JOURNAL OF DATA SCIENCE | Vol.2021:01
eISSN:2805-5160

Utilize Medical Text Data to Estimate Disease Types by Using Naïve Bayes and ANN Classifier

[1]Mujiono Sadikin, [2]DeshintaArrova Dewi, [3]Purwanto S Katijan, [1]Ibrohim Thohari,

[1]Faculty of Computer Science, Universitas Mercu Buana Jakarta, Indonesia, [2]Faculty of Information Technology, INTI International University, Nilai, Malaysia [3]Faculty of Business & Economic Universitas Esa Unggul Jakarta, Indonesia

Email: mujiono.sadikin@mercubuana.ac.id, deshinta.ad@newinti.edu.my, purwanto@esaunggul.ac.id, 41518120090@student.mercubuana.ac.id

Abstract

The primary concept of the hospital is the provision of health services to the community. In many cases, the utilization of information technology to record all hospital activity data can improve hospitals' quality services. However currently, the data is only stored in the database and used as history without further use. Many experiences show that optimizing data usage can greatly assist doctors in making decisions to minimize medical errors. For example, examination data that among others of anamnesis (medical abstract), blood pressure, temperature, and other patient’s symptom data can be used to classify the kind of disease. One of the challenges in medical data utilization is that these data consists of various formats, structured, and unstructured as well. In this study, we address the medical unstructured data format by using Natural Language Processing approach. The combination of its representation results with the structured format data is then used as the dataset to build the model for disease type prediction based on Naïve Bayes and Artificial Neural Network classifier. By using these two algorithms, the results of the classification of the kind of disease. The performed experiments show that the ANN model performs better with the best accuracy average of 89.29% compared to Naive Bayes, which is 80.60 %.

Keywords

Medical abstract, Natural Language Processing, Naïve Bayes, Artificial Neural Network

1. Introduction

The hospital is an important agency in providing health services to the community. In providing services to the community, hospitals must always improve the quality of services to increase public satisfaction with hospital services. Starting from services for emergency services, outpatient care, inpatient care, doctor consultation, drug administration, to doctor reliability must be maximized.

In this era of technology, data plays an important role in a hospital to provide services to the

Submission: 16 August 2021; **Acceptance:** 23 August 2021

community. The hospital records all activity data in the hospital, starting from the medical abstract, examination data, doctor's actions, disease diagnosis, type of disease, up to prescribing drugs. However, currently, all data has not been fully utilized. Data is only stored in the database and used as history without any further data utilization. If all data can be processed properly, it will greatly assist doctors in making decisions to minimize medical errors (P. S. Roshanov *et al.*, 2011). Data Mining is one of the stages of Knowledge Discovery in Databases (KDD). The steps in KDD include data cleansing, data integration, data selection, data transformation, data mining, patterns evaluation, and knowledge presentation (M. Ridwan, H. Suyono, and M. Sarosa, 2013). Data Mining can be used as one of the Decision Support System tools by doctors to assists their tasks. By using certain methods, data mining can provide drug recommendatio ns from medical record data (J. K. Abdul Aziz Priatna, Rani Megasari, 2018). An automatic recommendation will increase the doctor's awareness in making decisions (Guardian Y. Sanjaya, S. Harry, L. Lazuardi, and N. Faizah, 2012).

Anamnesis data is one of the results of examination data performed by doctors on patients in the form of unstructured text medical abstracts. To process the medical abstract data, it is necessary to conduct the text mining processing by using Natural-Language Processing (NLP) (N. Indrawati, 2010). With Natural-Language Processing (NLP), medical abstract data can be processed as the input of data mining techniques (T. F. M. Raj and S. Prasanna, 2013).

In this study, we examine Naïve Bayes and Artificial Neural Network classifier to predict the type of disease classification suffered by patients based on the results of a doctor's examination. Whereas to represent the unstructured form of medical abstract data into numerical format can be feed into these classifiers, we address it by using Natural Language Processing. The types of disease predicted are "acute" and "chronic". The disease is called acute if it is temporary and can be cured after receiving treatment, whereas the chronic illness is a lung disease, recurring, requires a long and well-organized treatment process, and needs the ability to limit a person's lifestyle (J. Olamaei and S. Ashouri, 2015). Three main stages carried out in this research are 1) Data cleaning which includes removing the noise from the dataset and filling in the blank values; 2) Representation of medical abstracts into vector form by using the Word2Vec word embedding method (B. S. Prakoso, et al. 2019). The Word2Vec library is adopted from the gensim python library (Y. D. Prabowo, T. L. Marselino, and M. Suryawiguna, 2019); 3) The classification stage that uses Naïve Bayes and Artificial Neural Network model. The accuracy value is used as the performance parameter evaluation.

