Temperament detection based on twitter data: classical machine learning versus deep learning

Annisa Ulizulfa [a,1], Retno Kusumaningrum [a,2,*], Khadijah Khadijah [a,3], Rismiyati Rismiyati [a,4]

[a] Department of Informatics, Universitas Diponegoro, Semarang, Indonesia
[1] anisaulizulfa@students.undip.ac.id, [2] retno@live.undip.ac.id, [3] khadijah@live.undip.ac.id, [4] rismiyati@live.undip.ac.id

* corresponding author

ARTICLE INFO

Selected paper from The 2020 3rd International Symposium on Advanced Intelligent Informatics (SAIN'20), Nanjing-China (Virtually), 25-26 November 2020, http://sain.ijain.org/2020/. Peer-reviewed by SAIN'20 Scientific Committee and Editorial Team of IJAIN journal.

Article history
Received November 16, 2020
Revised July 17, 2021
Accepted November 17, 2021
Available online March 31, 2022

Keywords
Temperament Detection
Twitter User
Support Vector Machine
K-Nearest Neighbour
Convolutional Neural Network

ABSTRACT

Deep learning has shown promising results in various text-based classification tasks. However, deep learning performance is affected by the number of data, i.e., when the number of data is small, deep learning algorithms do not perform well, and vice versa. Classical machine learning algorithms commonly work well for a few data, and their performance reaches an optimal value and does not increase with the increase in sample data. Therefore, this study aimed to compare the performance of classical machine learning and deep learning methods to detect temperament based on Indonesian Twitter. In this study, the proposed Indonesian Linguistic Inquiry and Word Count were employed to analyze the context of Twitter. The classical machine learning methods implemented were support vector machine and K-nearest neighbor, whereas the deep learning method employed was a convolutional neural network (CNN) with three different architectures. Both learning methods were implemented using multiclass classification and one versus all (OVA) multiclass classification. The highest average f-measure was 58.73%, obtained by CNN OVA with a pool size of 3, a dropout value of 0.7, and a learning rate value of 0.0007.

1. Introduction

The personality of a person has two components: temperament and character [1]. Temperament reflects propensity, while character reflects the configuration of habits. Furthermore, temperament is an innate human trait, while character arises because of the interaction between temperament and the environment. Detecting the temperament of a person can be a useful tool for detecting personality traits, which not only describe individual's behaviour, but also include ways of thinking and feelings that affect motivations, preferences, emotions, and even health [2]. The study of a person's personality traits is importance in psychology and personality recognition [3]. It can also benefit many other applications, such as depression and mental illness detection [4], cyberbullying detection [5], sentiment analysis [6], help match people to ideal job [7], and so on. The most performed temperament detection is the filling out of a questionnaire test. However, this leads to error-prone results because people might not seriously answer the questions or might give the question a random answer; hence, the answer does not reflect her or his temperament. One possible solution to this problem is based on social media data. Social media use a personal and private account, where each social media user can express herself or himself to the virtual world. Doing so reflects the person's personal life and daily life [8], which expresses the temperament of the person. The activities of users on their social media, such as Twitter, are similar to user activities when they interact in the real world. Twitter is often used as a medium to publish their activities via short messages (tweets) which can be in the form of text, photos, or videos [9]. This shows that activity data on social media users can describe their behaviour, opinions, and interests [10], [11]. Behaviour is an element of personality. Therefore, tracing the digital footprint on social media has become an effective solution for identifying temperament.

There are several studies of predicting temperament based on social media data, such as predicting temperament and mood using case-based reasoning [12] and predicting temperament using machine learning methods [10], [13]–[15]. However, in these studies, the use of classical machine learning is generally proposed, such as K-nearest neighbour (KNN) [10], [13], AdaBoost [14], bagging [14], decision tree C4.5 [10], [14], naïve Bayes [10], [14], random forest [13], [14], support vector machine (SVM) [13]–[15], and multilayer perceptron [10]. The main drawback of classical machine learning is the need to craft the representative features manually for each method to obtain the optimal performance. Deep learning, however, has shown promising results in various text-based classification tasks in which high-level features are automatically learned from data. However, the performance of deep learning is affected by the number of data, i.e., when the data are small, deep learning algorithms do not perform well, and vice versa. Classical machine learning algorithms commonly work well for a small number of data, and their performance reaches an optimal value and does not increase with the increase in sample data. Therefore, the aim of this study was to compare the performances of classical machine learning and deep learning to detect temperament based on Indonesian Twitter. Two classical machine learning algorithms, i.e., SVM and KNN, were implemented as baseline algorithms. This is because both algorithms provide better results than random forest in the study of multiclass classification [13]. Meanwhile, a convolutional neural network (CNN) was selected as a proposed deep learning algorithm in this study because the deep learning algorithm provided better performance than shallow learning in a study on depression detection [16] and situations understanding based on sentiment analysis [17] using Twitter data.

In this study, information from Twitter data was divided into two categories: behavioural and grammatical. These two categories produced a corpus of meta-attributes or meta-bases. The grammatical category is based on information from the Linguistic Inquiry and Word Count (LIWC) dictionary, which is only available in English, Chinese, Arabic, Spanish, Dutch, French, German, Italian, Russian, and Turkish [18]. The dictionary target in LIWC is approximately 90 variables related to linguistics, where each category is related to psychological aspects. Because Indonesian language data were employed in this study, an Indonesian LIWC dictionary was also proposed independently based on the categories provided in the LIWC2015 application. In addition, Keirsey's temperament model was implemented, including the guardian, artisan, idealist, and rational types. This model is widely accepted for the understanding of professional trends; thus, it is potentially applicable in recruitment and selection processes, which are promising areas for social media data analysis [14]. Therefore, the task in this study is multiclass classification. However, multiclass classification problems are generally more difficult to solve than binary classification problems. Binarization techniques appear to solve multiclass problems by dividing the problems into easier forms in the form of binary classification. The binarization technique that is commonly used is one versus all (OVA) [19]. Based on this explanation, the performances of classical machine learning (SVM and KNN) and deep learning (CNN) were compared for temperament detection of Indonesian Twitter users. The performances of both models were also studied in multiclass and OVA multiclass classification. In addition, this research utilised categories in LIWC to build an Indonesian LIWC dictionary for feature extraction.

