Multidisciplinary classification for Indonesian scientific articles abstract using pre-trained BERT model

Antonius Angga Kurniawan [a,1,*], Sarifuddin Madenda [a,2], Setia Wirawan [a,3], Ruddy J Suhatril [a,4]

[a] Department of Information Technology, Gunadarma University, Depok 16424, Indonesia
[1] anggaku@staff.gunadarma.ac.id; [2] sarif@staff.gunadarma.ac.id; [3] setia@staff.gunadarma.ac.id; [4] ruddy_js@staff.gunadarma.ac.id
* corresponding author

ARTICLE INFO

Article history
Received March 3, 2023
Revised April 12, 2023
Accepted April 24, 2023
Available online July 8, 2023

Keywords
Abstract
BERT
Classification
Fine-tuned hyperparameter
Multidisciplinary

ABSTRACT

Scientific articles now have multidisciplinary content. These make it difficult for researchers to find out relevant information. Some submissions are irrelevant to the journal's discipline. Categorizing articles and assessing their relevance can aid researchers and journals. Existing research still focuses on single-category predictive outcomes. Therefore, this research takes a new approach by applying a multidisciplinary classification for Indonesian scientific article abstracts using a pre-trained BERT model, showing the relevance between each category in an abstract. The dataset used was 9,000 abstracts with 9 disciplinary categories. On the dataset, text preprocessing is performed. The classification model was built by combining the pre-trained BERT model with Artificial Neural Network. Fine-tuning the hyperparameters is done to determine the most optimal hyperparameter combination for the model. The hyperparameters consist of batch size, learning rate, number of epochs, and data ratio. The best hyperparameter combination is a learning rate of 1e-5, batch size 32, epochs 3, and data ratio 9:1, with a validation accuracy value of 90.8%. The confusion matrix results of the model are compared with the confusion matrix results by experts. In this case, the highest accuracy result obtained by the model is 99.56%. A software prototype used the most accurate model to classify new data, displaying the top two prediction probabilities and the dominant category. This research produces a model that can be used to solve Indonesian text classification-related problems.

1. Introduction

Nowadays, scientific articles are growing very rapidly with increasingly diverse topics in various disciplines [1]. The scientific articles created have the possibility of containing one discipline and or multidisciplinary at the same time. Akshai and Anitha [2] conducted research for the early detection of plant diseases using deep learning with the Convolutional Neural Network (CNN) model. This example shows that this scientific article belongs to multidisciplinary research because it contains more than one discipline, namely agricultural and computer science.

One of the main concerns of many researchers is finding scientific articles relevant to their discipline and or object of research as scientific literature. This process can be time-consuming. In addition, sometimes, from the many articles submitted to the journal portal, there are still articles whose disciplines differ from those in the journal portal. Therefore, automatically classifying scientific articles into relevant disciplinary categories and knowing the relevance of one discipline to another in a scientific article can be helpful to researchers and journal portals.

Many researches have been conducted related to text classification such as abstract, sentiment, news category classification, and others. However, in previous researches there are still some things that can be developed to be better new research. In general, text classification conducted by previous researchers still uses traditional word embedding models such as Word2Vec, FastText, and TF-IDF [1], [3]–[9]. Currently, an advanced word embedding model provides impressive results in solving natural language processing (NLP) problems, namely the pre-trained BERT model [10]. Most of the text classifications using BERT conducted by previous researchers used English datasets and the research topics were mostly about sentiment or news category classification [11]–[17]. Such researches did not go through the text preprocessing stage on the dataset used [9], [12], [13], where the research stated that the use of text preprocessing can affect the accuracy results obtained by the model. In addition, most of the previous researches only conducted one test of the hyperparameter combination of learning rate, batch size, number of epochs, and data ratio [3], [12], [13], [16], [17]. In text classification, to get a model with the best accuracy, it is necessary to test the hyperparameters more than once or commonly referred to as fine-tuning hyperparameters. In previous research [1], [16], [18]–[21], the text classification carried out only focuses on prediction results with one category without seeing the possibility that a classified text can contain one and or more than one categories.

Based on the results of the search conducted by the previous researchers, pre-trained BERT models applied to Indonesian datasets for the classification of scientific articles based on abstracts is challenging. In addition, this research takes a new approach by proposing the development and implementation of multidisciplinary classification on abstracts of scientific articles in Indonesian using the pre-trained BERT model. The main contribution of this research, the first is to combine the pre-trained BERT model with Artificial Neural Network (ANN) to classify abstracts of scientific articles in Indonesian. The second is to perform text preprocessing such as lowercase text, text cleaning, tokenizing, and stopword removal on the dataset used. The third is to perform hyperparameter fine-tuning to carry out on the value of the learning rate, batch size, number of epochs, and data ratio to get a combination of hyperparameters with the most optimal model. The fourth, the model with the most optimal results is implemented into a prototype to classify abstracts of scientific articles automatically. In addition, two categories that have the highest probability are displayed, so that the relevance between one discipline and another in a scientific article can be known and can place a scientific article according to the appropriate discipline and or research object.

2. Method

2.1. Research Stages

The method used in this research is described into nine stages: Data Collection, Data Labeling, Text Preprocessing, Data Splitting, Model Architecture Development, Fine-Tuning Hyperparameter Model, Model Evaluation, Model Implementation into Software Prototype, and Testing New Data. Fig. 1 shows the general stages of this research.

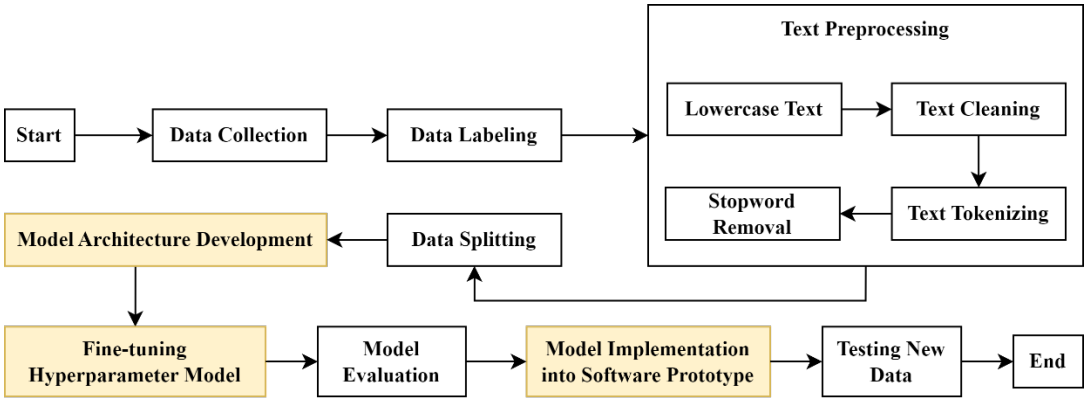

Fig. 1. Research stages

2.2. Data Collection

In this research, the data used are abstracts in the Indonesian language in scientific articles. The abstracts come from national journal portals that have been accredited by Science and Technology Index (SINTA). The abstracts used have been published and have gone through a review process by editors or expert reviewers in each journal.