In this study, we also examine whether the involvement of unstructured data from the dataset in word2vec training affects the accuracy performance. Thus, the experiment in this study consists of two main scenarios based on the involvement or un-involvement of the unstructured medical abstract text data.

2. Research Method

In this research, the classification technique Naïve Bayes and Artificial Neural Network were used by using the programming language Python to run the algorithm. Broadly speaking, the research stages include data collection, training the model Word2Vec, preprocessing data, representing data medical abstract into a vector form length N, data testing, and implement ing the Naïve Bayes algorithm and Artificial Neural Network. The data collection stage, obtained in the dataset was the patient examination form of Medical Abstract, systolic blood pressure, blood pressure diastolic, temperature, pulse, age, gender, giddiness, and type of disease. The stages training data of Word2Vec resulted in an NLP model used to convert Medical Abstract data. In the stages of representing the medical abstract, the selected keywords are generated into vectors form. At the algorithm implementation stage, results of the testing were obtained for the Naïve Bayes algorithm and Artificial Neural Network.

The process of representing medical abstracts into vector data requires the Word2Vec model to mapping keywords into vector data. However, some of the keywords of medical abstracts are not found in the Word2Vec model. It is happening because there is an inconsistent input of the medical abstract when the doctor takes the examination. To avoid keywords missing, the study was conducted using two scenarios for training the model Word2Vec. Figure 1 shows a diagram of the research stages in the first scenario. The process training model Word2Vec's first scenario only uses the corpus Wikipedia as a dataset. However, if the medical abstract keyword is not found in the Word2Vec model, an update is carried out to the Word2Vec model by adding the keyword to the model and then retrain the model. That way the model can accommodate inconsistencies of medical abstract input. Figure 2 shows a diagram of the research stages in the second scenario. The process Word2Vec model training the second scenario does not only use the corpus Wikipedia as a dataset. But the corpus Wikipedia is combined with a medical abstract as a dataset. In this way, the model Word2Vec accommodates inconsistencies of medical abstract input.