2. Method

The general process used in this research is illustrated in Fig. 1, which consists of four main processes: (i) creation of the Indonesian LIWC, (ii) construction of the dataset, (iii) model generation for temperament detection, and (iv) real-time processing of temperament detection for Twitter users. The first three processes are batch processing, whereas the last process is real-time processing. Each process consists of several subprocesses as inputs to and outputs from other related processes.

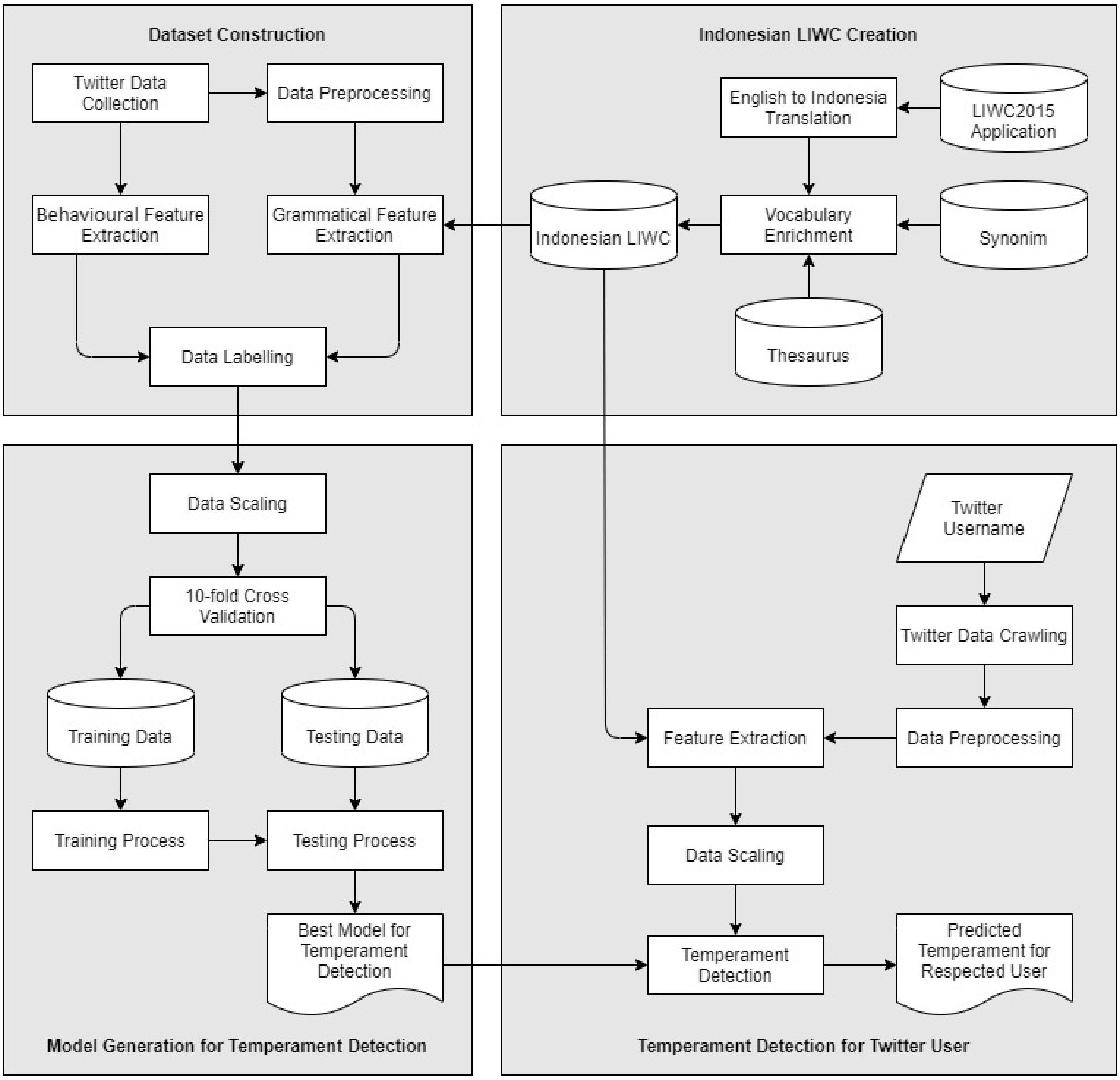

Fig. 1. Research methodology

2.1. Creation of Indonesian LIWC

The choice of language and words used daily by everyone is a medium to translate their thoughts and emotions into a form that can be understood by other individuals. This choice of language and words that are spoken or written is used by cognitive, personality, clinical, and social psychologists as material to try to understand humans [20]. LIWC was proposed in 1993 and continued to be developed in 2001, 2007, and 2015. In this study, LIWC2015 was employed as a reference. The LIWC2015 is one of the most versatile, easy-to-master instruments for converting any text into data, and helping psychologists who would normally not be good at data science [21]. Based on LIWC2015, only 68 categories out of approximately 90 categories were used because there are only 68 categories that have word examples in the LIWC2015 documentation. The first subprocess is a translation process from English to Indonesian for the collected words. Furthermore, the second subprocess is used to enrich vocabulary using two approaches: (i) an approach based on word synonyms (http://sinonimkata.com) and (ii) one based on the thesaurus (http://tesaurus.kemdikbud.go.id/tematis). This process is expected to increase the number of word samples for each category.

2.2. Dataset Construction

The first subprocess of this process is data collection. Data were collected by searching the usernames of Twitter users who were grouped based on their type of temperament; then, the crawling process was conducted to obtain information about those usernames. The information taken from each username was the latest 200 tweets, number of followers, following, favourites, and user status. Then, the crawled data were labelled on each username based on the type of temperament of the user. The collected data consisted of four classes of temperament, i.e., guardian, artisan, idealist and rational, where each class contained 200 data. The next subprocess is data pre-processing. Pre-processing is an important stage in research using text domain data. This is because raw text data have an unstructured format, redundancies, and inconsistencies [22]. Therefore, pre-processing is required to make the data clean or make them have the same format. There are several steps in pre-processing, as follows.

