of 0.763, classification error value of 0.237 and standard deviation value of ±1.1). In contrast, the algorithm that achieved the lowest performances overall was the Fast-Large Margin (classification accuracy score of 0.71, classification error value of 0.29 and standard deviation value of ±2.0). The trained model using Fast-Large Margin also had more data spread out from the average compared to the other algorithms. From 16 classes, INFP and INFJ resulted in high recall scores and low precision scores in most of the algorithms. For example, in the Artificial Neural Network algorithm, the INFP achieved 0.933 for recall score, 0.500 for the precision score, and in the Support Vector Machine model, the INTJ achieved a score of 0.844 and 0.500 for recall and precision scores, respectively. Both classes also achieved the lowest F1 score among the other classes. In contrast, the ESTJ and ESTP classes have high precision score and low recall score in most of the algorithms. For example, in the Artificial Neural Network

Table 8. Artificial Neural Network results

Algorithm	Class	Recall	Precision	F1
Artificial Neural Network	INTP	0.756	0.791	0.773
	INFJ	0.756	0.694	0.723
	INTJ	0.756	0.829	0.791
	ENFJ	0.756	0.895	0.819
	INFP	**0.933**	0.500	0.651
	ENTJ	0.711	0.744	0.727
	ENFP	0.622	0.966	0.757
	ISTP	0.889	0.606	0.721
	ISFJ	0.667	0.857	0.750
	ENTP	0.867	0.750	0.804
	ISTJ	0.667	0.882	0.759
	ISFP	0.778	0.897	0.833
	ESTP	0.864	0.792	0.826
	ESFP	0.682	0.882	0.769
	ESTJ	0.727	**1.000**	0.842
	ESFJ	0.773	0.944	**0.850**

algorithm, the ESTJ achieved 0.727 and 1.000 for recall and precision score, respectively. This is most likely due to the imbalanced class problem, as most introverted classes (e.g., INTP, INFJ) have a larger number in the dataset than the extroverted classes (e.g., ENTP, ESTJ). Finally, the ESTP class achieved the best F1 score in almost all algorithms. The best F1 score achieved by the ESTP class was the one trained with Fast-Large Margin (0.889) followed by Generalized Linear Model (0.864).

5. **Conclusion and Future Work.** Five algorithms were explored to build models that can predict personality from a text in this research. Generalized Linear Model, Fast-Large Margin, Artificial Neural Network, Gradient Boosted Tree, and Support Vector Machines were proposed in the (MBTI) Myers-Briggs Personality Type Dataset. Moreover, more than 100,000 features and almost 50 combinations of hyper-parameters were also explored and evaluated to find the optimal settings and features. This research aims to comprehensively explore the machine learning algorithms to provide the best model for personality prediction from the text. Table 7 illustrates the overview of all results for all the algorithms, and Table 9 as well as Table 8 demonstrate the detailed results of each class in the dataset for all the algorithms. The best algorithm in this experiment was the Artificial Neural Network with the score of 0.763, 0.814, 0.775, 0.763, 0.237 and $\pm$1.1 for recall, precision, F1, accuracy, error, and standard deviation of the accuracy score respectively.

On the other hand, the model trained with Fast-Large Margin achieved the lowest scores with the score of 0.72, 0.723, 0.733, 0.71, 0.29, and $\pm$2.0 for recall, precision, F1, accuracy, error, and standard deviation of the accuracy score respectively. The ESTP class achieved the best F1 score in almost all algorithms in the detailed results for each class. The best F1 score achieved by the ESTP class was the one trained with Fast-Large Margin (0.889) followed by Generalized Linear Model (0.864). As each class's number is not balanced in the dataset, some of the classes in several of the algorithms had a not balanced score of recall and precision (e.g., high recall – low precision or low recall – high precision).

For the future direction of this research, two immediate actions can be planned. The first one is to explore more enhanced features representation such as BERT [13], Word2Vec, Glove, and explore more recent deep learning algorithms such as convolutional neural network, long-short term memory, transformers [13]. The second future direction for this