The abstracts used consist of nine categories of science fields, namely Business, Law, Health, Computer, Communication, Mathematics, Education, Agriculture, and Engineering. These categories refer to several fields of science in accordance with the existing study programs in the formal and applied science clusters based on the Decree of the Minister of Research, Technology and Higher Education in 2019. The amount of data collected was 9,000 abstracts. Fig. 2 shows the number of abstracts in each category is 1,000 abstracts. The data is saved into a file with Comma Separated Value (CSV) format.

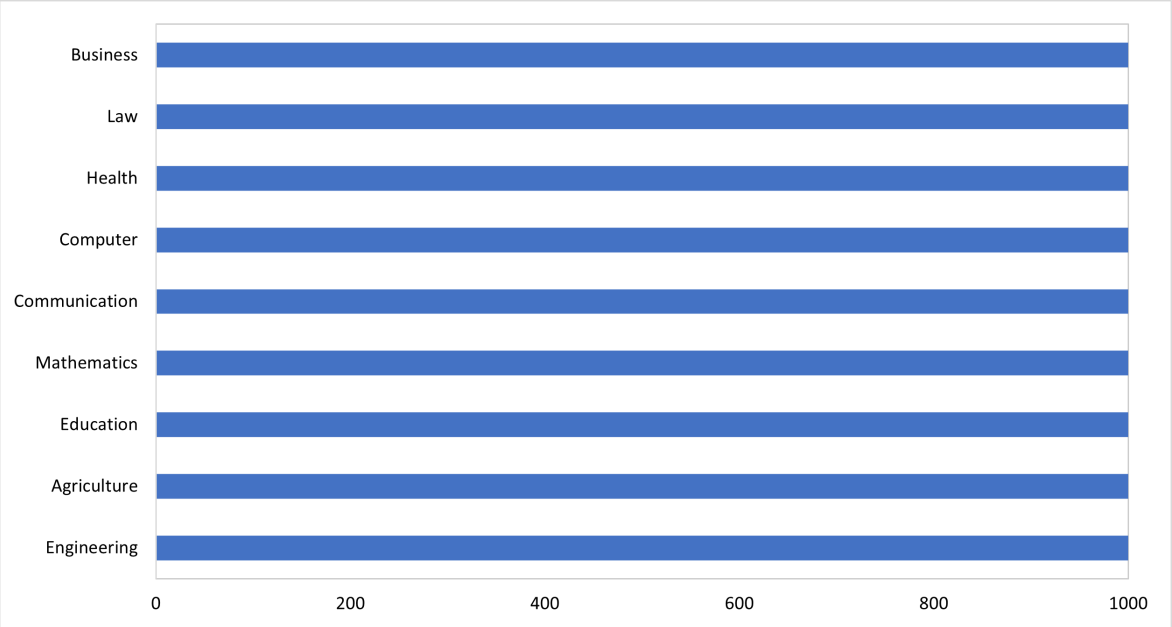

Fig. 2. Total number of abstracts per category

2.3. Data Labeling

The data labeling process is carried out to provide a separate annotation to each abstract into the specified science field category. The label is determined based on the field of science on the journal portal in which the abstract is published. The characteristics of the data are divided into two, namely abstract data with the category of one discipline and multidisciplinary fields of science. Labeling was done by automatically assigning numerical numbers from 0 to 8 to 9,000 abstracts based on their category labels. The labels for each category can be seen in Table 1.

Table 1. Category label

Category	Label
Business	0
Law	1
Health	2
Computer	3
Communication	4
Mathematics	5
Education	6
Agriculture	7
Engineering	8

The validation data is also manually labeled by experts. The validation data will be labeled by experts according to the list of science field categories in Table 1. The labeling process on validation data by

experts allows an abstract to have one and or more than one science field categories. Validation data that has been labeled by experts will become ground truth for the classification process of abstracts of multidisciplinary scientific articles.

2.4. Text Preprocessing

Text preprocessing is one of the key processes of classification, especially in natural language processing (NLP). The steps in text preprocessing aim to improve word structure and to reduce the ambiguity value when extracting features [22], [23]. Text preprocessing in this research consists of lowercase text, text cleaning, text tokenizing, and stopword removal.

2.5. Lowercase Text

The lowercase text process is carried out to convert the entire set of abstract text into lowercase letters. The process is done to help the machine read and process the abstract better. The lowercase text function used is a string function from the pandas python library, namely str.lower(). The function will check and read each line of the abstract. If there is a letter that is included in the capital letter it will be immediately changed to lowercase, after that the next line is checked and read until the last line of the abstract.

2.6. Text Cleaning

The process of preparing raw text for NLP so that machines can understand human language is known as text cleaning [22]. Text cleaning of abstract data consists of removing tabs, new lines, multiple spaces, punctuation, special characters, urls, numbers, and single characters.

2.7. Text Tokenizing

The text tokenizing process is carried out to break the text in the form of sentences in the abstract into pieces in the form of tokens, namely words. So that each line in the abstract column contains pieces of tokens that come from each abstract [6]. In this research, a function from NLTK python is used, namely word tokenize. At this stage a machine receives input in the form of text to be tokenized. Next, the text will be separated word by word. After that, the words are converted into an array on each line, then the result of the tokenization process is obtained.

2.8. Stopword Removal

Stopword removal is the process of filtering or selecting words that are important for the display of text. Stopword removal aims to eliminate words that lack meaning or are irrelevant to the data subject used. These words include prepositions, determiners, conjunctions, and other similar words [23]. The stopword removal process is done by using the nltk corpus stopwords library in python. In addition, this study also used a stopword corpus with the Indonesian language created by Devid and Martijn [24] with reference from Tala research [25]. This corpus contains a set of words that are considered not very important or have no meaning in a text or sentence. The input used in the stopword removal process is the result of the text tokenizing process. The stopword remover reads the word and checks word by word whether it is available in the stopword corpus. If so, the word is removed, otherwise, the word is kept.

2.9. Data Splitting

Data that has completed the text preprocessing process will then be separated into two parts, namely training data and validation data. In performing classification, the data used to train the model must be larger than the validation data. The validation data used will validate the model and prevent overfitting. Epoch performs the training and validation process sequentially. When the training is complete, the validation process continues. Therefore, the data used needs to be separated into training data and validation data. In this research, the validation data used will be stored and then manually labeled by experts.

In this research, there is also test data as much as 90 abstract data. Test data is data used to test the model after the training and validation process is complete. The test data is included in the unseen data.

This means that the data has never been seen or recognized by the model during the training process. This test data will be tested using the best model that has been implemented into the prototype. This is done to find out how well the model performs to predict data that has never been seen or recognized before.

2.10. Model Architecture Development

In this research, the Indonesian pre-trained BERT model was used. The model was pre-trained with the Indonesian Wikipedia. The Indonesian pre-trained BERT model used is "cahya/bert-base-indonesian-522M". The model was pre-trained with 522MB of Indonesian Wikipedia that had been converted to lowercase text and tokenized using WordPiece and a vocabulary size of 32,000 [26]. The Transformer-based model is commonly used for text classification, text generation, and others. Fig. 3 shows the architecture of the model built by combining the pre-trained model cahya/bert-base-indonesian-522M and ANN.