- Deletion of Mention, Hashtag, Retweet, and URL; Deleting the mention, hashtags, retweets, and URLs is done because most mentions, hashtags, retweets, and URLs do not have a correlation with the meaning of a tweet and are also not used in the next process [23].
- Translating Expression Symbol; A tweet usually not only contains words but also symbols, commonly called 'emoticons' and 'emojis', which clarify the emotional description of a tweet. The process of translating expression symbols is performed to translate symbols into words that can be processed at a later stage. In this study, the symbols contained in the tweet were translated into five classes of emotions: anger, sadness, joy, fear, and surprise [24].
- Case Folding; Case folding involves changing words in text into uniform lowercase letters to facilitate further processing [23]. In this research, folding was carried out for all words in the dataset so that the same word, but with a different letter format, is not detected as two different words.
- Elimination of Special Characters; The elimination of special characters is the process of removing all characters except for letters and hyphens '-' in the text [23]. This process is carried out because the characters to be processed at a later stage are only letters and hyphens '-', while the other characters have no meaning.
- Formalisation; Formalisation is changing the abbreviation in the text into its respective whole word representation, or a word that does not contain an abbreviation [23]. In this research, formalisation was carried out by creating a dictionary of abbreviations containing abbreviated words and whole words. Abbreviations contained in the text and dictionary abbreviations are formalised so that words can be processed in the next process.
- Stemming; Stemming is one of the processes in pre-processing. The purpose is to map word variations to their basic word forms. The stemming process is performed by removing the affix (beginning and ending) of a word to obtain the basic word [25].

The results of pre-processing in this study are clean datasets that are ready for use at the feature extraction stage. The third subprocess is behavioural feature extraction, which was implemented for the collected raw Twitter data. The number of followers, following, and favourites crawled from Twitter users were used as behavioural features. A user tweet contains up to 280 characters, including mentions, hashtags, retweets, and URLs. Therefore, the number of mentions, hashtags, retweets, and URLs from each tweet were extracted as the other behavioural features. The result of behavioural feature extraction from each Twitter user is a feature set represented as a vector of eight elements consisting of the number of followers, number of following, number of favourites, status, number of mentions, hashtags, retweets, and URLs. These behavioural features were extracted from each of the Twitter users collected. In addition, another kind of feature was employed — grammatical features. Therefore, the next subprocess is grammatical feature extraction, which uses the results of pre-processing as an input of this subprocess. Grammatical feature extraction extracts grammatical information based on the categories in Indonesian LIWC. Indonesian LIWC was used as a dictionary to match each word in a tweet with a word in a certain category. The number of tweet words in each category of Indonesian LIWC was recorded and used as a grammatical feature. Because the number of categories in Indonesian LIWC is 68, the result of the grammatical feature extraction is a feature set represented as a vector of 68 elements, where each element represents the number of tweet words in a category of Indonesian LIWC.

The last subprocess is data labelling. The behavioural and grammatical features were combined and then used as a feature set for each Twitter user. Therefore, the dataset contained 800 Twitter users, where each user had 76 features from behavioural and grammatical feature extraction. Each of the Twitter data was labelled to be used in the training process. Two types of dataset were used for constructing multiclass and OVA multiclass classification models. The first dataset for multiclass classification consists of four class labels, i.e., guardian, artisan, idealist, and rational, with 200 data points in each class label. However, the second dataset for OVA multiclass classification consists of only two class labels, i.e., respected class and other class. In the OVA multiclass classification, 400 data were used, where 200 data points were taken from the respected class and the other 200 data points were taken randomly from the non-respected class. For example, the OVA multiclass classification for guardian class labels used 200 data from the guardian class and 200 data taken randomly from the other class, including artisan, idealist, and rational types. Therefore, there were four types of dataset for building the OVA multiclass classification model.

2.3. Model Generation for Temperament Detection

The first subprocess of model generation for temperament detection is data scaling. The datasets used in classification mostly have a variety of feature values. This means that the value of one feature might be too large, while the other might be too small. Data scaling and data normalisation are processes that have the same goal, which is to make data be in the same range [26]. The same range data will minimize bias in the neural network and speeds up the training process [27]. There are no specific rules regarding the selection of certain types of data-scaling dataset. In this study, the min-max and Z-score algorithms were selected based on study [28]. The min-max algorithm scales the data into numbers in the range of 0–1. The smallest data value is changed to 0, and the largest data value is changed to 1. The min-max algorithm is shown in Equation (1).

$$v' = \frac{v - x_{min}}{x_{max} - x_{min}} \tag{1}$$

where v' is the division result between the difference in the input data and minimum feature value x_{min} and the difference between the maximum x_{max} and minimum feature value x_{min}. Unlike the min-max algorithm, the Z-score does not change the dataset value to a certain value of x. This algorithm is a form of standardisation that is used to convert normal variants into standard score forms [29]. The formula to compute the Z-score is shown in Equation (2).

$$v' = \frac{v - \bar{x}}{\sigma_x} \tag{2}$$

where v' is the division result between the difference of input data v and average feature value $\bar{x}$ and the standard deviation of the feature, σ_x. The result of the transformation is data with an average value of 0 and a standard deviation of 1. The next subprocess is implementing 10-fold cross validation for splitting the dataset into training data and testing data. Ten folds cross validation (k = 10) is the most commonly used to evaluate and compare machine learning algorithms [30]. A 10-fold cross validation means that the dataset is split into 10 folds and the experiment is repeated 10 times, where each experiment uses one different fold as testing data and the nine remaining folds as training data. Because the number of instance data in the dataset for multiclass classification is 800, each experiment used 720 training data and 80 testing data. However, the OVA multiclass classification only used 400 data; hence, 360 data were used as training data and 40 data were used as testing data. The training and testing experiments were repeated 10 times using different folds as testing data. Then, the results from the 10 experiments were averaged to report the final results of the classification model performance. The next subprocess is the training and testing process, where the training process is used to build the temperament detection model, and the testing process is used to validate the trained model. Two approaches were utilized for the training and testing processes: multiclass classification and OVA multiclass classification. Both approaches involved implementing the SVM, KNN, and CNN algorithms. An illustration of the training process is presented in Fig. 2.