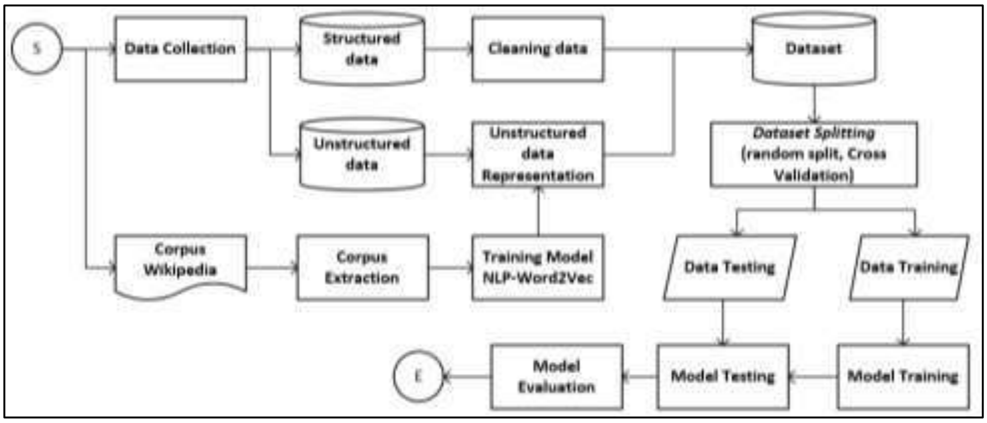


Figure 1 Research Stages of the First Scenario

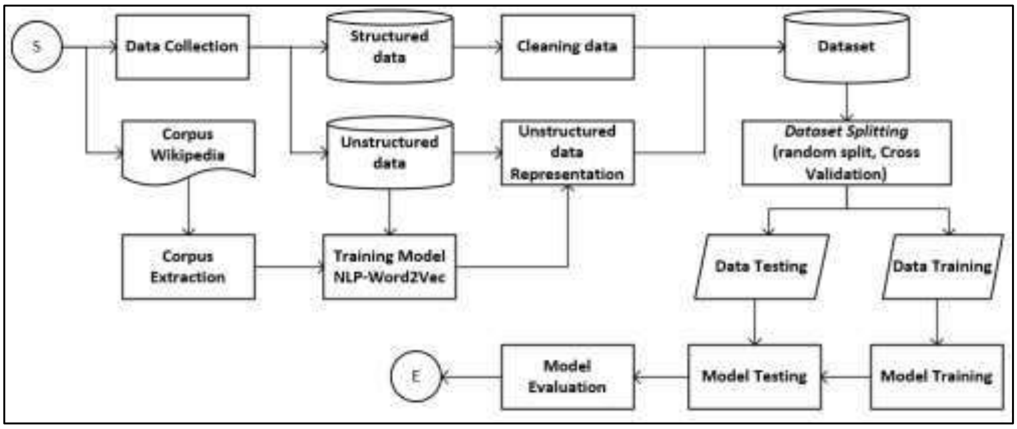


Figure 2 Research Stages of the Second Scenario

Data Collection

Data collection stages are carried out by conducting observations and interviews. From the results of observations and interviews, it was found that there was no use of doctor's examination data to serve as a Decision Support System in the form of a classification of disease types. Data is only stored and used as patient history so it does not provide added value in the form of more useful information. There were 4,404 records of examination data collected. Then the examination data is separated between unstructured data and structured data because the treatment of these two types of data is different. Unstructured data includes anamnesis while the data structured include blood systole pressure, blood diastolic pressure, temperature, pulse, age, gender, unsteady condition, and type of disease.

Data Cleansing

In the data cleansing step, we remove any noise from the dataset. The instant data which have empty values are re-evaluated to determine whether the data is suitable for use or not. The instant data that has an empty value within a certain tolerance can still be used. By using a common technique, the treatment of swapping the blank value with the average value recognized from these variables value (W. I and S. S. U. Rahman S, 2015). This average value is used as a constant for replacing the empty values variable dataset regardless of the relationship between properties that affect the Data Mining algorithm used. At this stage, the data being cleaned is structured data which includes the variables Calm, Anxious, Age, Sex, Systol, Dyastol, Temperature, Pulse, Weight, Height, Unsteady condition, Risk of falling, Type of disease.

Conversion of Text Data

The conversion of Medical Abstract data into numeric vector intended for the data can be processed in Data Mining. The method used in this process is Word2Vec word embedding. Word2Vec can understand the meaning and turn it into a vector of words in the document based on the hypothesis that words that have similar meanings have a proximity vector (Irwan budiman, M. R. Faisal, and D. T. Nugrahadi, 2020). The Word2Vec can be performed by two alternatives models, namely the Continuous Bag-of-Word (CBOW) and Skip-Gram. Each model has several layers, i.e. the input layer, projection layer, and output layer. Architectura lly, both models have architectures that are opposite to each other. Skip-gram is used to predict the context of a word as input. While CBOW is used to predict words from the surrounding context as input. Figure 3 is the architecture CBOW and Skip-gram proposed by(Milokov B. Jang, I. Kim, and J. W. Kim, 2019).

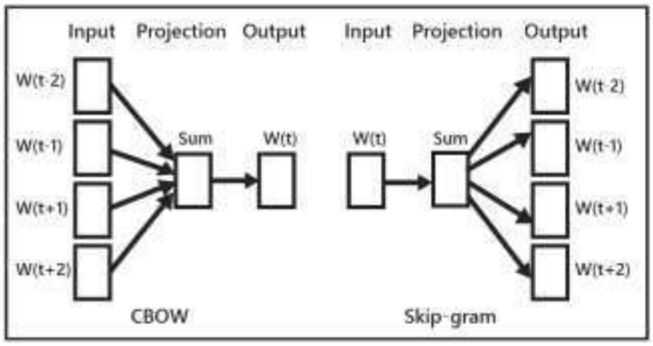


Figure 3 CBOW and Skip-Gram Architecture

The process of medical abstract data representation into the numerical vector by using the method word embedding requires a word2vec model training data. To generate the model, it is required a large amount of text in Indonesian. For this purpose, we use Wikipedia Corpus which is a collection of articles from Wikipedia to train the Word2Vec model. In this study, the Corpus Wikipedia contains 408,952 articles in Indonesian which were extracted using wiki corpus. The Extraction Process of the Wikipedia Corpus is carried out to convert the Wikipedia corpus into text.