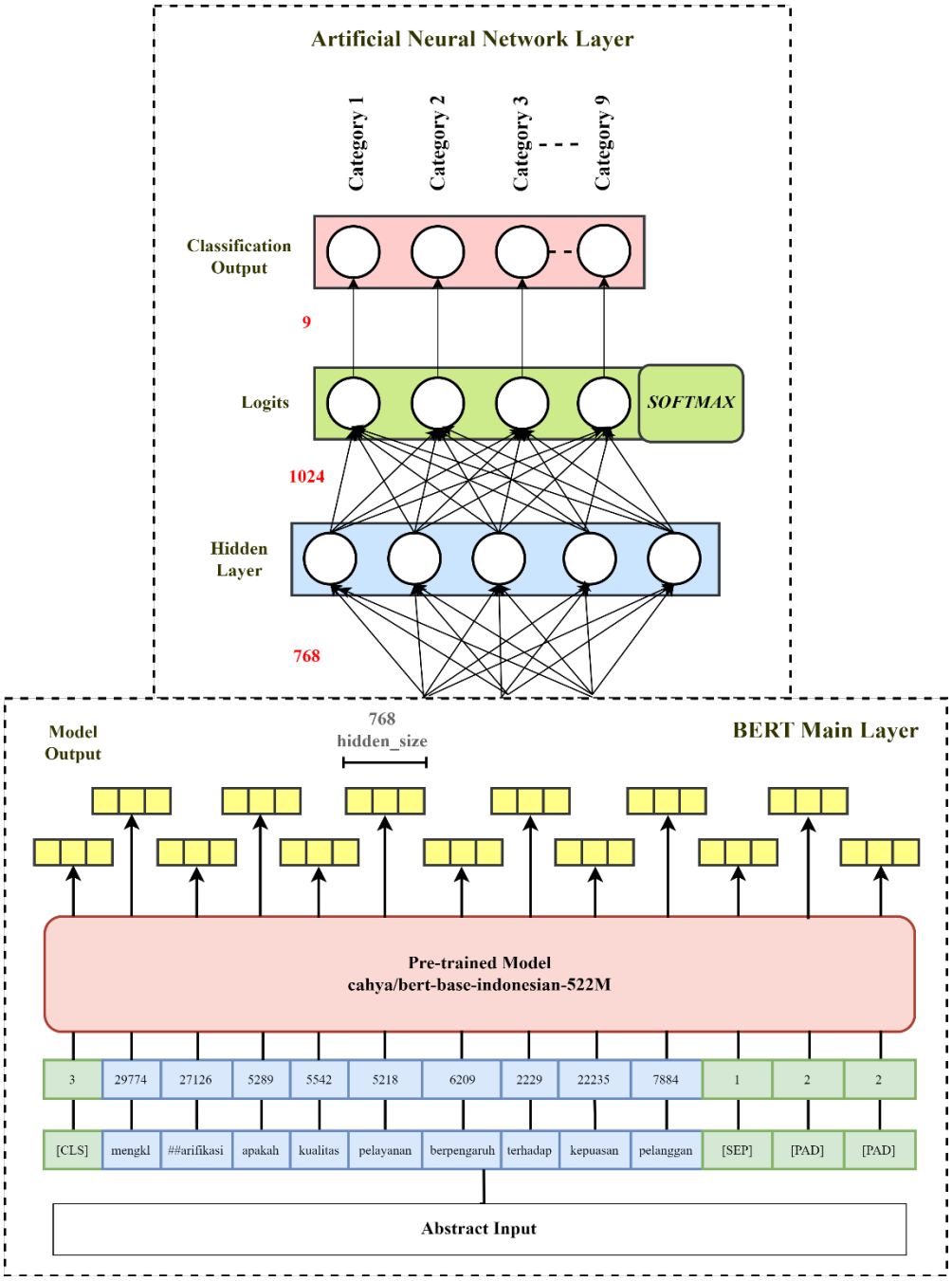

Fig. 3. Model architecture for classification of abstracts of scientific articles

Before the dataset is classified, a sentence or text needs to be converted into the input representation required by BERT. In Fig. 3 there is an example of one sentence to be classified. A special token [CLS] is added to the beginning of the classified sentence to represent the meaning of the whole sentence. Then a special token [SEP] is added at the end of the sentence to separate the first sentence from the next. The record is then set to the initialized maximum length by either truncating the record if it is longer than the maximum length or giving it the special character [PAD] if it is shorter than the maximum length. After that, each text is matched with a unique number or ID contained in the vocabulary and the unique number is stored as a token ID. Each single character, subword, and word has its own special number in the vocabulary. The word or token must be changed using the ID. Based on the example sentence entered, the ID sequence is [3, 29774, 27126, 5289, 5542, 5218, 6209, 2229, 22235, 7884, 1, 2, 2]. The token ID will then proceed through the next stack of encoders. Then self-attention is applied by each encoder and given an output through the feed-forward network. After that it continues into the next encoder. This research uses the Indonesian BERT Base model, so the process takes place 12 times. After all encoders have been successfully passed, the next output is given in the form of a vector of each token according to its position. The given vector has a hidden size of 768 as seen in Fig. 3. The token [CLS] represents the output vector derived from the BERT model. Furthermore, the vector is used as input for abstract classification. Average collection of word tokens is performed by [CLS] with the aim of obtaining vectors derived from sentences. The output layer as the last layer creates logits. In this case, logits are output values in the form of rough probability predictions based on the classified sentence or text. Softmax then converts this logit to a probability value by computing the exponential value of each logit value, so the total probability is exactly 1. In this case, the probability values range from 0 to a positive number.

The architecture of the model built has one input layer, where in the input layer there is also an attention mask. In this case, the attention layer usually uses padding tokens in the context seen for each token in the sentence. To tell the attention layer to ignore token padding, it is necessary to add an attention mask. With the right attention mask, the resulting prediction will be the same for a given sentence, whether using padding or not using padding [27]. In this research, the attention mask is useful to show the model which tokens need attention and which tokens do not. Attention mask will show the model the position of the padding index, so the model does not need to process that token. In this case if there is a sentence whose number of tokens is less than the specified maximum token length, then there will be a [PAD] or padding token.

Next, there is one hidden layer with ReLU activation function. ReLU is used because the output of this layer ranges between 0 and infinity. ReLU significantly speeds up the convergence process performed in stochastic gradient descent compared to Sigmoid or Tanh. Furthermore, ReLU basically only creates a range of zeros. That is, x equals zero if x is less than equal to zero and x equals x if x is greater than zero [28]. The number of neurons or nodes used in the hidden layer is 1024. This is because too few neurons used in the hidden layer can lead to so-called underfitting. Underfitting occurs because there are too few neurons in the hidden layer to detect enough signal in complex datasets. On the other hand, using too many neurons in the hidden layer can lead to overfitting. Overfitting occurs when neural networks are highly intelligent and the training set has insufficient information to train all of the neurons in the hidden layer. Despite having enough training data. The use of a large number of neurons in the hidden layer can lengthen the time it takes to train the network. It can increase training time and prevent the neural network from training properly. Overfitting prevents the model from predicting possible outputs for unknown or untrained inputs, preventing the model from producing accurate outputs [29].

