Jurnal Elektronika dan Telekomunikasi (JET), Vol. 21, No. 2, December 2021, pp. 104-111
Accredited by RISTEKDIKTI, Decree No: **158/E/KPT/2021**
doi: 10.14203/jet.v21.104-111

Prediction Of Myers-Briggs Type Indicator Personality Using Long Short-Term Memory

Mawadatul Maulidah [a], *, Hilman Ferdinandus Pardede [a, b]

[a] Graduate School for Computer Sciences
Nusa Mandiri University
Jl. Raya Margonda No.545, Pondok Cina, Beji
Depok, Indonesia
[b] Research Center for Informatics
Indonesian Institute of Sciences
Jl. Cisitu No. 21/154D
Bandung, Indonesia

Abstract

Personality is defined as the mix of features and qualities that make up an individual's particular character, including thoughts, feelings, and behaviors. With the rapid development of technology, personality computing is becoming a popular research field by providing users with personalization. Many researchers have used social media data to automatically predict personality. This research uses a public dataset from Kaggle, namely the Myers-Briggs Personality Type Dataset. The purpose of this study is to predict the accuracy and F1-score values so that the performance for predicting and classifying Myers–Briggs Type Indicator (MBTI) personality can work optimally by using attributes from the MBTI dataset, namely posts and types. Predictive accuracy analysis was carried out using the Long Short-Term Memory (LSTM) algorithm with random oversampling technique with the Imblearn library for MBTI personality type prediction and comparing the performance of the method proposed in this study with other popular machine learning algorithms. Experiments show that the LSTM model using the RMSprop optimizer and learning speed of 10^{-3} provides higher performance in terms of accuracy while for the F1-score the LSTM model using the RMSprop Optimizer and learning speed of 10^{-2} gives a higher value than the proposed machine learning algorithm so that the model MBTI dataset using LSTM with random oversampling can help in identifying the MBTI personality type.

Keywords: Long short-term memory, myers–briggs type indicators, personality, prediction, random oversampling.

I. Introduction

Personality plays an important role in predicting many individual factors such as mental and physical health, fitness, and career well-being. Therefore, gaining deep insight into a person's personality type is the key. Katherine Cook and her daughter, Isabel Briggs Myers, were the first to come up with this method, which is a development of the personality theory that had previously been proposed by Carl Gustav Jung. The MBTI serves as an instrument that people use in trying to better understand their own beliefs and motivations. Although the reliability and validity of the MBTI are often criticized, MBTI remains the most highly used method of personality measurement [1].

There has been a myriad of studies that aim to summarize various aspects of personality research. Amirhosse, et al. has conducted research on Machine Learning Approach for Prediction of Personality Type Based on Myers–Briggs Type Indicator in 2020 [2] by developing a new machine learning method to automate the process of detecting meta programs and predicting personality types based on MBTI personality type indicators. The toolkit in this study uses natural language processing toolkit (NLTK) and XGBoost as the proposed method with Gradient Boosting as a library and Python to implement machine learning algorithms. Abidin et al. [3] conducted research by comparing the performance of the proposed method with other popular machine learning algorithms. The experimental evaluation results give Random Forest a higher performance with the highest accuracy score than the other three machine learning algorithms, thereby helping companies to identify personality types in selecting suitable candidates. Tutupoly et al. [4] and Mehta et al. [5] has surveyed research on personality prediction based on a few data sources and modalities. Both of them gave specific observations on deep learning-based approaches.

Research on personality prediction was also carried out by Bharadwaj et al. [6] by presenting an analysis of text written by a person such as an essay, tweet, or blog post and creating a personality profile of that person using various machine learning models with a combination of feature vectors then compared, implemented, and provided solutions. SVM models with S/N dimensions have an accuracy of up to 88%. Li et al. [7] applied the text mining method to individual psychometric assessments. A methodology is described for reducing large, unstructured text to low-dimensional numerical feature vectors, from which the authors' text-based MBTI indicators can be reliably deduced. The best results from the KNN model with accuracy values for each dimension are as follows: E - I 95%; J - P 76.25%;

* Corresponding Author.
Email: 14002373@nusamandiri.ac.id

Received: August 02, 2021 ; Revised: October 12, 2021
Accepted: December 07, 2021 ; Published: December 31, 2021

F - T 91.25%, N - S 90%. Keh and Cheng [8] have conducted a study using a pre-trained language model to predict MBTI personality type based on scratched labeled text. The proposed method achieves an accuracy of 0.47 to correctly predict all 4 personality types and 0.86 to correctly predict at least 2 personality types.