The next step is to create a model Word2Vec that is used to map the semantic proximity positions between words from an input text. In this study, we use two scenarios based on the involvement or un-involvement of the test dataset in the Word2Vec training phase. Both of these scenarios were carried out because there was an inconsistency of medical abstract input which resulted in the keyword not being found in the model Word2Vec.

First Scenario: In the first scenario, we only use the corpus Wikipedia to train the Word2Vec model. However, if the keyword medical abstract is not found in the Word2Vec model, update the model Word2Vec performed by adding keywords to the model and then retraining the model. Models updates are performed while the representation process is medical abstract running. So if the gensim does not find the keyword in the model, the representation processing time will be longer because it requires additional time for model training.

Second Scenario: In the second scenario of the Word2Vec model training process, we do not only use the corpus Wikipedia as a dataset. But we combine the corpus Wikipedia with a medical abstract dataset. Medical abstracts are treated with simply preprocessing to eliminate the character of numbers and symbols. Then the medical abstract is added to the Wikipedia corpus for training the model Word2Vec. In this way, the models created already accommodate input medical abstract inconsistencies.

In the representation process for each scenario, we performed cleaning text for the Medical Abstract data. The process of cleaning text data uses a library of Indonesian Natural Language Processing in python. The cleaning process for the Medical Abstract data text includes Lemmatization, Removing number, stopword removal, and Pos tagger.

Step 1: We use lemmatization to transform words into the root of words. Lemmatizat ion changes the word by considering the context of the word, which means that it is not just removing some characters in a word. So that the resulting word is more accurate and has meaning.

Step 2: The next step is removing numbers, which it aims to remove numbers from text data. Numbers are omitted because they don't have much effect on the root word.

Step 3: Stopword removal is used to eliminate common words that are considered meaningless. Examples of stopwords in Indonesian are "yang", "dan", "di", "dari", etc. That way the process can be focused on words that are considered important.

Step 4: POS Tagging is used to get Part-Of-Speech tags from text which is useful for categorizing word classes. Then the text is separated based on the word class category.

After the text data has been cleaned, it can be represented as vector data using two experimenta l scenarios of the word2vec model. Table 1 shows the samples of representation of medical abstract data into numerical vector data.

TABLE 1 VECTOR DATA REPRESENTATION

Text	V1	V2	….	V25
batuk	-0.0045260135	0.01768395	….	0.00079621444
dahak	0.017375236	0.005819824	….	-0.014331724
nafas	0.011860242	0.014599305	….	-0.015677009
demam	0.0043433174	0.0083342595	….	-0.0074927625

Medical abstracts that have been represented as vectors and structured data that have been cleaned are combined back into a dataset. The dataset is used to implement data mining. In this research, the algorithm used is the Naïve Bayes algorithm and the Artificial Neural Network algorithm. In each algorithm, there are three stages, namely training, testing, and evaluating.

Dataset Splitting

To carry out the training and testing process, the dataset is split into training data and testing. For the splitting purpose, we use two split data techniques namely Random Split and Kfold Cross-Validation. Random split divides the dataset randomly into training data and testing data with certain comparisons. K-fold Cross Validation is used to split the data into K parts of the dataset to the same size to eliminate bias in the data. The training and testing process is carried out as much as the specified K (F. Tempola, M. Muhammad, and A. Khairan, 2018). Figure 4 shows the iteration process in the K Fold Cross-validation method.

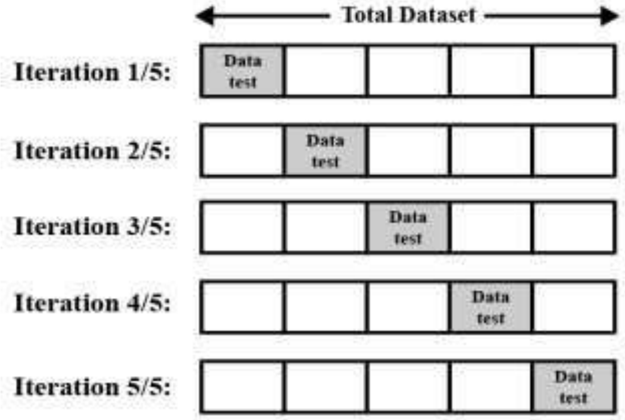


Figure 1 Cross-Validation with 5 K-fold value

In this study, two methods were split data carried out with the same comparison for each algorithm and the Word2Vec model training scenario. From a total of 3108 rows of data, three comparisons of the random split were made between the training data: testing data, namely 9:1, 8:2, and 7:3. Whereas, in Cross-Validation Kfold there are 3 grades K is 5 K, 10 K, and 15 K.