Finally, an output layer useful for classification using a fully connected neural network and Softmax as an activation function with 9 neurons. In terms of the number of neurons determined, it must be equal to the number of class labels or categories used in the data. In this research, the output categories produced are more than two categories of science fields, where the output produced is 9 categories, namely Business, Law, Health, Computer, Communication, Mathematics, Education, Agriculture, and Engineering.

2.11. Fine-Tuning Hyperparameter Model

In the research, four training scenarios were tested. In the training process, fine-tuning of the model was carried out by adjusting four hyperparameters, namely learning rate, batch size, number of epochs, and data ratio with a maximum word length of 150. The four training scenarios can be seen in Table 2.

Table 2. Scenario of hyperparameter tuning

Hyperparameters	Hyperparameter Combination			
	Learning Rate	*Batch Size*	*Epochs*	*Data Ratio*
Learning Rate	1e-5	32	4	9:1
	2e-5	32	4	9:1
	5e-5	32	4	9:1
Batch Size	Best Learning Rate in the previous scenario	8	4	9:1
		16	4	9:1
		32	4	9:1
		64	4	9:1
Epochs	Best Learning Rate in the previous scenario	Best Batch Size in the previous scenario	2	9:1
			3	9:1
			4	9:1
			5	9:1
Data Ratio	Best Learning Rate in the previous scenario	Best Batch Size in the previous scenario	Best Epoch in the previous scenario	9:1
				8:2
				7:3

The first training scenario tests the learning rate values used during the model training process. Learning rate is one of the most important parameters for improving model performance. Learning rate is used to determine the step size at each iteration to get the weight value adjustment with the minimum error in the training process. Choosing a learning rate that is too low and a learning rate that is too high will reduce the performance of the model [30]. Determining the right learning rate value can find the weight for the minimum error value, so that an optimal solution can be achieved in the model. The learning rates tested in the first training scenario are 1e-5, 2e-5, and 5e-5. In addition, testing was conducted using other specified hyperparameters, namely the number of epochs of 4, batch size of 32, and data ratio of 9:1. These other hyperparameters were chosen because based on the results of research conducted by previous researchers using 4 epochs [10], batch size 32 [10], [12] has good accuracy results.

A second training scenario was tested to establish the best batch size in the created model's training procedure. The number of training samples used by the neural network in one iteration (batch) is referred to as batch size. Determining the batch size is important to be adjusted to the model built because the size of the number of samples entering the neural network can determine a more optimal weight value [30]. The testing hyperparameters used consist of the best learning rate obtained from testing in the first training scenario as well as using other predetermined hyperparameters, namely the number of epochs of 4 and the data ratio of 9:1. The batch sizes tested in the second training scenario were 8, 16, 32, and 64.

A third training scenario is tested to determine the effect of the number of epochs on the accuracy validation score of the built model. Epochs are useful for determining the number of times the training process is carried out in the neural network on the entire amount of data. This is quite important because training all data to update and get the most optimal weight value is not enough if only done using one epoch [31]. The hyperparameters for the third training scenario use the best hyperparameters from the learning rate and batch size generated in the previous scenario. In addition, other predetermined hyperparameters such as a data ratio of 9:1 were used. The number of epochs tested in the third training scenario were 2, 3, 4, and 5.

The fourth training scenario was evaluated to see how the ratio of training data and validation data affected the model's accuracy validation value. Training data is used to train an algorithm to identify a

suitable model, and validation data is used to test and determine the performance of the trained model. Comparison of the amount of training data and validation data is quite important during the learning process carried out by the model because it can help ensure that the model created is more accurate. The hyperparameters for the fourth training scenario used the best hyperparameters from the learning rate, batch size, and epoch generated in the previous scenario. In the fourth training scenario, the training data and validation data ratios are 9:1, 8:2, and 7:3.

2.12. Model Evaluation

In this section, the evaluation of the model that has been built is carried out to see the validation value of accuracy, computation time, and confusion matrix from the model training process. Accuracy validation is the value of the accuracy of model predictions obtained during the testing process. Computation time is the time required to train and test the model. A confusion matrix is an evaluation method for measuring the performance or accuracy of a model produced in a classification process [32].

The validation data that has been labeled by experts is then converted into a confusion matrix. The confusion matrix generated by the model is then compared with the confusion matrix by the expert. This aims to validate the classification suitability results produced by the model both inside and outside the diagonal line of the confusion matrix due to the possibility of abstracts that are multidisciplinary.

2.13. Model Implementation into Prototype

In this research, the model that has been built, trained, and evaluated with the most optimal evaluation results will be saved. The model will be implemented into a web-based software prototype to classify new input data into abstract categories that are more dominant than the 9 predetermined categories. The prototype software was built using a Web Server Gateway Interface (WSGI) application framework, Flask. The programming language used is python and the interface used is html.

2.14. Testing New Data

Testing is done by processing new abstract data that has never been trained or recognized by the model as much as 90 abstract data. In this study, the same steps as the training process were carried out, namely text preprocessing, then the new abstract data was used as input to be processed into the most optimal model that had previously been stored. The model will classify the input text into one of the more dominant categories from the 9 initialized abstract label categories.

In the prototype built, an input form will appear. Next, enter the abstract of the new text you want to predict. After entering the abstract, press Enter. Next, the model will process the classification results of the newly entered abstract. This process also performs a text preprocessing process and then will display the prediction results of the science field category based on the abstract entered. Finally, the results of the prediction probability values obtained from the model are shown for the two categories with the highest probabilities.

3. Results and Discussion

3.1. Example of Data Labeling Results by Experts

An example of the results of data labeling by experts on validation data can be seen in Table 3. In abstract number 1 it can be said that the abstract is likely to fall into the multidisciplinary category. This is because in abstract number 1 there is a relevance between the fields of Engineering and Agriculture, where in the abstract the object of research is related to Engineering and the scientific field is Agriculture. In abstract number 2 it can be said that the abstract is likely to fall into the category of one discipline. This is because in abstract number 2 the scientific field and the object of research carried out are only related to the field of Law.

3.2. Model Evaluation Results

In this research, hyperparameter fine-tuning is performed on the model using four training scenarios. This aims to get the most optimal hyperparameter combination for the model built. The hyperparameters used from the four training scenarios consist of learning rate, batch size, number of epochs, and number of data ratios. The first training scenario tested the value of the learning rate consisting of 1e-5, 2e-5, and 5e-5. Table 4 displays the model's results for the first training scenario.