Further related research, Choong and Varathan [9] evaluated the performance of individual features and classifiers for the J/P (Judging-Perceiving) dichotomy of predictive performance between character-level TF, TF-IDF and word-level TF, TF-IDF in personality computing. Although their results were obtained from the same settings as the previous research, their study still managed to outperform the previous studies. This study also compares five machine learning algorithms, and the LightGBM Model with Combo Feature on the Kaggle dataset resulted in the highest F1-Macro score. Patel et al. [10] conducted research with the aim of gaining knowledge about the user's personality using the social media platform of the user in question. This social media platform can be Facebook or Twitter. The system is capable to collect tweets from users when given their Twitter handle. The algorithm used is able to process tweets and produce the required results. In addition, the system is also able to provide accurate results efficiently by providing the user's personality type. Frkovic et al. [11] compared the structural hyperparameters, the hyperparameters investigated in this study are the number of hidden layers and layer size. Through a number of experiments, we demonstrated the choice of hyperparameters and concluded with recommendations for selecting general hyperparameters. The model with the highest F1-score is the LSTM.

Further research on MBTI personality was carried out by Hernandez and Knight [12] by conducting research using various types of repetitive neural networks (RNNs) such as simple RNNs, gate repeat units (GRUs) which are gate mechanisms in repetitive neural networks, long short-term memory (LSTM) which is a neural network architecture. An artificial recurrent neural network architecture is used in the field of deep learning, and Bidirectional LSTM to construct a classifier capable of predicting a person's MBTI personality type based on a sample of text from their social media posts. They compared the results and found that LSTM gave the best results. They used the same dataset used in their previous study, the Myers–Briggs Personality Type Dataset from Kaggle. Further research was conducted with the same dataset by Cui and Qi [13] using Baseline, Logistic Regression, Naïve Bayes, and SVM algorithms to predict a person's MBTI personality type from one of their social media posts. They compared the results of all the proposed methods and found that SVM performed better.

From several related studies that have been described above, only applying performance evaluation techniques in the form of accuracy values without paying attention to the condition of the data in a balanced state or not. The condition of unbalanced data occurs in the MBTI personality dataset where some personality types have more data than others. The imbalanced problem is a problem that arises, namely the classification performance value shows a high accuracy value because the number of major classes is very large. However, it has a very poor classification performance when classifying data from minor classes. Regarding the problem of imbalanced data, one of the strategies that can be applied is the sampling technique, both oversampling and undersampling.

Some of the research challenges we found included an imbalance in the amount of data for certain personality classes. This can be a problem, because systems with this condition often perform well in the majority class but often fail to predict personality for the minority class. So, we do oversample to produce "more data" for the minority class. In addition, one of the main advantages of oversampling is that no information is lost from the majority and minority classes during the process. For Machine Learning algorithms that are affected by skewed distributions, such as artificial neural networks (ANN) and SVM, random oversampling is a very effective technique to improve performance. However, tuning the target class distribution is recommended in many scenarios because seeking a balanced distribution for a highly unbalanced data set can lead to overfitting of the minority class distribution, which in turn results in increased generalization errors. Another thing we have to be aware of is the rising cost of computing. Increasing the number of examples in the minority class distribution (especially for highly skewed data sets) can result in computational improvements when we train our model and given that the model sees the same example multiple times, this is not a good thing. Nonetheless, oversampling is a pretty good solution and should be tested.

Myers-Briggs Type Indicator (MBTI) is commonly used to predict personality. However, this test includes answering long questionnaire questions and is often inaccurate since people who take the test may fake it. Studies show that how we write for instance in social media could also reflect our personality. With the rapid development of the internet and social media, users' posting would reflect better on their personality since they would be more openly expressing their personal views and insights about their lives. Postings from social media platform such as Facebook or Twitter can be analyzed, so it can help uncover many types of interactions. It can be used to predict a person's suitable job as well as find out about his or her efficiency in the same job; professionality, romantic attitude, and nature can also be studied. Hence, extracting information from social media can be done using what is called text mining. This has been studied by many researchers. Here, we apply LSTM for predicting people's personality. LSTM is a type of Machine Learning that is based on the Recurrent Neural Network (RNN) approach that can predict the current MBTI personality type. The LSTM network is the best choice thanks to its ability to store memory for a long time at the same time; complex correlations between data provide information that is very useful in determining predictions.

The purpose of this study was to obtain a LSTM model with and without random oversampling which has the highest and the best accuracy for MBTI personality prediction and has practical and theoretical benefits. This research provides practical benefits, namely providing scientific contributions to research in the field of classification and prediction of data mining and text mining, especially to predict personality types. While the theoretical benefit is to provide empirical evidence regarding the application of the Long Short Term Memory model and other classification models such as Extreme Gradient Boosting (XGB), Logistic Regression (LR), Stochastic Gradient Descent (SGD), Random Forest (RF), Support Vector Machine (SVM), K-nearest Neighbors (KNN) with and without random oversampling on MBTI personality prediction.

II. METHOD

The Methods chapter describes the dataset, preprocessing data, the proposed method, and the comparison method that will be used in this study.