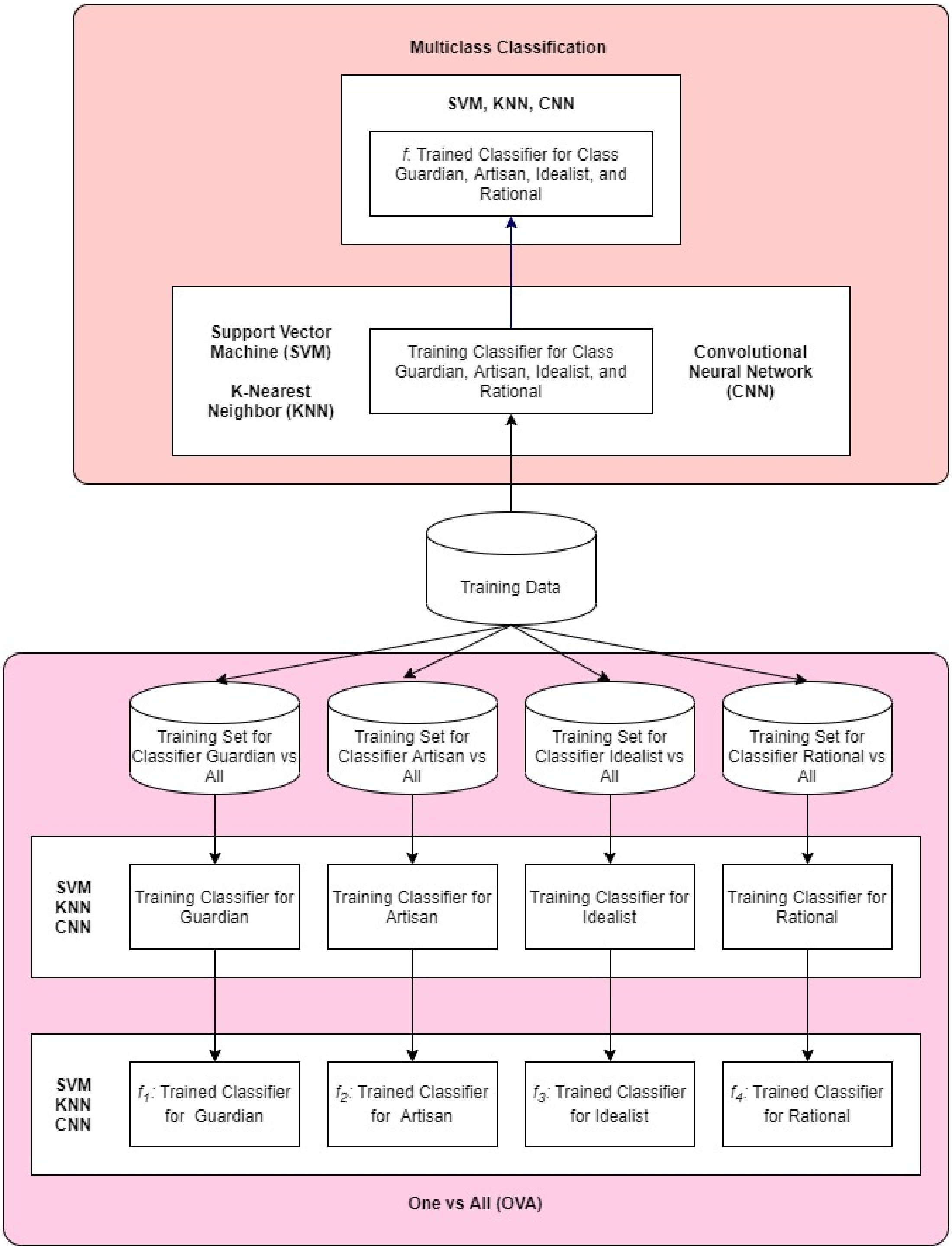

Fig. 2. Illustration of training process

Each algorithm applied for multiclass classification and OVA multiclass classification was tested using various combinations of hyperparameters to obtain the best classification model. Hyperparameter Combination for Deep Learning; As mentioned before, CNN was applied as a deep learning algorithm with three different architecture models. The first architecture adopts the architecture proposed in a previous work [16], as shown in Fig. 3. Architecture 1 uses one convolutional layer with 250 feature maps and a rectified linear unit (ReLU) activation function, global max pooling on the feature map, and softmax output activation function for multiclass classification or a sigmoid activation function for OVA multiclass classification.

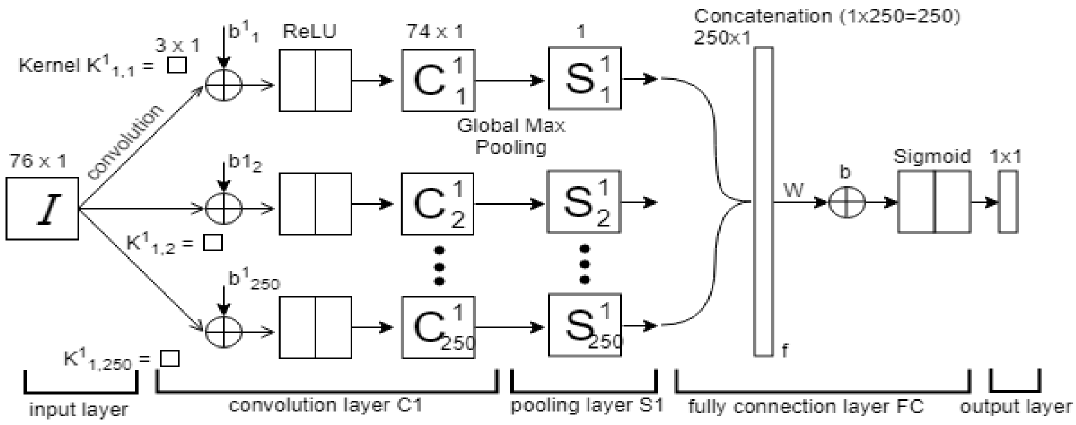

Fig. 3. CNN Architecture 1

Architectures 2 and 3 used the same output activation function as Architecture 1, but different architectures in the convolutional and pooling layers. Architecture 2 used three convolutional layers with 128 feature maps and ReLU activation functions, and one max pooling for feature map extraction, while Architecture 3 used three convolutional layers with 128 feature maps, a ReLU activation function, and two max pooling layers for feature map extraction. In addition to the first architecture, two architectures were applied for multiclass classification and OVA multiclass classification. The second and third architectures are shown in Fig. 4 and Fig. 5, respectively.