Table 3. Examples of abstract data with single-discipline and multidiscipline categories

No	Abstract	Category
1	*Pada umumnya, pembasmian hama padi dilakukan dengan cara penyemprotan pestisida. Hal ini akan mengakibatkan tanah dan tanaman padi tercemar. Penelitian ini bertujuan untuk membuat alat pembasmi hama otomatis yang rama lingkungan tanpa menggunakan pestisida. … Pada perancangan ini, mikrokontroler Atmega 328 Arduino UNO digunakan sebagai pusat pengendali sistem, sensor LDR digunakan sebagai pengganti sakelar lampu DC di malam hari untuk membuat hama mendekat sesuai dengan karakteristiknya yang tertarik dengan cahaya, … Pemrograman menggunakan software Arduino IDE. Berdasarkan hasil pengujian selama 3 hari, alat pembasmi hama padi ini dapat membasmi 39 hama kepik hitam, penggerek batang padi, dan walang sangit.* **English Translation:** In general, the eradication of rice pests is done by spraying pesticides. This will result in contaminated soil and rice plants. This research aims to make an automatic pest exterminator tool that is environmentally friendly without using pesticides. ... In this design, the Atmega 328 Arduino UNO microcontroller is used as the system control center, the LDR sensor is used as a substitute for a DC light switch at night to make pests approach according to their characteristics that are attracted to light, ... Programming using Arduino IDE software. Based on the results of testing for 3 days, this rice pest exterminator can eradicate black ladybugs, rice stem borers, and stink bugs.	Engineering and Agriculture (Multidicipline)
2	*Praperadilan merupakan wewenang pengadilan negeri untuk memeriksa dan memutus menurut cara yang diatur dalam undang-undang, tentang sah atau tidaknya suatu penangkapan …. ke pengadilan. … Penelitian ini menggunakan metode penelitian hukum normatif yakni adanya kekosongan norma hukum di dalam Pasal 21 ayat (3) KUHAP dan Pasal 46 ayat (3) Peraturan Kapolri Nomor 14 Tahun 2012 … Hasil dari penelitian ini adalah mekanisme penyidik Kepolisian dalam melakukan penahanan kepada tersangka …. kepada keluarga tersangka. Akibat hukum bagi penyidik Kepolisian yang belum menyampaikan tembusan surat perintah tersebut dapat dijadikan dasar oleh pihak keluarga untuk menyatakan bahwa penahanan tersebut tidak sah karena telah melanggar hak asasi atau kebebasan hidup seseorang.* **English Translation:** Pretrial is the authority of the district court to examine and decide in the manner provided for in the law, about the legality or not of an arrest .... to the court. ... This research uses normative legal research methods, namely the existence of a legal norm vacuum in Article 21 paragraph (3) of the Criminal Procedure Code and Article 46 paragraph (3) of the National Police Chief Regulation Number 14 of 2012 ... The result of this research is the mechanism of Police investigators in detaining a suspect .... to the suspect's family. The legal consequences for Police investigators who have not submitted a copy of the warrant can be used as a basis by the family to declare that the detention is invalid because it has violated the human rights or freedom of life of a person.	Law (One-discipline)

Table 4 shows that different learning rates can affect model training results. The highest accuracy validation value is obtained when the learning rate used is 1e-5. In addition, in Table 4 it can also be seen that the higher the value of the learning rate, the faster the time required, but the lower the validation accuracy. This is because the smaller the learning rate value, the smaller the change in gradient descent, so the possibility of finding the weight value with the minimum error will be greater. This can

make the model more precise and can achieve a more optimal accuracy validation value. However, a smaller learning rate value requires a longer time to reach the most optimal solution. This can be shown in Table 4. However, considering the insignificant time difference between learning rate 1e-5 and 2e-5 and the higher validation accuracy value at learning rate 1e-5, it was decided to set learning rate 1e-5 as the best learning rate value in the first training scenario.

The second training scenario was tested to determine the optimal batch size for the model. The batch sizes tested in the second training scenario were 8, 16, 32, and 64. The learning rate hyperparameter used is the best learning rate obtained from testing in the first training scenario, which is 1e-5. Table 5 displays the model's results for the second training scenario.

Table 4. First Scenario Testing Results - Learning Rate

Learning Rate	Batch Size	Epochs	Data Ratio	Validation Accuracy	Time
1e-5	**32**	**4**	**9:1**	**90%**	**16min 44s**
2e-5	32	4	9:1	88.1%	16min 31s
5e-5	32	4	9:1	87.3%	15min 21s

According to the results in Table 5, the higher the batch size, the shorter the time required for training. In the second training scenario, the best validation accuracy is achieved when the batch size used is 32. This is because the larger the batch size used can help reduce noise in error calculation. However, a batch size that is too large also does not guarantee that the accuracy of the model will be better, because the number of samples used in one iteration becomes smaller.

Table 5. Second Scenario Testing Results – Batch Size

Learning Rate	Batch Size	Epochs	Data Ratio	Validation Accuracy	Time
1e-5	8	4	9:1	86.8%	25min 16s
1e-5	16	4	9:1	89.4%	20min 3s
1e-5	**32**	**4**	**9:1**	**90%**	**16min 44s**
1e-5	64	4	9:1	89.2%	13min 45s

A third training scenario is tested to determine the effect of the number of epochs on the accuracy validation score of the model. The hyperparameters for the third training scenario use the optimal batch size and learning rate obtained from testing in the previous scenario, namely a learning rate of 1e-5 and a batch size of 32. The number of epochs tested in the third training scenario, namely 2, 3, 4, and 5. Results for his third training scenario of the model are shown in Table 6.

Table 6. Third Scenario Testing Results – Epochs

Learning Rate	Batch Size	Epochs	Data Ratio	Validation Accuracy	Time
1e-5	32	2	9:1	89.7%	9min 17s
1e-5	**32**	**3**	**9:1**	**90.8%**	**12min 45s**
1e-5	32	4	9:1	90%	16min 44s
1e-5	32	5	9:1	89.2%	21min 3s

The results in Table 6 show that the higher the number of epochs, the longer the training process takes. The optimal number of epochs for a model built with the highest accuracy validation value is 3 epochs. The more the number of epochs, the more weight values will be updated. So the possibility to get the most optimal weight value will be greater. However, the number of epochs that are too many is

also not necessarily good for the model built, because at an epoch of 5 the accuracy validation value has decreased. This is because the number of epochs depends on the amount of data used during training.

The fourth training scenario was evaluated to see how the ratio of training data and validation data affected the model's validation accuracy value. Hyperparameters for the fourth training scenario use the best batch size, learning rate, and epoch resulting from testing in the first to third scenarios, namely learning rate 1e-5, batch size 32, and number of epochs of 3. In the fourth training scenario, the training data and validation data ratios were 9:1, 8:2, and 7:3. Results for his fourth training scenario for the model are shown in Table 7.

Table 7. Fourth Scenario Testing Results – Data Ratio

Learning Rate	Batch Size	Epochs	Data Ratio	Validation Accuracy	Time
1e-5	**32**	**3**	**9:1**	**90.8%**	**12min 45s**
1e-5	32	3	8:2	89.3%	12min 21s
1e-5	32	3	7:3	89.9%	11min 36s

Based on the results in Table 7, it appears that the more training data that is used, the longer the training process takes. The highest accuracy validation result obtained is when the data ratio used is 9:1 with an accuracy validation of 90.8%.