A. Dataset

The dataset in this study was taken from the online personality forum at personalitycafe.com and is also available free of charge on Kaggle (https://www.kaggle.com/datasnaek/mbti-type/), which is an online data science community. This dataset contains 8675 posts from forum users. Each user has 50 samples of user post text on social media, the sample text is separated by the order '|||' with a total of 433,750 user comments. The MBTI dataset only has 2 columns, namely type, and posts which have 16 MBTI personality types (e.g., Introvert-Intuition-Thinking-Judging (INTJ), Extraverted-Sensing-Feeling-Perceiving (ESFP)). Each user has an MBTI type which is labeled with four dimensions, namely Introversion (I)/Extraversion (E), Intuition (N)/Sensing (S), Thinking (T)/Feeling (F), and Judging (J)/Perception (P). There are no null values in the dataset and all values are textual, so they must be converted to numeric. In addition to the dataset, the results of their MBTI tests are also included. So, here we aim to find the relations between their postings and their MBTI types of personality.

B. Data Preprocessing

Preprocessing in this study uses several NLTK functions. The preprocessing stage for this research has several stages, namely data cleaning (such as case folding, filtering, and stopword removal), tokenization, and lemmatization. This stage is an important component that must be carried out in the preprocessing of this dataset to achieve optimal performance.

1) Data Cleansing

Data cleansing is an important component of machine learning in terms of text mining methods and is an important technique in the pre-processing stage for raw data. At the data cleaning stage, the data that has been prepared must be completely clean of noise, inconsistency, and errors in the data set to get good accuracy when modeling. The data to be pre-processed is 8675 data containing 50 post comments from each user, and the data contains a large data set containing information that is not significant for our purposes.

a) Links were removed from this data set because they often contain meaningless information that can be viewed without exploring the content of the link (e.g., http://www.youtube.com/watch?v=qsXHcwe3krw).

b) All text is converted to lowercase to facilitate further processing.

c) The information in square brackets is omitted because it contains information that is paraphrased in already written text or nonsensical text which will only serve to obscure the data.

d) The next stage is to remove words that have no relation to the predicted value. For example, clearing data from punctuation marks like (!”#$&'()*+,-./:;<=>?@[\]^_`{|}~) which will be replaced with space characters. The elimination of punctuation marks is done because the punctuation training process will be ignored so that the training process will be simpler. Stopword is a collection of words that do not have unique features or words contained in a document. The stop words used in our study are used in English to refer to words that generally do not affect the meaning of a sentence, such as “the”, “and”, and “was”.

e) Words containing numbers are removed from the data set due to a very high number of alphanumeric sequences and Unicode characters (e.g., 'x93a', '&#x27').

2) Tokenization

Tokenization is the process of dividing the text into a meaningful set of chunks. These pieces we can call tokens. So, it separates every word that composes a separated document like whitespace characters, punctuation marks, etc.

In our research, the tokenization process uses *word_tokenize()* function. Word tokenization is the process of separating the text into words. NLTK provides many ways to get word tokens. These tokenizers can be divided by punctuation and non-alphabetic characters. This especially calls the Treebank word tokenizer so that their two outputs are identical. In this study ,88781 unique tokens were found.

3) Lemmatization

In our research, we preprocess the posts by using the Lemmatization technique. Lemmatization is the process of grouping together the different inflected forms of a word so they can be analyzed as a single item. The process of lemmatization uses *PorterStemmer()* and *WordNetLemmatizer()*.

Stemmer porter is commonly used stemmer because of its good results. Lemmatization is similar with stemming, but it brings a context to the words, hence we use this instead in our model. So, it links words with similar meanings to one word. The lemmatization method takes two arguments, one is the word for lemmatization and the second is the word for part of speech (POS).

C. Imbalance Data

Several different techniques were used for dealing with unbalanced datasets. The naivest class of techniques is a sampling: converting the data presented to the model by undersampling the general class, oversampling (duplicates) the rare class, or both. A group of researchers applied a comprehensive range of modern data sampling techniques with the imbalance-learning contribution module to *sklearn*. This submodule is installed as part of the basic *sklearn* installation by default, meanwhile *imblearn* implements over-sampling and under-sampling using custom classes.

After going through the above processes and the data is considered clean, the next step is to separate the dataset into two, namely training data and random test data. However, because the MBTI personality dataset is unstable, the research uses a sampling technique to balance the dataset with the random oversampling from the Imblearn library. For details on the separation of training data and test data, see Table 1.

Table 1 is the distribution of the dataset before and after random oversampling using *train_test_split* division with a ratio of 80% train data and 20% test data. The dataset before random oversampling had a train data of 7808 data and a test data of 867 data, whereas after the random over-sampling technique the number of train data became 11100 data and a test data of 2776 data.

D. Long Short-Term Memory (LSTM) Modeling

LSTM was first introduced in 1997 by Hochreiter and Schmidhuber. LSTM is a type of Recurrent Neural Network (RNN). LSTM itself can find the hidden layer of each cell and is designed to store previous cell information. The LSTM method is used by classifying long-term data by storing it in memory cells. Until this research has been carried out by many researchers in order to develop the LSTM method, the LSTM method itself had four main components, namely: Input Gate, repeated connections, forget gate, and output gate [11].