A. *Naïve Bayes Algorithm*

The naive Bayes algorithm is an algorithm that uses probabilistic and statistical models invented by the British scientist Thomas Bayes (F. E. Prabowo and A. Kodar, 2019). Classifications are carried out by predicting future opportunities based on past experiences. This algorithm aims to predict a class based on the training data provided. Figure 5 is a general form of the Naïve Bayes formula. Figure 5 is the general form of the formula Naïve Bayes.

$$P(y|X) = \frac{P(X|y) \, . \, P(y)}{P(X)}$$

Description:

- X : Data with unknown class

- y : Hypothesis class data X is a specific class
- P(y|X) : Hypothesis probability y based on condition X
- P(y) : Hypothesis probability y
- P(X|y) : Probability X based on these conditions
- P(X) : Probability of X

B. *Artificial Neural Network*

This algorithm is a deep learning algorithm inspired by the human brain network and implemented into a computer program (A. S. Kurniawansyah, 2018). The Artificial Neural Network is running based on a reasoning model of the human brain which can complete several calculation processes during the learning process. Like the human brain network, an Artific ia l Neural Network has neurons consisting of several simple interconnected processors. Neurons are connected by weight pass signals from one neuron to another neuron. Hidden layers and output layers have an additional input called bias. Figure 5 is an example of the architecture of an Artificial Neural Network.

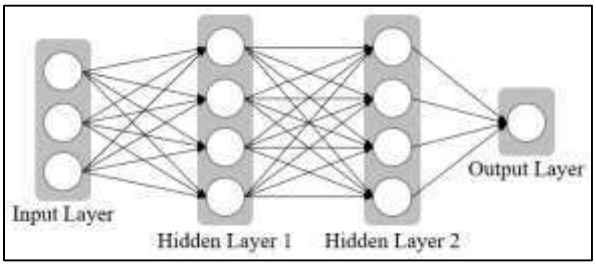


Figure 5 Architecture of Artificial Neural Network

The architecture above is commonly referred to as Multi-Layer Perceptron (MLP) or Fully Connected Layer. The architecture above has four layers that are input layer with 3 neurons, the hidden layer 1 with 4 neurons, the hidden layer 2 with 4 neurons, and an output layer with 1 output node. So that in the architecture above there are 3x4 weight + 4 bias, 4x4 weight + 4 bias, and 4x1 weight + 1 bias with a total of 41 parameters used.

In the study, we used the Keras library to build the ANN models. The built architectura l model consists of 110 nodes on the input layer. Whereas, the number of layers created was 4 hidden layers with the activation of ReLU (Rectified Linear Unit activation function) and 1 output layer with the activation of the sigmoid function. The others ANN parameters chosen are adam optimizer, binary_crossenthropy as variable loss, and accuracy as the evaluatio n parameter. Whereas in the training phase we use 150 epochs (period), the batch_size is 10.

3. Results and Discussion

To represent the Medical Abstract text data into numerical vectors we used Natural Language Processing model. The NLP Model training data is generated from a total of 408,952 Indonesian language articles collected from the Corpus Wikipedia which are extracted using wiki corpus. The modeling of Word2Vec is carried out by two scenarios. Both of these scenarios are carried out to overcome inconsistencies in the input medical abstract.