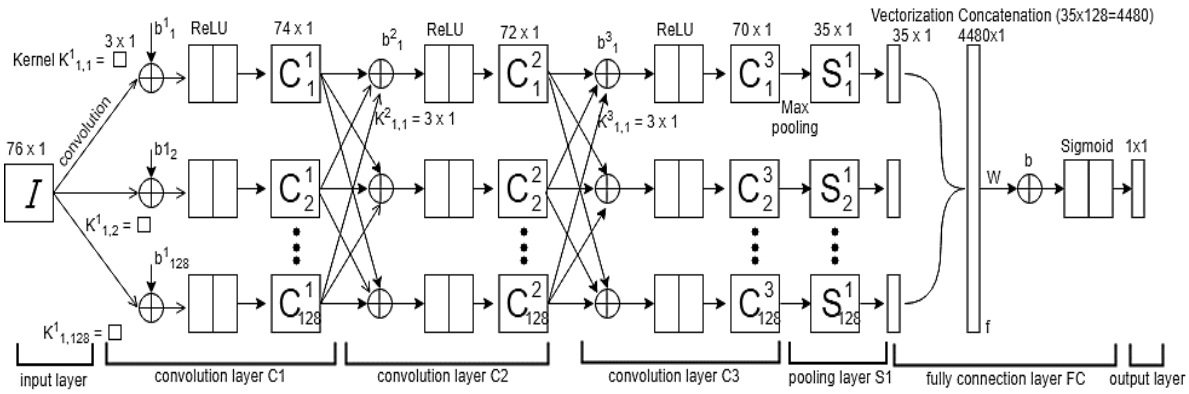

Fig. 4. CNN Architecture 2

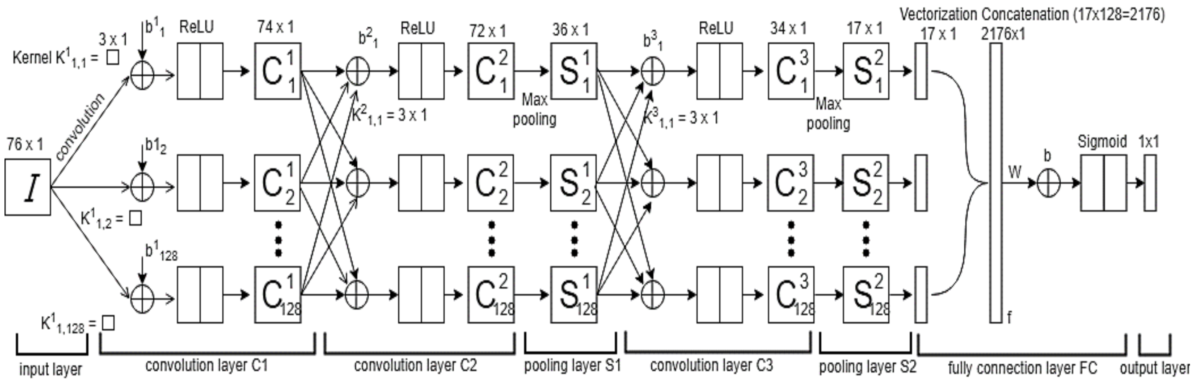

Fig. 5. CNN Architecture 3

All architectures used Adam as an optimizer. The hyperparameters for each architecture are listed in Table 1.

Table 1. Hyperparameters of CNN

Architecture 1		Architecture 2		Architecture 3	
Parameters	*Values*	*Parameters*	*Values*	*Parameters*	*Values*
Kernel	3, 5, 7	*Pool size*	2, 3, 5	*Pool size*	2.2, 3.3, 5.5
Dropout	0.3, 0.5, 0.7	Dropout	0.3, 0.5, 0.7	Dropout	0.3, 0.5, 0.7
Learning rate	0.007, 0.00007, 0.0007	*Learning rate*	0.007, 0.00007, 0.0007	*Learning rate*	0.007, 0.00007, 0.0007
Feature Map	250	*Feature Map*	128	*Feature Map*	128
Convolutional activation function	ReLU	Convolutional activation function	ReLU	Convolutional activation function	ReLU
Output activation function	*Softmax; sigmoid*	Output activation function	*Softmax; sigmoid*	Output activation function	*Softmax; sigmoid*
Type of pooling	*Global max pooling*	Type of pooling	*Max pooling*	Type of pooling	*Max pooling*

Hyperparameter Combination for Classical Machine Learning; As mentioned before, two classification algorithms were applied: KNN and SVM. Both models were trained twice — once for multiclass classification and once for OVA multiclass. The hyperparameters of each algorithm are listed in Table 2.

Table 2. Hyperparameters of SVM and KNN

SVM		KNN	
Parameters	*Values of Parameter*	*Parameters*	*Values of Parameter*
Scaling	Min-max, Z-score	Scaling	Min-max, Z-score
Kernel	linear, rbf, poly, sigmoid	K	1–10

Furthermore, the testing process was performed to evaluate all trained models. An illustration of the testing process for both multiclass classification and OVA multiclass classification is shown in Fig. 6.