This allows the model to remember information for long periods and consequently understand the context better. These features are ideal for NLP problems such as this one since the context of words in a sentence and sentences in a paragraph are important.

Network architecture is formed to produce optimal accuracy. In general, the training model can have a various number of layers, but this study consists of four layers namely the Embedding layer, LSTM layer, one Dense layer with various input features. The overall dataset is divided into training data and test data. In addition, optimization function uses 'Adadelta', 'Adam', 'RMSprop', 'SGD' and the learning rate uses 10^{-2}, 10^{-3}, 10^{-4}, and categorical cross-entropy loss are used for the optimizer and loss functions, respectively.

For softmax activation and ReLU function is used. The step for making the LSTM model is shown in Figure 1, which is used in this study as follows.

TABLE 1
SPLIT DATASET BEFORE AND AFTER RANDOM OVERSAMPLING

Dataset Before Random Over Sampling		Dataset After Random Over Sampling	
Data Train	Data Test	Data Train	Data Test
7808	867	11100	2776

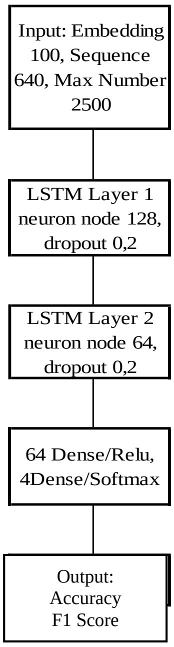
Input: Embedding 100, Sequence 640, Max Number 2500
LSTM Layer 1 neuron node 128, dropout 0,2
LSTM Layer 2 neuron node 64, dropout 0,2
64 Dense/Relu, 4Dense/Softmax
Output: Accuracy F1 Score

Figure 1. Illustration of LSTM Architecture with 2 Hidden Layers

1. The first layer created is the Embedding Layer which uses a vector with a length of 100 to represent each word.
2. The next layer is two layers LSTM.
   a. Layer 1 with 128 units of neurons and dropout 0.2, recurrent dropout 0.2, return sequences (‘True’).
   b. Layer 2 with 64 units of neurons and dropout 0.2.
   c. Dense with 64 units, and activation of ReLU function.
3. The last layer is the Dense Output Layer with 4 neuron units and the activation function uses softmax. Because the research is a binary classification, the loss function is *categorical_crossentropy* because more 4 class (Keras) is used and the optimization function uses 'Adadelta', 'Adam', 'RMSprop', 'SGD' and the learning rate uses 10^{-2}, 10^{-3}, 10^{-4}. The batch size used is 128 with 10 epochs, and the model evaluation parameter is 'accuracy'.

Constraints that often occur in the LSTM architectural model are overfitting model conditions where the accuracy and loss values during training are different with the one during validation. Performance on training data will always seem to increase but at some point, even with the same number of epochs, there is a decrease in performance during validation. To get the performance of the model, an effort will be made by tuning the hyperparameters in order to produce a good fitting performance. In this study, we will use the Dropout layer. This Dropout layer is used to reduce overfitting in the LSTM model.

The test data that had previously been carried out in the preprocessing stage, were then used as a test for the model that had been trained. This study uses Keras API in the training and validation process using the 'accuracy' metric to calculate the level of model accuracy on each training data and test data. To calculate the accuracy, we use (1).

$$Accuracy = \left(\frac{TP+TN}{TP+FP+TN+FN}\right) \times 100\% \qquad (1)$$

where $TP+TN$ is the number of correct predictions and $TP+FP+TN+FN$ is the total number of predictions made.

E. Classification Methods

In addition to using LSTM as a predictive model, this study will also train several models by classifying different algorithms such as Extreme Gradient Boosting (XGB), Logistic Regression (LR), Stochastic Gradient Descent (SGD), Random Forest (RF), Support Vector Machine (SVM), and K-nearest Neighbors (KNN) for 4 dimensions of the target. The six models will be analyzed with the value of accuracy as the output value. The model in this research will be implemented by Python Library Scikit-learn with Google Collab as its tool.

III. Results

A. Data Description

In this study, the dataset used had 8675 data and 2 attributes (type and posts). The types in this dataset had 16 MBTI personality types. In addition to using 16 personality types, this study also chose to create 4 class dimensions, one for each category. And in this study, the data would be divided into 80% training data and 20% test data. In addition, there was data that was used as data validation by 10%. Distribution of the number of posts in the 4-dimensional class is as follows.

Introversion (I) - Extroversion (E)	: 1999 / 6676
Intuition (N) - Sensing (S)	: 1197 / 7478
Thinking (T) - Feeling (F)	: 4694 / 3981
Judging (J) - Perceiving (P)	: 5241 / 3434

Figure 2 shows the representation of the imbalanced data in the MBTI dataset.