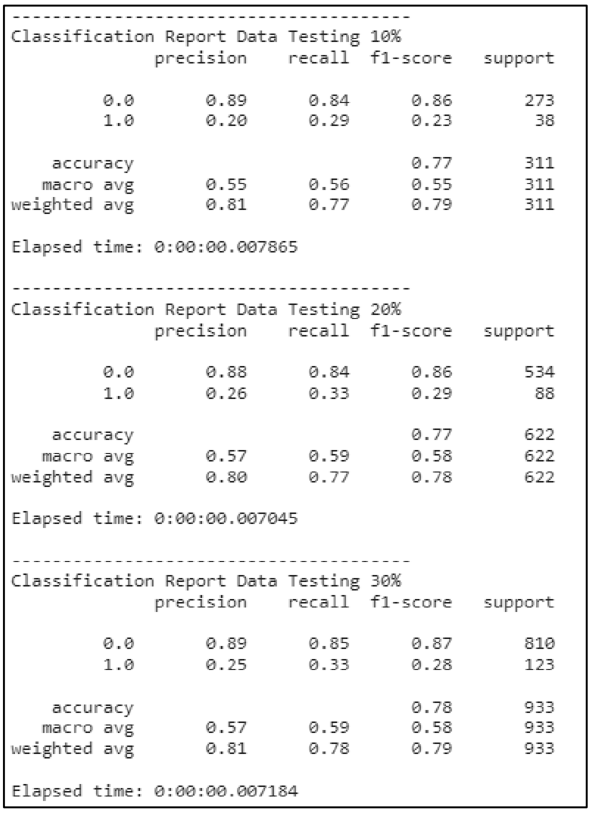
The Process for training the model Word2Vec in the first scenario uses the corpus Wikipedia as training data only. The process of extracting articles was carried out for 9 minutes 47 seconds and produced a 791 MB txt file. The next stage is to train the txt data that has been created using the Word2Vec gensim library. The Process training data is carried out for 13 minutes and produces 3 files model training data size 25 vectors.

The second scenario of the model training process Word2Vec is not only to use the corpus Wikipedia as a dataset but also to combine Wikipedia corpus with a medical abstract as the training data. Before the medical abstract is combined with the Wikipedia corpus, it was performed a simple preprocessing, such as Lemmatization and eliminating numeric characters and symbols into a dataset. The process of extracting articles and medical abstraction that was carried out produced a txt file measuring 791 MB which is extracted in 1 hour 28 minutes and 1 second. The next stage is to train the data txt that has been created using the Word2Vec gensim library. The Process training data is carried out for 12 minutes 45 seconds and produces 3 files model training data size 25 vector.

The result of the Word2Vec training process is numerical vector representation which then is used to generate the dataset for the classifier model. The dataset is generated to combine the numerical vector as a representation of the medical abstract text with the other numerica l attributes data. In this study, the representation process was carried out by taking 4 keywords from the data Medical Abstract. Keywords are represented using two model scenarios that have been made. Each model scenario size is 25 data vectors semantic proximity position keyword input. So that from one record Medical Abstract obtained 100 vectors data for each scenario. By combining the vectors with the other 10 attributes data, we get 110 attributes of each instant data.

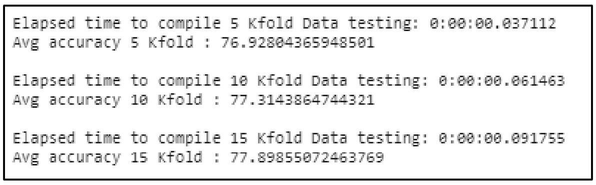
C. *Naïve Bayes Algorithm in First Scenario*

In the implementation of the first scenario of the Naïve Bayes algorithm, we use two dataset splitting techniques i.e. Random Split and Kfold Cross-Validation. In the Random Split, the data is divided into training data and testing data randomly with the training data and testing data proportions as 90%:10%; 80%:20%; and 90%:10%. Figure 6 depicts the performance test of the first scenario of the Naïve Bayes classifier model. In the first scenario, the accuracy performances of the model achieved in the testing phase are consecutively 77% of 10% testing data, 77% of 20% testing data, and 78% of 30% testing data. Whereas the first scenario of the Naïve Bayes by using the method Kfold Cross-Validation The dataset is divided into several K values. Figure 7 shows the accuracy performance testing phase of the Naïve Bayes Algorit hm model by using the first scenario experiment and Kfold Cross-Validation as data splitting technique. We use three Ks: 5, 10, and 15 as well. Those three Ks provide the accuracy performance of 76.92% for 5K, 77.31% for 10 K, and 77.89% for the15K.



```
---------------------------------------
Classification Report Data Testing 10%
              precision    recall  f1-score   support

         0.0       0.89      0.84      0.86       273
         1.0       0.20      0.29      0.23        38

    accuracy                           0.77       311
   macro avg       0.55      0.56      0.55       311
weighted avg       0.81      0.77      0.79       311

Elapsed time: 0:00:00.007865

---------------------------------------
Classification Report Data Testing 20%
              precision    recall  f1-score   support

         0.0       0.88      0.84      0.86       534
         1.0       0.26      0.33      0.29        88

    accuracy                           0.77       622
   macro avg       0.57      0.59      0.58       622
weighted avg       0.80      0.77      0.78       622

Elapsed time: 0:00:00.007045

---------------------------------------
Classification Report Data Testing 30%
              precision    recall  f1-score   support

         0.0       0.89      0.85      0.87       810
         1.0       0.25      0.33      0.28       123

    accuracy                           0.78       933
   macro avg       0.57      0.59      0.58       933
weighted avg       0.81      0.78      0.79       933

Elapsed time: 0:00:00.007184
```