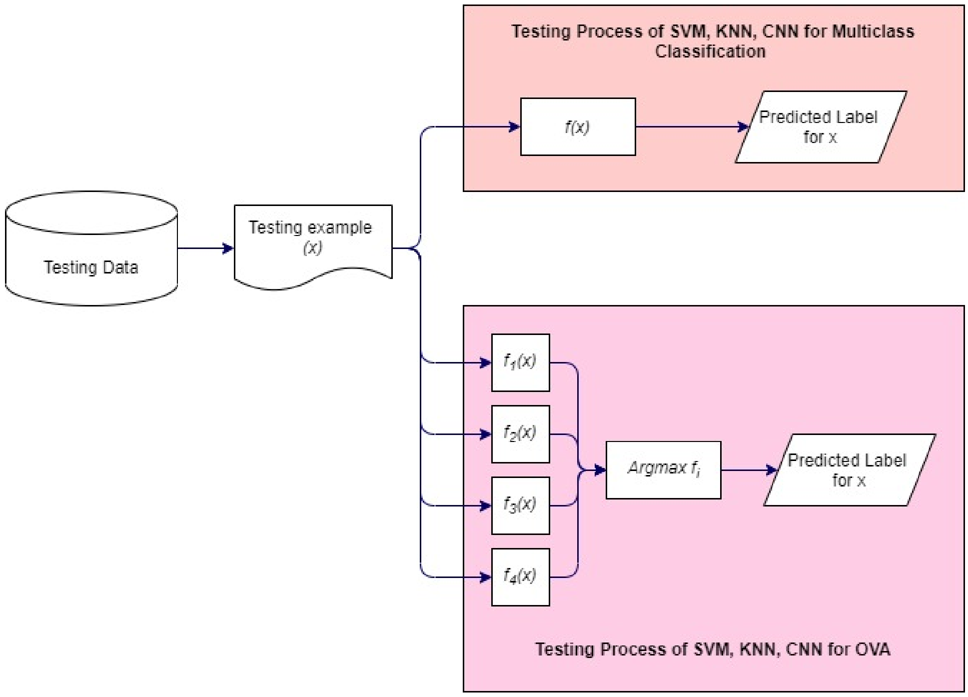

Fig. 6. Illustration of testing process

The purpose of the evaluation was to choose the best model according to the results of the testing process. The evaluation process is conducted by comparing the testing results of various trained models in the form of an f-measure. Models of multiclass or OVA multiclass classification with the best f-measure are selected and stored as models for temperament detection applications, as explained in Section 2.4.

2.4. Temperament Detection for Twitter User

In this process, input is accepted in the form of a Twitter username, and then crawling for Twitter data from related users is done. The next step is to apply preprocessing, feature extraction (both behavioural and grammatical), data scaling, and temperament detection processes. These processes are the same as those applied to the process of dataset construction and model generation for temperament detection, as explained in Sections 2.2 and 2.3, respectively. The temperament detection process was performed using the model with the best performance evaluation of the testing process.

3. Results and Discussion

3.1. Scenario Experiments

There were three scenarios for the experiment in this research. In Scenario 1, CNN was implemented with three different architectures, as shown in Fig. 3, Fig 4, Fig 5, while, in Scenario 2, classical machine learning was implemented, including SVM and KNN. Subsequently, the results of Scenarios 1 and 2 were compared in Scenario 3. The details of each scenario are shown in Fig. 7.

3.2. Experiment Results and Analysis

The best model of multiclass classification in Scenario 1a (Table 3) was obtained when using kernel size 7, dropout value 0.3, and learning rate 0.0007 with a 27.46% average value of the f-measure from the guardian, artisan, idealist, and rational classes. The best model of OVA multiclass classification obtained a 50% average value of the f-measure from the guardian, artisan, idealist, and rational classes when using kernel size 7, dropout value 0.7, and learning rate 0.00007.

In Scenario 1b, the best model of multiclass classification was obtained when using pool size 5, dropout value 0.5, and learning rate 0.0007 with a 32.75% average value of the f-measure from the guardian, artisan, idealist, and rational class. The best model of OVA multiclass classification obtained a 57.68% average value of the f-measure from the guardian, artisan, idealist, and rational classes when using pool size 7, dropout value 0.5, and learning rate 0.00007. The best model of multiclass classification in scenario 1c was obtained when using pool size 3, dropout value 0.3, and learning rate 0.0007 with a 31.56% average value of the f-measure from the guardian, artisan, idealist, and rational class. The best model of OVA multiclass classification obtained a 58.73% average value of the f-measure from the guardian, artisan, idealist, and rational classes when using pool size 3, dropout value 0.7, and learning rate 0.0007.

Table 3. Testing Results in Scenario 1 (F-measure)

	CNN 1	CNN 2	CNN 3
Multiclass	27.46%	32.75%	31.56%
OVA Multiclass	50%	57.68%	58.73%

Table 4 shows the testing results of Scenario 2. The best model of multiclass classification in Scenario 2a was obtained when using min-max data scaling and a linear kernel with a 30.38% average value of the f-measure from the guardian, artisan, idealist, and rational classes. The best model of OVA multiclass classification obtained a 57.27% average value of the f-measure from the guardian, artisan, idealist, and rational classes when using Z-score data scaling and a linear kernel. The best model of multiclass classification in Scenario 2b was obtained when using Z-score data scaling and one nearest neighbour with a 28.89% average value of the f-measure from the guardian, artisan, idealist, and rational classes. The best model of OVA multiclass classification obtained a 54.01% average value of the f-measure from the guardian, artisan, idealist, and rational classes when using Z-score data scaling and five nearest neighbours.

Table 4. Testing Results in Scenario 2 (F-measure)