Based on four training scenarios that have been carried out on the model, a combination of hyperparameters with the highest accuracy validation value is obtained. The combination consists of hyperparameters with a learning rate of 1e-5, a number of epochs of 3, a batch size of 32, and a data ratio of 9:1. This hyperparameter combination resulted in an accuracy validation of 90.8% with a computation time of 12 minutes 45 seconds. The model with the best hyperparameter combination is named the AbBERT model (Abstract BERT).

This research has also tested several traditional machine learning models and deep learning models based on models used in previous studies such as SVM [1], [4], CNN [7], [16], LSTM [8], and other models such as Naive Bayes, Logistic Regression, KNN, Random Forest. Testing of these models is done using the same dataset and architecture as the AbBERT model. Table 8 shows the results of accuracy validation on each of these models.

Table 8. Accuracy Validation Results of Several Machine Learning and Deep Learning Models

Model	Validation Accuracy
Naïve Bayes	87.5%
SVM	89.3%
Logistic Regression	88.3%
KNN	77.7%
Random Forest	86.1%
CNN	84.4%
LSTM	76.7%

The amount and data used for classification using several machine learning and deep learning models are the same as the AbBERT model, which is 9,000 abstract data. The ratio of training data and validation data used is 9:1 with a maximum word length of 150. Then for CNN and LSTM models using the same hyperparameters as the best hyperparameters generated in the previous four training scenarios, namely batch size of 32, the number of epochs of 3, learning rate of 1e-5. According to the results in Table 8, the SVM model has the highest accuracy validation value, with an accuracy validation of 89.3%. However, these results are still lower when compared to the accuracy validation value of the AbBERT model.

Next, the confusion matrix results of the AbBERT model with an accuracy validation value of 90.8% can be seen in Fig. 4. Based on the results in the confusion matrix, it can be seen that of the 900 data used as validation data, there are 817 or about 91% of the data that is predicted correctly. So it can be

said that the model that has been built can classify validation data into each category quite well. In addition, from the confusion matrix, it can also be seen that there are 83 or around 9% of data that are predicted to be less in accordance with the category label.

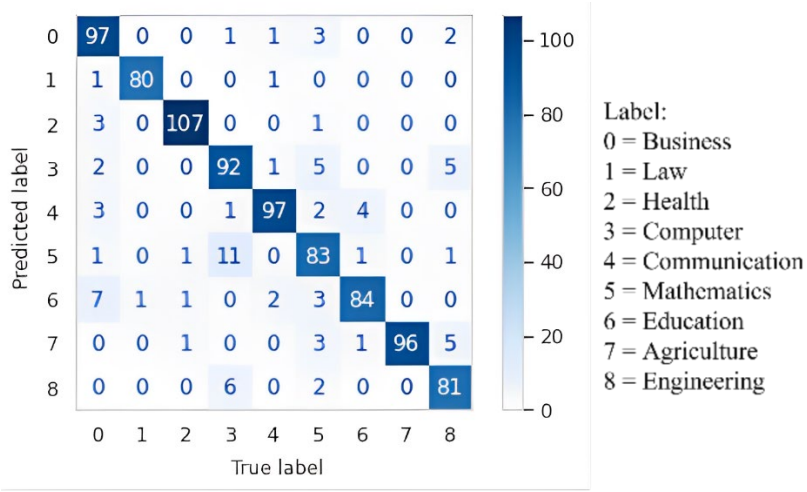

Fig. 4. AbBERT model confusion matrix results

In this research, the labeling of validation data that has been carried out by experts is then poured into a confusion matrix as shown in Fig. 5.

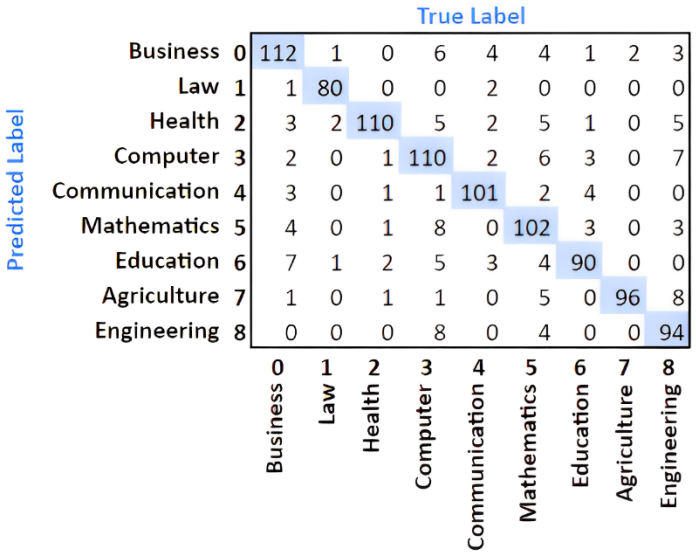

Fig. 5. Confusion matrix results by experts

Fig. 5 shows that there is abstract data that has a relevance between one category and another. This can be seen in the confusion matrix, where the sum exceeds the amount of validation data given, so it can be said that there is data that is categorized with one category and or more than one category by experts. Then when compared with the confusion matrix generated by the AbBERT model as shown in Fig. 4, there are similarities in the off-diagonal part. Although in the confusion matrix by the expert there are more data categorized outside the diagonal compared to the confusion matrix by the AbBERT model, there is a possibility that the data is predicted by the model as a category within the diagonal line. This cannot be said to be wrong, because the data outside the diagonal is likely to be abstract with multidisciplinary categories. This shows that the AbBERT model is not completely wrong in predicting 83 abstract data or 9% of the data mentioned earlier. This is because there is a relationship between the categories of science in an abstract that is tested. So that the AbBERT model will display predictions according to the more dominant category.

However, there are still 4 abstract data that are not predicted correctly by the model. This can be seen in the Computer and Education categories. Where there are 11 abstract data predicted as Mathematics in the Computer category, while based on labeling by experts, there are only 8 abstracts that can be categorized as Mathematics. Then, in the Education category there is 1 data predicted as Agriculture, while in the confusion matrix by experts, there is no data that can be categorized as Agriculture in the Education category. So it can be said that there are only 4 data predicted incorrectly by the model, or it can be concluded that the highest accuracy value of the AbBERT model is 99.56%. Prediction errors can occur due to words that contextually have the same meaning but are also found in several different categories, or because there are no words in the abstract that have information values that match the category to be used as a reference by the model in classification.

This research produces the AbBERT model, which is a development of the pre-trained BERT model combined with the ANN deep learning model. Fine-tuning hyperparameters has been done on the AbBERT model so that the model can produce more optimal accuracy values for text classification in Indonesian natural language processing. In this case the AbBERT model can be used to solve other problems related to text classification in Indonesian language to be even better.

3.3. Results of Model Implementation into Prototype and New Abstract Data Testing

In this research, the AbBERT model was successfully implemented into a prototype to classify new abstract data. In the prototype there is an input form that is used to input the abstract text that you want to classify. The abstract is processed into the text preprocessing stage which consists of lowercase text, text cleaning, tokenizing, and stopword removal. After text preprocessing, the abstract will be forwarded to the AbBERT model or the best model that was previously stored. The model will process the abstract to be classified into disciplinary categories that correspond to the nine predetermined categories. An example of the abstract classification results using the prototype is shown in Fig. 6.