B. Long Short-Term Memory

In the LSTM model used in this study with and without random oversampling, here are some hyperparameter configurations for testing using three learning rate variants (10^{-2}, 10^{-3}, 10^{-4}) and four optimizer variants (Adadelta, Adam, RMSprop, SGD) with batch size 128, epoch 10, max sequence length 2500, input vector length 640, embedding 100, dropout 0.2, LSTM Unit 128, and 64 neuron nodes and Softmax activation.

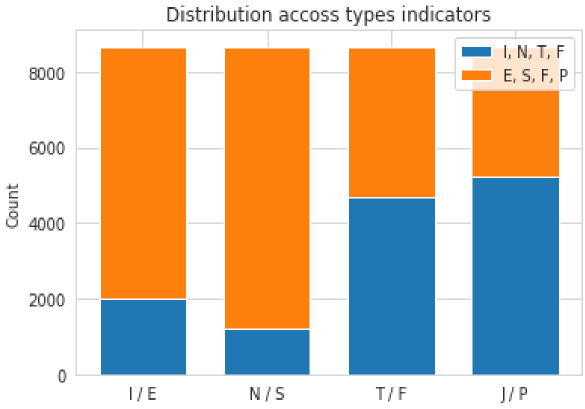

Figure 2. Unbalanced representation of MBTI types in the dataset

According to the hyperparameters that have been described, the test was carried out using the LSTM model without random oversampling and the results can be seen in Table 2 for accuracy results and Table 3 for results from F1-Score.

From Table 2, it can be concluded that the LSTM model with accuracy results without random oversampling using four optimizer variants and three learning rates gives the highest results on the RMSprop optimizers and a learning rate of 10^{-4} with an accuracy value of 42.38%.

Based on the result in Table 3, can be concluded that the LSTM model with F1-Score results without random oversampling using four optimizer variants and three learning rates gives the highest results on the RMSprop optimizers and a learning rate of 10^{-4} with an F1-Score value of 39.33%.

The test was carried out using the LSTM model with random oversampling based on the hyperparameters that have been described and the results shows in Table 4 for accuracy and Table 5 for the results from F1-Score.

Based on the result in Table 4, can be concluded that the LSTM model with accuracy results with random oversampling using four optimizer variants and three learning rates, gives the highest results on the RMSprop optimizers and a learning rate of 10^{-3} with an accuracy value of 86.31%.

From the Table 5, it can be concluded that the LSTM model with F1-Score results with random oversampling using four optimizer variants and three learning rates gives the highest results on the RMSprop optimizers and a learning rate of 10^{-2} with an F1-Score value of 86.08%.

Based on the results of trials using the LSTM model with the optimizers Adadelta, Adam, SGD, RMSprop, and Learning Rates of 10^{-2}, 10^{-3}, 10^{-4} with and without random oversampling, it gives the highest accuracy, namely the LSTM model with random oversampling using Optimizer RMSprop and Learning Rate of 10^{-3}. Confusion Matrix and historical data for accuracy models and loss models can be seen in Figure 3, Figure 4 and Figure 5.

Figure 3 shows the Confusion Matrix generated from the random oversampling technique from the performance evaluation is carried out on the application of the LSTM method with the ROS technique with RMSprop optimizers and a learning rate of 10^{-3}.

TABLE 2
ACCURACY RESULTS FROM LSTM MODELS WITHOUT RANDOM OVERSAMPLING

Learning Rate	**Accuracy (%)**			
	Adadelta	**Adam**	**RMSprop**	**SGD**
10^{-2}	31.37	20.23	22.28	14.34
10^{-3}	25.03	33.80	35.72	25.16
10^{-4}	27.52	34.57	**42.38**	29.32

TABLE 3
F1-SCORE RESULTS FROM LSTM MODELS WITHOUT RANDOM OVERSAMPLING

Learning Rate	**Accuracy (%)**			
	Adadelta	**Adam**	**RMSprop**	**SGD**
10^{-2}	31.03	6.94	8.12	7.66
10^{-3}	18.76	22.21	24.00	15.40
10^{-4}	28.40	30.18	**39.33**	26.50

TABLE 4
ACCURACY RESULTS FROM LSTM MODELS WITH RANDOM OVERSAMPLING

Learning Rate	Accuracy (%)			
	Adadelta	Adam	RMSprop	SGD
10^{-2}	68.37	83.07	82.28	81.84
10^{-3}	31.88	84.40	**86.31**	61.85
10^{-4}	25.65	83.18	83.50	32.71

TABLE 5
F1-SCORE RESULTS FROM LSTM MODELS WITH RANDOM OVERSAMPLING

Learning Rate	F1-Score (%)			
	Adadelta	Adam	RMSprop	SGD
10^{-2}	43.71	67.01	**86.08**	80.59
10^{-3}	83.36	33.18	85.72	58.89
10^{-4}	84.58	25.01	82.90	29.85

The accuracy value can also be calculated manually by showing a comparison between the total number of correct diagnoses/predictions *(TP + TN)* and the total number of diagnoses/predictions *(TP+TN+FP+FN)*. The value of total number of correct diagnoses is the sum of 491, 653, 615, and 637, while the value of the total number of diagnoses/predictions is 2776. The result of manual calculation of the confusion matrix using (1) is 86.31%.