	SVM	KNN
Multiclass	30.38%	28.89%
OVA Multiclass	57.27%	54.01%

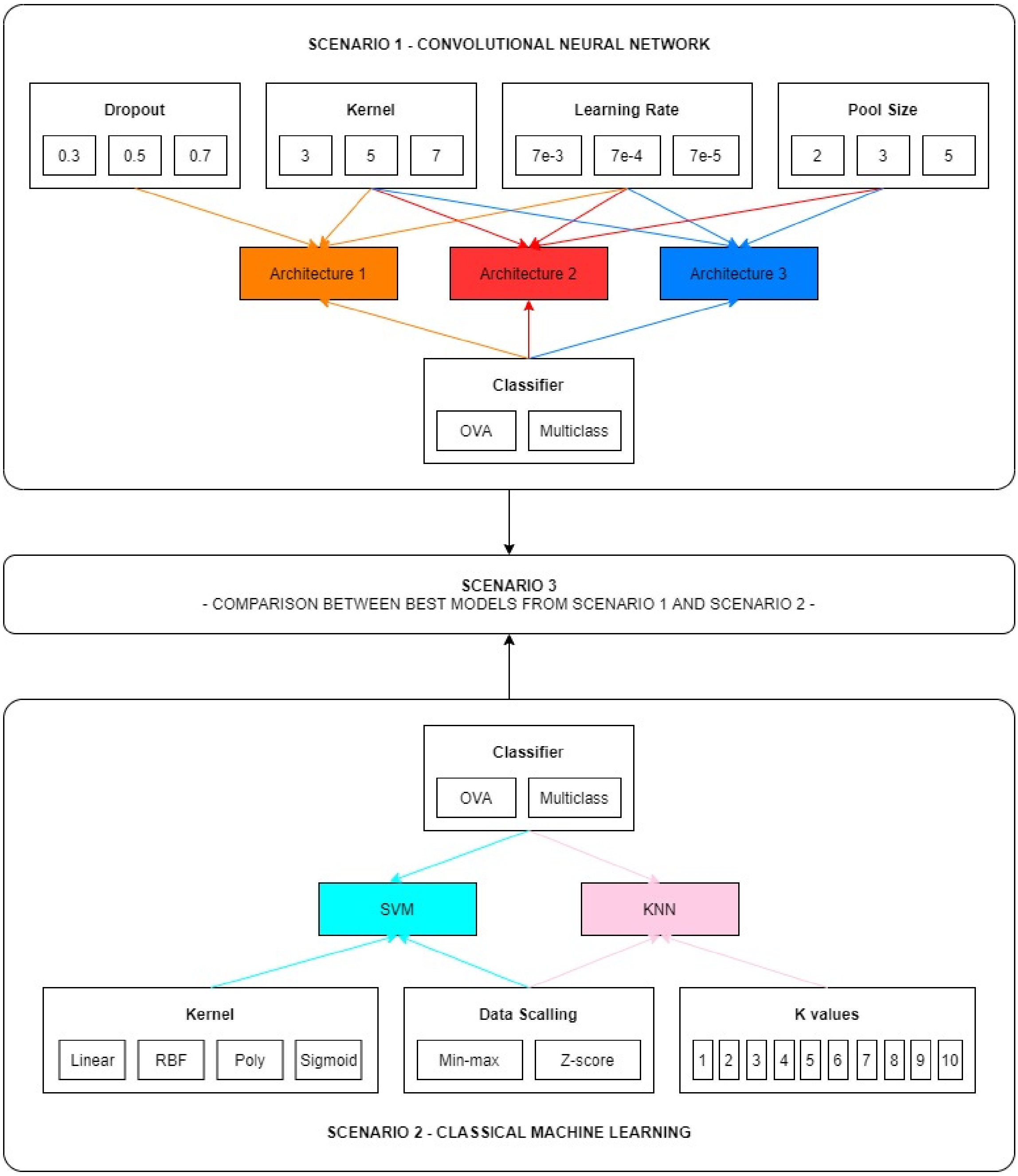

Fig. 7. Illustration of Scenario Experiments

The experimental results of multiclass and OVA multiclass classification in Scenario 1 show that the values of precision, recall, and f-measure obtained by using OVA multiclass classification were higher than the values obtained by multiclass classification. The best average value of the f-measure of OVA multiclass classification from Scenario 1 was obtained by CNN Method 3 in Scenario 1c with an f-measure of 58.73% and a combination of parameters: pool size 3, dropout value 0.7, and learning rate 0.0007. In Scenario 2, the OVA multiclass classifications also reached higher average values of precision, recall, and f-measure compared with multiclass classification. The highest value of the f-measure from Scenario 2 was obtained by the SVM classifier in Scenario 2a with an f-measure of 57.27% when using the Z-score data scaling linear kernel. The experimental results from all scenarios show that OVA multiclass classification always achieved higher results compared with multiclass classification. This was because, in multiclass classification, the classifier distinguishes four types of class that have high similarities, so it is difficult to distinguish the area of each class, resulting in low evaluation results. In contrast, in the OVA multiclass classification, each classifier is easier to learn during training because it only distinguishes two classes. In addition, the differences in characteristics between classes are higher because one class is compared with all other classes.

Table 5. Testing Results in Scenario 3 (F-measure)

	Scenario 1	Scenario 2
Multiclass	31.56%	30.38%
OVA Multiclass	58.73%	57.27%

4. Conclusion

The CNN deep learning method with OVA multiclass classification was the best classification method for temperament detection of Twitter users in the Indonesian language with an f-measure of 58.73%. The best classifier models were obtained when using OVA multiclass classification, resulting in better performance than that of multiclass classification, both with classical machine learning and deep learning methods. OVA multiclass classification produced better performance because the formation of the model was done in stages by breaking the multiclass problem into four binary problems by distinguishing one class from another for each class. The deep learning method obtained better performance results than those of the classical machine learning method because deep learning methods learn deeper in the mapping of raw input data into feature representation using a layered algorithmic structure.

Acknowledgment

The authors would like to acknowledge the research funding by Universitas Diponegoro under the grant of research of international publications, numbers 329-92/UN7.P4.3/PP/2019 and 329-92/UN7.6.1/PP/2020.

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.
Funding statement. The research funded by Universitas Diponegoro under the grant of research of international publications, numbers 329-92/UN7.P4.3/PP/2019 and 329-92/UN7.6.1/PP/2020.
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] D. Keirsey, *Please understand me II: Temperament, character, intelligence*. Prometheus Nemesis Book Company, 1998. Available at: Google Scholar.

[2] N. Majumder, S. Poria, A. Gelbukh, and E. Cambria, “Deep Learning-Based Document Modeling for Personality Detection from Text,” *IEEE Intell. Syst.*, vol. 32, no. 2, pp. 74–79, Mar. 2017, doi: 10.1109/MIS.2017.23.

[3] D. Xue *et al.*, “Personality Recognition on Social Media With Label Distribution Learning,” *IEEE Access*, vol. 5, pp. 13478–13488, 2017, doi: 10.1109/ACCESS.2017.2719018.

[4] S. C. Guntuku, D. B. Yaden, M. L. Kern, L. H. Ungar, and J. C. Eichstaedt, "Detecting depression and mental illness on social media: an integrative review," *Curr. Opin. Behav. Sci.*, vol. 18, pp. 43–49, Dec. 2017, doi: 10.1016/j.cobeha.2017.07.005.

[5] Y. Win, "Classification using Support Vector Machine to Detect Cyberbullying in Social Media for Myanmar Language," in *2019 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia)*, 2019, pp. 122–125, doi: 10.1109/ICCE-Asia46551.2019.8942212.

[6] A. S. M. Alharbi and E. de Doncker, "Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information," *Cogn. Syst. Res.*, vol. 54, pp. 50–61, May 2019, doi: 10.1016/j.cogsys.2018.10.001.

[7] M. L. Kern, P. X. McCarthy, D. Chakrabarty, and M.-A. Rizoiu, "Social media-predicted personality traits and values can help match people to their ideal jobs," *Proc. Natl. Acad. Sci.*, vol. 116, no. 52, pp. 26459–26464, Dec. 2019, doi: 10.1073/pnas.1917942116.