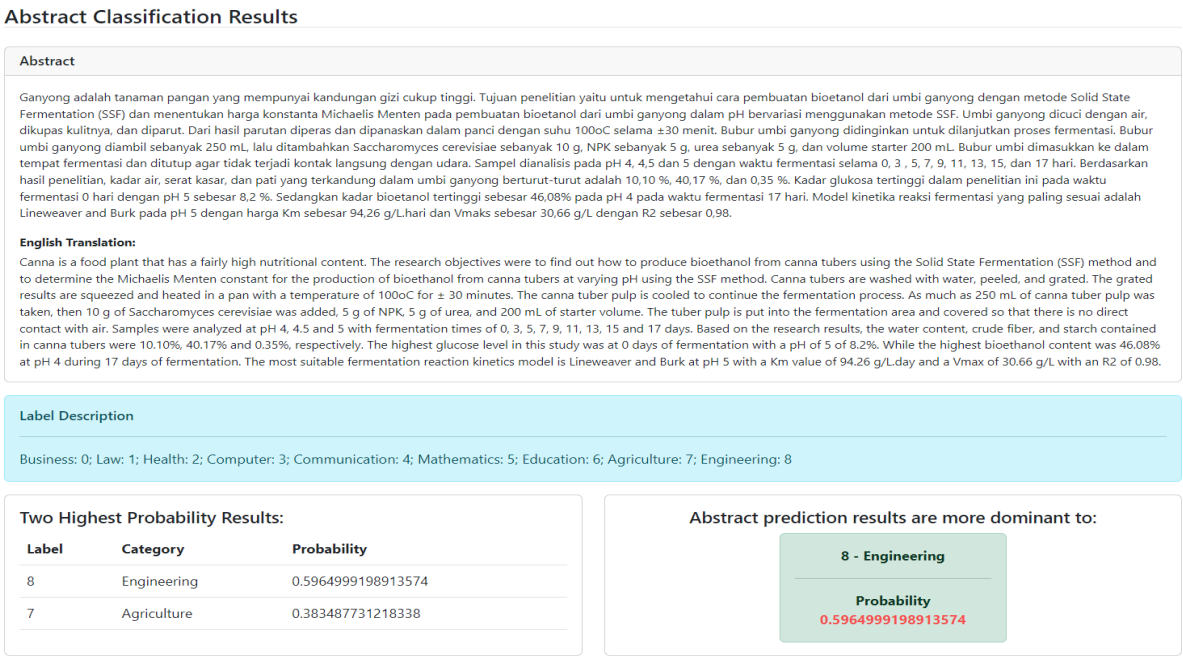

Fig. 6. An example of test results with new abstract data

The results of the abstract classification are displayed in the form of the abstract text, the two highest probability results based on the nine predetermined categories, then the prediction results of the category of the most dominant abstract. Thus, based on the results displayed, it can be determined the tendency of a scientific article to the discipline and or research object. The two highest probability results displayed can see the possibility of a scientific article belonging to a multidisciplinary scientific article.

A total of 90 new abstract data that has never been trained or recognized is used as input to test the AbBERT model. Based on the results obtained, it is possible to conclude that the model developed has

a high level of accuracy. This is evidenced by testing using 90 new abstract data getting prediction results in accordance with each discipline as much as 86 abstract data. There are 4 abstracts that get different prediction results from the specified discipline category. In this case it does not mean that the model used has inaccurate accuracy, but in this case, it shows that the model built can recognize or find out new abstract data that is classified as having a higher probability in accordance with the scientific discipline and / or object of research. Therefore, it can be said that the abstract covers multidisciplinary or belongs to a multidisciplinary scientific article.

4. Conclusion

Classification of Indonesian scientific articles based on abstracts was successfully carried out. The classification successfully displayed the prediction probability value of classification for each category, especially the two categories with the highest probability value. The results found that the most suitable hyperparameter combination consisted of a batch size of 32, a number of epochs of 3, a learning rate of 1e-5, and a ratio of training and test data of 9:1. The model with this hyperparameter combination resulted in an accuracy validation value of 90.8% and was then named the AbBERT (Abstract BERT) model. When compared to other models such as Naive Bayes, SVM, Logistic Regression, KNN, Random Forest, CNN, and LSTM, the AbBERT model built has evaluation results with higher accuracy validation values. The confusion matrix generated by the AbBERT model is then compared with the confusion matrix created based on the labeling of validation data by experts. Based on the analysis, it is known that the AbBERT model is not completely wrong in predicting data outside the diagonal line of the confusion matrix. This is due to the relationship between the categories of science fields in an abstract that is tested. The highest accuracy value that the AbBERT model can get in this case is 99.56%.

The implementation of the AbBERT model into a software prototype was successful. In this prototype, new abstract data that has never been trained or recognized can be used as input to be classified based on its category. The result of the classification will be displayed not only the highest prediction of a category, but also the two highest classification prediction probability values based on the specified category. However, because the prototype is built with an Indonesian-based model, so that in the prototype the abstracts of scientific articles that can be classified are only Indonesian abstracts. Therefore, in future research, a prototype should be built with a model that has been trained with many languages so that the prototype can classify abstracts in various languages. The number of datasets and categories used can be enriched and more varied so that the training process carried out by the model is even better. In addition, in future research, the addition of hidden layers and a more varied number of neurons can be tested in order to find out the comparison of the accuracy validation results obtained by the model.

Acknowledgment

The authors would like to thank the Department of Information Technology, Gunadarma University, which has supported this research.

Declarations

Author contribution. All authors contributed equally to the paper's main contributor. The final paper was read and approved by all authors.
Funding statement. The funding agency should be spelled out completely, followed by the grant number in square brackets and the year.
Conflict of interest. The authors declare that they have no conflicts of interest.
Additional information. There is no additional information for this paper.

References

[1] F. R. Lumbanraja, E. Fitri, Ardiansyah, A. Junaidi, and R. Prabowo, "Abstract Classification Using Support Vector Machine Algorithm (Case Study: Abstract in a Computer Science Journal)," *J. Phys. Conf. Ser.*, vol. 1751, no. 1, pp. 0–12, 2021, doi: 10.1088/1742-6596/1751/1/012042.

[2] A. KP and J. Anitha, "Plant disease classification using deep learning," in *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*, May 2021, pp. 407–411, doi: 10.1109/ICSPC51351.2021.9451696.

[3] I. N. Khasanah, "Sentiment Classification Using fastText Embedding and Deep Learning Model," *Procedia CIRP*, vol. 189, pp. 343–350, 2021, doi: 10.1016/j.procs.2021.05.103.

[4] I. M. Fadhil and Y. Sibaroni, "Topic Classification in Indonesian-language Tweets using Fast-Text Feature Expansion with Support Vector Machine (SVM)," in *2022 International Conference on Data Science and Its Applications (ICoDSA)*, Jul. 2022, pp. 214–219, doi: 10.1109/ICoDSA55874.2022.9862899.