Figure 6 Naive Bayes First Scenario Random Split

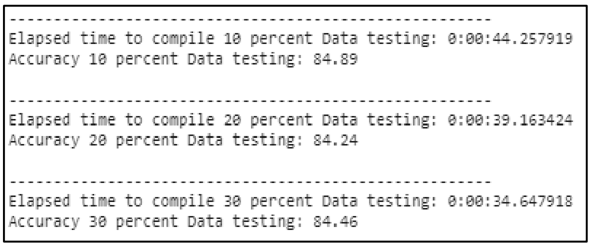


```
Elapsed time to compile 5 Kfold Data testing: 0:00:00.037112
Avg accuracy 5 Kfold : 76.92804365948501

Elapsed time to compile 10 Kfold Data testing: 0:00:00.061463
Avg accuracy 10 Kfold : 77.3143864744321

Elapsed time to compile 15 Kfold Data testing: 0:00:00.091755
Avg accuracy 15 Kfold : 77.89855072463769
```

Figure 7 Naive Bayes First Scenario Cross-Validation

Artificial Neural Network First Scenario

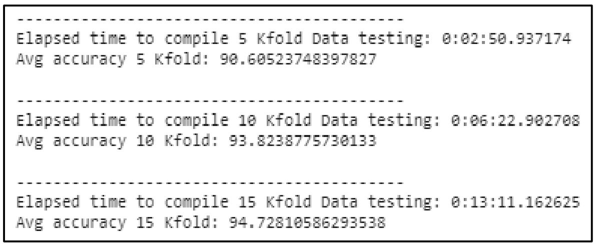
In the implementation of the algorithm in Artificial Neural Network the first scenario, there are two data split methods used, namely Random Split and Kfold Cross-Validation. In Random Split, the dataset is divide into training and data testing data. Experiments of the model Artificial Neural Network with the method were Random Split carried out in 3 tests, namely for testing data 10% of the dataset, 20% of the dataset, and 30% of the dataset. Figure 8 is the performance test result of the model Artificial Neural Network in the first scenario. From this experiment, the accuracy model is 68.49% for testing data 10%, 73.47% for testing data 20%, and 72.56% for testing data 30%. Figure 8 shows the results of the Artificial Neural Network first scenario random split method.



```
---------------------------------------------------
Elapsed time to compile 10 percent Data testing: 0:00:44.257919
Accuracy 10 percent Data testing: 84.89

---------------------------------------------------
Elapsed time to compile 20 percent Data testing: 0:00:39.163424
Accuracy 20 percent Data testing: 84.24

---------------------------------------------------
Elapsed time to compile 30 percent Data testing: 0:00:34.647918
Accuracy 30 percent Data testing: 84.46
```

Figure 8 ANN First Scenario Random Split

Implementation of Artificial Neural Network the first scenario of using the method Kfold Cross-Validation The dataset is divided into several K values. Split data is done 3 times that among others, with a value of 5 K, 10 K, and 15K. Figure 9 is the performance test result of the model Artificial Neural Network the first scenario with Kfold Cross-Validation. Of testing with each value, Kfold showed performance accuracy models Artificial Neural Network of 59.68% for the 5 K, 64.02% to 10 K, and 61.27% for the15K. Figure 9 shows the results of the Artificial Neural Network first scenario Kfold Cross-Validation method.