Based on the results of manual calculations from the confusion matrix with the accuracy values obtained equal to the results of evaluation and testing, it is proven that Based on the accuracy values obtained from the Myers-Briggs Type Indicator (MBTI) dataset, the LSTM is a good model for building MBTI personality prediction models.

Figure 4 and Figure 5 are historical data from the accuracy model and loss model of the LSTM model with RMSprop optimizer and learning rate of 10^{-3} respectively. In the accuracy model, we can see that the train and test accuracy have significant accuracy even though there is a decrease in several epochs when running the test execution. However, the resulting accuracy value is still significantly high.

C. Classification Models

The classification model uses the Extreme Gradient Boosting (XGB), Logistic Regression (LR), Stochastic Gradient Descent (SGD), Random Forest (RF), Support Vector Machine (SVM), K-nearest Neighbors (KNN), and with the same preprocessing data settings and the same distribution of training data and test data, which is 80% for data training and 20% for test data.

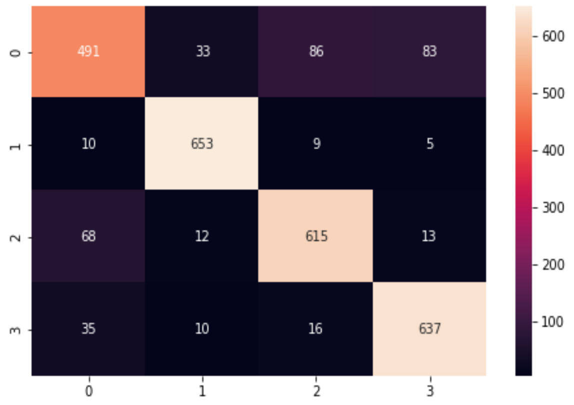

Figure 3. Confusion Matrix Model LSTM Optimizers RMSprop and Learning Rate 10^{-3}.

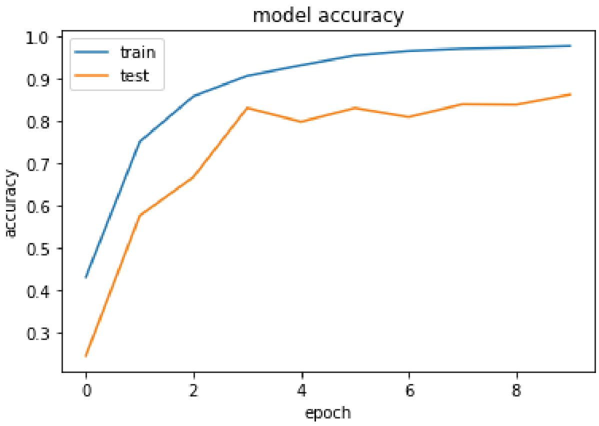

Figure 4. Data History of Model Accuracy Model LSTM Optimizers RMSprop and learning rate of 10^{-3}.

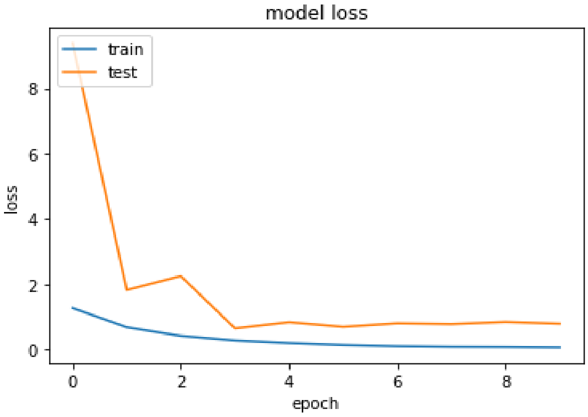

Figure 5. Data History of Model Loss Model LSTM Optimizers RMSprop and learning rate of 10^{-3}.

Some of the classification methods used in this experiment have somewhat similar performance when the parameters are optimized, as shown in Table 6.

From Table 6, the results of the classification model research show that of all research models that have been evaluated based on 4 dimensions of personality type, the SVM model provides relatively high performance of 75.71.

Based on Table 7, the results of the classification model research show that of all research models that have been evaluated based on 4 dimensions of personality type, the Logistic Regression (LR) model provides relatively high performance of 70.99.