Table 9. Detailed results

Alg	Class	Recall	Precision	F1	Alg	Class	Recall	Precision	F1
GLM	INTP	**0.867**	0.591	0.703	GBT	INTP	0.689	0.646	0.716
	INFJ	0.733	0.623	0.673		INFJ	0.689	0.721	0.679
	INTJ	0.778	0.593	0.673		INTJ	0.711	0.681	0.628
	ENFJ	0.644	0.744	0.690		ENFJ	0.733	0.805	0.782
	INFP	0.733	0.569	0.641		INFP	0.756	0.642	0.689
	ENTJ	0.667	0.833	0.741		ENTJ	0.689	0.861	0.725
	ENFP	0.644	0.674	0.659		ENFP	0.756	0.708	0.721
	ISTP	0.756	0.791	0.773		ISTP	**0.844**	0.884	0.776
	ISFJ	0.644	0.806	0.716		ISFJ	0.778	0.593	0.828
	ENTP	0.667	0.750	0.706		ENTP	0.667	0.732	0.701
	ISTJ	0.622	0.903	0.737		ISTJ	0.711	0.615	0.750
	ISFP	0.756	0.829	0.791		ISFP	0.756	0.723	0.824
	ESTP	0.864	0.864	**0.864**		ESTP	0.818	**0.947**	**0.851**
	ESFP	0.773	0.548	0.642		ESFP	0.762	0.889	0.816
	ESTJ	0.682	**1.000**	0.811		ESTJ	0.636	0.933	0.789
	ESFJ	0.636	0.933	0.757		ESFJ	0.682	0.882	0.800
FLM	INTP	0.756	0.607	0.673	SVM	INTP	0.756	0.680	0.716
	INFJ	0.711	0.485	0.577		INFJ	0.822	0.578	0.679
	INTJ	0.844	0.418	0.559		INTJ	0.844	0.500	0.628
	ENFJ	0.622	0.778	0.691		ENFJ	0.756	0.810	0.782
	INFP	0.667	0.652	0.659		INFP	0.689	0.689	0.689
	ENTJ	0.622	0.848	0.718		ENTJ	0.644	0.829	0.725
	ENFP	0.689	0.775	0.729		ENFP	0.689	0.756	0.721
	ISTP	0.756	0.810	0.782		ISTP	0.733	0.825	0.776
	ISFJ	0.756	0.872	0.810		ISFJ	0.800	0.857	0.828
	ENTP	0.533	0.828	0.649		ENTP	0.600	0.844	0.701
	ISTJ	0.622	0.903	0.737		ISTJ	0.667	0.857	0.750
	ISFP	0.8	0.878	0.837		ISFP	0.778	0.875	0.824
	ESTP	**0.909**	0.870	**0.889**		ESTP	**0.909**	0.800	**0.851**
	ESFP	0.864	0.792	0.826		ESFP	**0.909**	0.741	0.816
	ESTJ	0.727	**1.000**	0.842		ESTJ	0.682	**0.938**	0.789
	ESFJ	0.636	0.933	0.757		ESFJ	0.727	0.889	0.800

research is applying and implementing the personality prediction model to other application such as a virtual assistant or affective systems such as virtual humans [15, 16], and others. Additionally, a large and balanced dataset can also be collected from social media and annotated by the experts for personality prediction using Myers-Briggs Type Indicator (MBTI).

Acknowledgment. This work is supported by Bina Nusantara University Grant. We also would like to extend our gratitude to NVIDIA Corporation with the donation of the Titan V Pascal GPU used for this research.

REFERENCES

[1] I. B. Myers, M. H. McCaulley, N. L. Quenk and A. L. Hammer, *MBTI Manual: A Guide to the Development and Use of the Myers-Briggs Type Indicator*, Consulting Psychologists Press, 1998.

[2] W. C. Buboltz, Jr., P. Johnson, C. Nichols, M. A. Miller and A. Thomas, MBTI personality types and SII personal style scales, *Journal of Career Assessment*, vol.8, no.2, pp.131-145, 2000.

[3] R. Harrington and D. A. Loffredo, MBTI personality type and other factors that relate to preference for online versus face-to-face instruction, *The Internet and Higher Education*, vol.13, nos.1-2, pp.89-95, 2010.

[4] A. Vinciarelli and G. Mohammadi, A survey of personality computing, *IEEE Trans. Affective Computing*, vol.5, no.3, pp.273-291, 2014.

[5] M. Skowron, M. Tkalčič, B. Ferwerda and M. Schedl, Fusing social media cues: Personality prediction from Twitter and Instagram, *Proc. of the 25th International Conference Companion on World Wide Web*, pp.107-108, 2016.

[6] T. Tandera, D. Suhartono, R. Wongso, Y. L. Prasetio et al., Personality prediction system from facebook users, *Procedia Computer Science*, vol.116, pp.604-611, 2017.

[7] V. Ong, A. D. S. Rahmanto, Williem and D. Suhartono, Exploring personality prediction from text on social media: A literature review, *Internetworking Indonesia*, vol.9, no.1, pp.65-70, 2017.

[8] A. Chowanda and A. D. Chowanda, Recurrent neural network to deep learn conversation in Indonesian, *Procedia Computer Science*, vol.116, pp.579-586, 2017.

[9] B. Y. Pratama and R. Sarno, Personality classification based on Twitter text using Naive Bayes, KNN and SVM, *2015 International Conference on Data and Software Engineering (ICoDSE)*, pp.170-174, 2015.

[10] J. Sun, Z. Tian, Y. Fu, J. Geng and C. Liu, Digital twins in human understanding: A deep learning-based method to recognize personality traits, *International Journal of Computer Integrated Manufacturing*, pp.1-14, 2020.

[11] M. Murata, K. Orikane and R. Akae, Automatic selection and analysis of verb and adjective synonyms from Japanese sentences using machine learning, *International Journal of Innovative Computing, Information and Control*, vol.15, no.6, pp.2135-2147, 2019.

[12] A. Kazameini, S. Fatehi, Y. Mehta, S. Eetemadi and E. Cambria, Personality trait detection using bagged SVM over BERT word embedding ensembles, *arXiv Preprint*, arXiv: 2010.01309, 2020.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, Attention is all you need, *arXiv Preprint*, arXiv: 1706.03762, 2017.

[14] O. P. John, S. Srivastava et al., *The Big-Five Trait Taxonomy: History, Measurement, and Theoretical Perspectives*, University of California Berkeley, 1999.

[15] R. Sutoyo, A. Chowanda, A. Kurniati and R. Wongso, Designing an emotionally realistic chatbot framework to enhance its believability with AIML and information states, *Procedia Computer Science*, vol.157, pp.621-628, 2019.

[16] W. Zhu, A. Chowanda and M. Valstar, Topic switch models for dialogue management in virtual humans, in *Intelligent Virtual Agents. IVA 2016. Lecture Notes in Computer Science*, D. Traum, W. Swartout, P. Khooshabeh, S. Kopp, S. Scherer and A. Leuski (eds.), Cham, Springer, 2016.