[5] R. Adipradana, B. P. Nayoga, R. Suryadi, and D. Suhartono, "Hoax analyzer for Indonesian news using RNNs with fasttext and glove embeddings," *Bull. Electr. Eng. Informatics*, vol. 10, no. 4, pp. 2130–2136, Aug. 2021, doi: 10.11591/eei.v10i4.2956.

[6] R. Kusumaningrum, I. Z. Nisa, R. P. Nawangsari, and A. Wibowo, "Sentiment analysis of Indonesian hotel reviews: from classical machine learning to deep learning," *Int. J. Adv. Intell. Informatics*, vol. 7, no. 3, pp. 292–303, Nov. 2021, doi: 10.26555/ijain.v7i3.737.

[7] M. S. David and S. Renjith, "Comparison of word embeddings in text classification based on RNN and CNN," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1187, no. 1, p. 012029, Sep. 2021, doi: 10.1088/1757-899X/1187/1/012029.

[8] M. I. Alfarizi, L. Syafaah, and M. Lestandy, "Emotional Text Classification Using TF-IDF (Term Frequency-Inverse Document Frequency) And LSTM (Long Short-Term Memory)," *JUITA J. Inform.*, vol. 10, no. 2, p. 225, Nov. 2022, doi: 10.30595/juita.v10i2.13262.

[9] K. Boonchuay, "Sentiment Classification Using Text Embedding for Thai Teaching Evaluation," *Appl. Mech. Mater.*, vol. 886, pp. 221–226, Jan. 2019, doi: 10.4028/www.scientific.net/AMM.886.221.

[10] J. Devlin, M.-W. Chang, K. Lee, K. T. Google, and A. I. Language, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Naacl-Hlt 2019*, no. Mlm, pp. 1-16, 2019, doi: 10.48550/arXiv.1810.04805.

[11] M. Munikar, S. Shakya, and A. Shrestha, "Fine-grained Sentiment Classification using BERT," *Int. Conf. Artif. Intell. Transform. Bus. Soc. AITB 2019*, vol. 1, pp. 1–5, 2019, doi: 10.1109/AITB48515.2019.8947435.

[12] S. Abdul, Y. Qiang, S. Basit, and W. Ahmad, "Using BERT for Checking the Polarity of Movie Reviews," *Int. J. Comput. Appl.*, vol. 177, no. 21, pp. 37–41, 2019, doi: 10.5120/ijca2019919675.

[13] W. Maharani, "Sentiment Analysis during Jakarta Flood for Emergency Responses and Situational Awareness in Disaster Management using BERT," *2020 8th Int. Conf. Inf. Commun. Technol. ICoICT 2020*, pp. 1–5, 2020, doi: 10.1109/ICoICT49345.2020.9166407.

[14] J. Ravi and S. Kulkarni, "Text embedding techniques for efficient clustering of twitter data," *Evol. Intell.*, pp. 1-11, Feb. 2023, doi: 10.1007/s12065-023-00825-3.

[15] M. Khadhraoui, H. Bellaaj, M. Ben Ammar, H. Hamam, and M. Jmaiel, "Survey of BERT-Base Models for Scientific Text Classification: COVID-19 Case Study," *Appl. Sci.*, vol. 12, no. 6, p. 2891, Mar. 2022, doi: 10.3390/app12062891.

[16] X. Chen, P. Cong, and S. Lv, "A Long-Text Classification Method of Chinese News Based on BERT and CNN," *IEEE Access*, vol. 10, pp. 34046–34057, 2022, doi: 10.1109/ACCESS.2022.3162614.

[17] G. Danilov, T. Ishankulov, K. Kotik, Y. Orlov, M. Shifrin, and A. Potapov, "The Classification of Short Scientific Texts Using Pretrained BERT Model," vol. 281, pp. 83-87, 2021, doi: 10.3233/SHTI210125.

[18] I. M. Rabbimov and S. S. Kobilov, "Multi-Class Text Classification of Uzbek News Articles using Machine Learning," in *Journal of Physics: Conference Series*, May 2020, vol. 1546, no. 1, pp. 012097, doi: 10.1088/1742-6596/1546/1/012097.

[19] A. Bogdanchikov, D. Ayazbayev, and I. Varlamis, "Classification of Scientific Documents in the Kazakh Language Using Deep Neural Networks and a Fusion of Images and Text," *Big Data Cogn. Comput.*, vol. 6, no. 4, p. 123, Oct. 2022, doi: 10.3390/bdcc6040123.

[20] A. Barua, O. Sharif, and M. M. Hoque, "Multi-class Sports News Categorization using Machine Learning Techniques: Resource Creation and Evaluation," *Procedia Comput. Sci.*, vol. 193, pp. 112–121, 2021, doi: 10.1016/j.procs.2021.11.002.

[21] D. Ali, M. M. S. Missen, and M. Husnain, "Multiclass Event Classification from Text," *Sci. Program.*, vol. 2021, pp. 1–15, Jan. 2021, doi: 10.1155/2021/6660651.

[22] 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.

[23] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," *Inf. Process. Manag.*, vol. 50, no. 1, pp. 104–112, Jan. 2014, doi: 10.1016/j.ipm.2013.08.006.

[24] D. Haryalesmana and M. Wieriks, "Indonesian Stopword Corpus," 2016. [Online]. Available at: https://github.com/masdevid/ID-Stopwords

[25] F. Z. Tala, "A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia," Institute for Logic, Language and Computation. Universiteit van Amsterdam, The Netherlands., 2003. [Online]. Available at: https://eprints.illc.uva.nl/id/eprint/740/1/MoL-2003-02.text.pdf.

[26] W. C, "BERT-base-indonesian-522M," *Hugging Face*, 2021. [Online]. Available at: https://huggingface.co/cahya/bert-base-indonesian-522M.

[27] A. Vaswani *et al.*, "Attention is all you need," in *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6000–6010, doi: 10.48550/arXiv.1706.03762.

[28] Y. Bai, "RELU-Function and Derived Function Review," *SHS Web Conf.*, vol. 144, p. 02006, Aug. 2022, doi: 10.1051/shsconf/202214402006.

[29] S. Pothuganti, "Review on over-fitting and under-fitting problems in Machine Learning and solutions," *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.*, vol. 7, pp. 3692–3695, Sep. 2018. [Online]. Available at: http://www.ijareeie.com/upload/2018/september/11A_PS_NC.PDF.

[30] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," *ICT Express*, vol. 6, no. 4, pp. 312–315, Dec. 2020, doi: 10.1016/j.icte.2020.04.010.

[31] H. Jindal, N. Sardana, and R. Mehta, "Analyzing Performance of Deep Learning Techniques for Web Navigation Prediction," *Procedia Comput. Sci.*, vol. 167, pp. 1739–1748, 2020, doi: 10.1016/j.procs.2020.03.384.

[32] M. Hossin and M. N. Sulaiman, "A Review on Evaluation Metrics for Data Classification Evaluations," *Int. J. Data Min. Knowl. Manag. Process*, vol. 5, no. 2, pp. 01–11, Mar. 2015, doi: 10.5121/ijdkp.2015.5201.