TABLE 6
ACCURACY RESULT FROM THE CLASSIFICATION MODEL WITHOUT RANDOM OVERSAMPLING

Classifier	Accuracy (%)				
	I/E	N/S	F/T	J/P	Average
XGB	77.46	86.40	68.93	65.36	74.54
LR	76.25	86.34	72.33	64.73	74.91
SGD	77.23	86.34	72.51	64.78	75.22
RF	77.12	86.28	67.32	62.82	73.39
SVM	**77.35**	**86.34**	**72.85**	**66.28**	**75.71**
KNN	64.73	64.73	64.73	64.73	64.73

TABLE 7
F1-SCORE RESULT FROM THE CLASSIFICATION MODEL WITHOUT RANDOM OVERSAMPLING

Classifier	F1-Score (%)				
	I/E	N/S	F/T	J/P	Average
XGB	68.47	80.15	68.49	60.50	69.40
LR	**69.52**	**80.23**	**72.22**	**61.98**	**70.99**
SGD	67.73	80.01	72.61	63.58	70.98
RF	67.81	80.12	64.97	58.36	67.82
SVM	67.79	80.01	72.69	62.90	70.85
KNN	67.70	80.11	39.51	50.69	59.50

From Table 8, the results of the classification model research show that of all research models that have been evaluated based on 4 dimensions of personality type, the Random Forest (RF) model provides relatively high performance of 84.78.

Table 9 shows the results of the classification model research show that of all research models that have been evaluated based on 4 dimensions of personality type, the Random Forest (RF) model provides relatively high performance of 84.98.

According to the test results obtained from the classification model with and without random oversampling which can be seen in Table 6, Table 7, Table 8, and Table 9, the category of personality type N/S dominates higher than other personality type categories by having the highest accuracy values and F1-Score for each category of personality type, it can be concluded that the category of personality type N/S has a high influence.

IV. DISCUSSIONS

In the research that we have tested, the results with the highest accuracy value from the LSTM model with random oversampling are 86.31% accuracy and the F1-score value is 86.08%. While this appears to indicate a weak overall ability of our model to correctly classify all four dimensions of MBTI, it does not indicate the effectiveness of our model for achieving predictive estimates of the overall MBTI type.

TABLE 8
ACCURACY RESULT FROM THE CLASSIFICATION MODEL WITH RANDOM OVERSAMPLING

Classifier	Accuracy (%)				
	I/E	N/S	F/T	J/P	Average
XGB	71.92	77.27	70.50	65.09	71.20
LR	76.25	86.34	72.33	64.73	74.91
SGD	66.01	70.59	71.03	62.18	67.45
RF	**94.95**	**98.93**	**71.19**	**74.06**	**84.78**
SVM	87.38	97.59	74.17	71.91	82.76
KNN	93.79	96.82	53.67	76.16	80.11

TABLE 9
F1-SCORE RESULT FROM THE CLASSIFICATION MODEL WITH RANDOM OVERSAMPLING

Classifier	F1-Score (%)				
	I/E	N/S	F/T	J/P	Average
XGB	71.82	77.25	70.51	65.05	71.16
LR	66.69	69.36	71.31	63.14	67.63
SGD	66.55	68.55	71.04	62.49	67.16
RF	**94.50**	**98.86**	**73.06**	**73.48**	**84.98**
SVM	87.38	97.59	74.18	71.91	82.77
KNN	93.78	96.83	40.47	75.19	76.57

Other models that focus on MBTI's multi-class classification can achieve higher perfect classification accuracy, but they do so with the risk of misprediction. That is, multi-class classifications treat all classes as independent of each other, so they fail to capture relatedness in constructs from one type to another (e.g., INFPs are much more similar to INTJs than ESTJs). Nonetheless, our model represents a trade-off of these two aspects: we achieve a lower level of perfect classification in exchange for a higher value of roughly correct classification (i.e., a "good" classification).

Based on Table 10, the comparison results in personality prediction research shows that from all research models that have been evaluated and tested, the Long Short-Term Memory (LSTM) model provides relatively good performance with an accuracy value of 86.31% which is higher than the accuracy value supporting models that have been tested. Based on the accuracy values obtained from the Myers-Briggs Type Indicator (MBTI) dataset, the LSTM model is good for building the MBTI personality prediction model. The comparison table of previous research results with this study that can be seen in Table 11.

We can see that Table 11 is a previous study using the same model, namely LSTM and we can conclude that our research provides a significant contribution value and has succeeded in producing high accuracy and F1-Score values so that the MBTI dataset using the LSTM model with random oversampling can predict MBTI personality. In addition to the problem of unbalanced data, knowing a personality is important for a person because it reflects the person's behavior and is as an attitude in social relations. It helps people identify their true strengths and weaknesses. This is of course very important and beneficial for themselves and others. That way, one can find the right way to correct these shortcomings and develop their potentials.

TABLE 10
COMPARISON MODEL RESULTS

Model	Without ROS		With ROS	
	Accuracy (%)	F1-Score (%)	Accuracy (%)	F1-Score (%)
XGB	74.54	69.40	71.20	71.16
LR	74.91	70.99	74.91	67.63
SGD	75.22	70.98	67.45	67.16
RF	73.39	67.82	84.78	84.98
SVM	75.71	70.85	82.76	82.77
KNN	64.73	59.50	80.11	76.57
LSTM	**42.38**	**39.33**	**86.31**	**86.08**

TABLE 11
COMPARISON OF PREVIOUS RESEARCH RESULTS

No	Researcher	Method	Metric	Result (%)
1	Cui & Qi (2017)	LSTM	*Accuracy*	79.03
2	Hernandez and Knight (2017)	LSTM	*Accuracy*	67.78
3	Keh & Cheng (2019)	BERT	*Accuracy*	74.48
4	Mehta et al. (2020)	BERT + MLP	*Accuracy*	77.10
5	Frković et al. (2020)	LSTM	*F1-Score*	71.60
6	**Proposed Method**	**LSTM With ROS**	***Accuracy***	**86,31**
			F1-Score	**86,08**

CONCLUSION

Based on the results of the research conducted, it can be concluded that data mining can help in predicting the accuracy of the MBTI personality. This study aims to create a model to predict the MBTI personality type. The dataset used is a public dataset from the Kaggle online repository site.