[8] B. Y. Pratama and R. Sarno, "Personality classification based on Twitter text using Naive Bayes, KNN and SVM," in *2015 International Conference on Data and Software Engineering (ICoDSE)*, 2015, pp. 170–174, doi: 10.1109/ICODSE.2015.7436992.

[9] I. A. Harsehanto and M. D. R. Wahyudi, "Analysis of Personality Characteristic Using the Naïve Bayess Classifier Algorithm (Case Study Official Twitter of Basuki Tjahaja Purnama's and Anies Baswedan)," *IJID (International J. Informatics Dev.*, vol. 7, no. 2, p. 14, Jan. 2019, doi: 10.14421/ijid.2018.07203.

[10] A. C. E. S. Lima and L. N. de Castro, "Predicting Temperament from Twitter Data," in *2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, 2016, pp. 599–604, doi: 10.1109/IIAI-AAI.2016.239.

[11] S. C. Guntuku, W. Lin, J. Carpenter, W. K. Ng, L. H. Ungar, and D. Preoţiuc-Pietro, "Studying Personality through the Content of Posted and Liked Images on Twitter," in *Proceedings of the 2017 ACM on Web Science Conference*, 2017, pp. 223–227, doi: 10.1145/3091478.3091522.

[12] A. K. John, A. Adewale M., and E. Chinnasa, "Temperament and Mood Detection Using Case-Based Reasoning," *Int. J. Intell. Syst. Appl.*, vol. 6, no. 3, pp. 50–61, Feb. 2014, doi: 10.5815/ijisa.2014.03.05.

[13] C. F. Claro, A. C. E. S. Lima, and L. N. de Castro, "Predicting Temperament using Keirsey's Model for Portuguese Twitter Data," in *Proceedings of the 10th International Conference on Agents and Artificial Intelligence*, 2018, pp. 250–256, doi: 10.5220/0006700102500256.

[14] A. C. E. S. Lima and L. N. de Castro, "TECLA: A temperament and psychological type prediction framework from Twitter data," *PLoS One*, vol. 14, no. 3, p. e0212844, Mar. 2019, doi: 10.1371/journal.pone.0212844.

[15] D. Suhartono *et al.*, "Personality Prediction Based on Twitter Information in Bahasa Indonesia," 2017, pp. 367–372, doi: 10.15439/2017F359.

[16] A. Husseini Orabi, P. Buddhitha, M. Husseini Orabi, and D. Inkpen, "Deep Learning for Depression Detection of Twitter Users," in *Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic*, 2018, pp. 88–97, doi: 10.18653/v1/W18-0609.

[17] S. Liao, J. Wang, R. Yu, K. Sato, and Z. Cheng, "CNN for situations understanding based on sentiment analysis of twitter data," *Procedia Comput. Sci.*, vol. 111, pp. 376–381, 2017, doi: 10.1016/j.procs.2017.06.037.

[18] J. W. Pennebaker, R. L. Boyd, K. Jordan, and K. Blackburn, "The development and psychometric properties of LIWC2015," 2015. Available at: Google Scholar.

[19] P. Pawara, E. Okafor, M. Groefsema, S. He, L. R. B. Schomaker, and M. A. Wiering, "One-vs-One classification for deep neural networks," *Pattern Recognit.*, vol. 108, p. 107528, Dec. 2020, doi: 10.1016/j.patcog.2020.107528.

[20] Y. R. Tausczik and J. W. Pennebaker, "The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods," *J. Lang. Soc. Psychol.*, vol. 29, no. 1, pp. 24–54, Mar. 2010, doi: 10.1177/0261927X09351676.

[21] D. P. Dudău and F. A. Sava, "The development and validation of the Romanian version of Linguistic Inquiry and Word Count 2015 (Ro-LIWC2015)," *Curr. Psychol.*, Jun. 2020, doi: 10.1007/s12144-020-00872-4.

[22] G. Orellana, B. Arias, M. Orellana, V. Saquicela, F. Baculima, and N. Piedra, "A Study on the Impact of Pre-Processing Techniques in Spanish and English Text Classification over Short and Large Text Documents," in *2018 International Conference on Information Systems and Computer Science (INCISCOS)*, 2018, pp. 277–283, doi: 10.1109/INCISCOS.2018.00047.

[23] Y. A. Putra and M. L. Khodra, "Deep learning and distributional semantic model for Indonesian tweet categorization," in *2016 International Conference on Data and Software Engineering (ICoDSE)*, 2016, pp. 1–6, doi: 10.1109/ICODSE.2016.7936108.

[24] R. M. Cahyaningtyas, R. Kusumaningrum, Sutikno, Suhartono, and D. E. 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)*, 2017, pp. 253–258, doi: 10.1109/ICICOS.2017.8276371.

[25] R. B. S. Putra and E. Utami, "Non-formal affixed word stemming in Indonesian language," in *2018 International Conference on Information and Communications Technology (ICOIACT)*, 2018, pp. 531–536, doi: 10.1109/ICOIACT.2018.8350735.

[26] X. H. Cao, I. Stojkovic, and Z. Obradovic, "A robust data scaling algorithm to improve classification accuracies in biomedical data," *BMC Bioinformatics*, vol. 17, no. 1, p. 359, Dec. 2016, doi: 10.1186/s12859-016-1236-x.

[27] G. Aksu, C. O. Güzeller, and M. T. Eser, "The Effect of the Normalization Method Used in Different Sample Sizes on the Success of Artificial Neural Network Model," *Int. J. Assess. Tools Educ.*, pp. 170–192, Apr. 2019, doi: 10.21449/ijate.479404.

[28] M. Faisal, E. M. Zamzami, and Sutarman, "Comparative Analysis of Inter-Centroid K-Means Performance using Euclidean Distance, Canberra Distance and Manhattan Distance," *J. Phys. Conf. Ser.*, vol. 1566, no. 1, p. 012112, Jun. 2020, doi: 10.1088/1742-6596/1566/1/012112.

[29] I. Bin Mohamad and D. Usman, "Standardization and Its Effects on K-Means Clustering Algorithm," *Res. J. Appl. Sci. Eng. Technol.*, vol. 6, no. 17, pp. 3299–3303, Sep. 2013, doi: 10.19026/rjaset.6.3638.

[30] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-Validation," in *Encyclopedia of Database Systems*, Boston, MA: Springer US, 2009, pp. 532–538. doi: 10.1007/978-0-387-39940-9_565.