The modeling used in this study is Long Short-Term Memory (LSTM) using four Optimizer variants (Adadelta, Adam, RMSprop, SGD) and three Learning Rate variants (10^{-2}, 10^{-3}, 10^{-4}) assisted without and with Random Over Sampling, so that the data is balanced. LSTM modeling using RMSprop Optimizer and Learning Rate of 10^{-3} with random oversampling is the LSTM model with high performance by providing an accuracy value of 86.31% while for F1-score values, LSTM model using RMSprop Optimizer and Learning Rate of 10^{-2} provides high performance with an F1-score of 86.08%. Based on the accuracy values and F1-score obtained from the Myers-Briggs Type Indicator (MBTI) dataset, the LSTM model is good for building the MBTI personality prediction model.

The Random Over Sampler (ROS) technique used has made a great contribution and is successful because it can produce a fairly high accuracy value for the MBTI personality dataset.

There are still some problems that need to be solved in this area. Text is a very challenging task because natural languages are more ambiguous and difficult to handle. Implementing more data, natural language processing methods, or newer text features is in our plans for the future.

REFERENCES

[1] M. J, *(MBTI) Myers-Briggs Personality Type Dataset,* 2017, Kaggle, Mar 2021. [Online]. Available: https://www.kaggle.com/datasnaek/mbti-type

[2] M. H. Amirhosseini and H. Kazemian, "Machine Learning Approach to Personality Type Prediction Based on the Myers–Briggs Type Indicator®," *Multimodal Technol. Interact.,* vol. 4, no. 1, pp. 1-15, 2020.

[3] N. H. Z. Abidin, M. A. Remli, N. M. Ali, D. N. E. Phon, N. Yusoff, H. K. Adli and A. H. Busalim, "Improving Intelligent Personality Prediction using Muyers-Briggs Type Indocator and Random Forest Classifier," *International Journal of Advanced Computer Science and Applications,* vol. 11, no. 11, pp. 192-199, 2020.

[4] T. A. Tutupoly and I. Alfarobi, "Komparasi algoritma C4.5 dan naive bayes yang dikembangkan menjadi web intellegence pada perhitungan bonus tahunan karyawan di PT. ABC," *Jurnal Mitra Pendidikan (JMP Online),* vol. 3, no. 1, pp. 92-103, 2019.

[5] Y. Mehta, S. Fatehi, A. Kazameini, C. Stachl, E. Cambria and S. Eetemadi, "Bottom-up and top-down: predicting personality with psycholinguistic and language model features," in *2020 IEEE International Conference on Data Mining (ICDM),* Sorrento, 2020.

[6] S. Bharadwaj, S. Sridhar, R. Choudhary and R. Srinath, "Persona traits identification based on Myers-Briggs Type Indicator (MBTI) - a text classification approach," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI),* Bangalore, 2018.

[7] C. Li. et al., "Feature extraction from social media posts for psychometric typing of participants," in *Lecture Notes in Computer Science*, vol. 10915, D. D. Schmorrow dan C. M. Fidopiastis (eds) Augmented Cognition: Intelligent Technologies. AC 2018, New York, Springer, Cham, 2018, pp. 267-286.

[8] S. S. Keh and I.-T. Cheng, "Myers-briggs personality classification and personality-specific language generation using pre-trained language models," 15 July 2019. [Online]. Available: https://arxiv.org/abs/1907.06333.

[9] E. J. Choong and K. D. Varathan, "Predicting judging-perceiving of myers-briggs type indicator (mbti) in online social forum," *PeerJ 9:e11382,* pp. 1-27, 2021.

[10] S. Patel, M. Nimje, A. Shetty and S. Kulkarni, "Personality analysis using social media," *International Journal of Engineering Research & Technology (IJERT),* vol. 09, no. 03, pp. 306-309, 2020.

[11] M. Frković, N. Čerkez, B. Vrdoljak and S. Skansi, "Evaluation of Structural Hyperparameters for Text Classification with LSTM Networks," in *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO),* Opatija, 2020.

[12] R. Hernandez and I. S. Knight, "Predicting Myres-Briggs Type Indicator with text classification," Curran Associates Inc., New York, 2017.

[13] B. Cui and C. Qi, "Survey analysis of machine learning methods for natural language processing for MBTI personality type prediction," 2017.