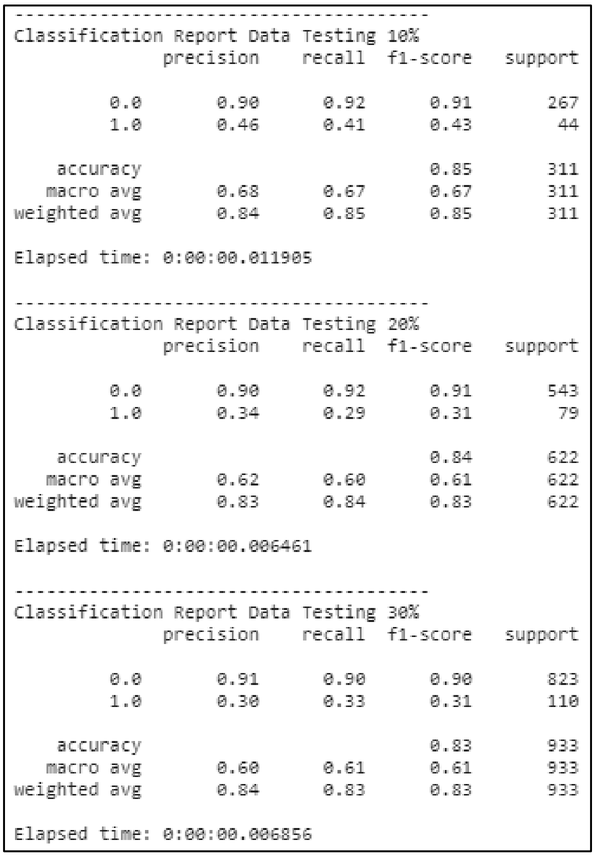


```
-----------------------------------------
Elapsed time to compile 5 Kfold Data testing: 0:02:50.937174
Avg accuracy 5 Kfold: 90.60523748397827

-----------------------------------------
Elapsed time to compile 10 Kfold Data testing: 0:06:22.902708
Avg accuracy 10 Kfold: 93.8238775730133

-----------------------------------------
Elapsed time to compile 15 Kfold Data testing: 0:13:11.162625
Avg accuracy 15 Kfold: 94.72810586293538
```

Figure 9 ANN First Scenario Cross-Validation

Naïve Bayes Algorithm Second Scenario

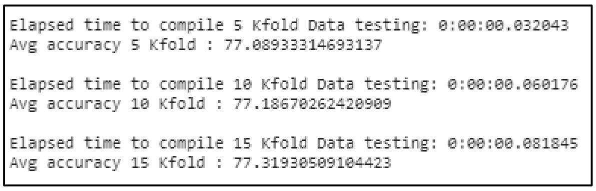
Like in the first scenario, the implementation of the algorithm in Naïve Bayes the second scenario, there are two data split methods used, namely Random Split and Kfold Cross- Validation. In Random Split, the data is divided into training data and testing data as many as three comparisons, namely testing data 10% from the dataset, 20% from the dataset, and 30% from the dataset. Figure 10 is the performance test result of the second scenario Naïve Bayes Algorithm model. In the second scenario, the model performance obtained in this test is 85% accuracy for testing data 10% of the dataset, 84% accuracy for testing data 20% of the dataset, and 83% accuracy for testing data of 30% of the dataset. Figure 10 shows the results of the Naive Bayes second scenario random split method.



```
--------------------------------------
Classification Report Data Testing 10%
              precision    recall  f1-score   support

         0.0       0.90      0.92      0.91       267
         1.0       0.46      0.41      0.43        44

    accuracy                           0.85       311
   macro avg       0.68      0.67      0.67       311
weighted avg       0.84      0.85      0.85       311

Elapsed time: 0:00:00.011905

--------------------------------------
Classification Report Data Testing 20%
              precision    recall  f1-score   support

         0.0       0.90      0.92      0.91       543
         1.0       0.34      0.29      0.31        79

    accuracy                           0.84       622
   macro avg       0.62      0.60      0.61       622
weighted avg       0.83      0.84      0.83       622

Elapsed time: 0:00:00.006461

--------------------------------------
Classification Report Data Testing 30%
              precision    recall  f1-score   support

         0.0       0.91      0.90      0.90       823
         1.0       0.30      0.33      0.31       110

    accuracy                           0.83       933
   macro avg       0.60      0.61      0.61       933
weighted avg       0.84      0.83      0.83       933

Elapsed time: 0:00:00.006856
```

Figure 10 Naive Bayes Second Scenario Random Split

On the method Kfold Cross-Validation The dataset is divided into several K values. Then iteration is carried out to calculate the average accuracy value. Figure 11 is the result of the performance test of the Naïve Bayes Algorithm model in the second scenario with the method Kfold Cross-Validation. There are three grades K used is 5 K, 10 K, and 15K. From the three values Kfold, the performance is accuracy model77.08% for 5K, 77.18% for 10 K, and 77.31% for the 15K. Figure 11 shows the results of the Naive Bayes second scenario Kfold Cross-Validation method.



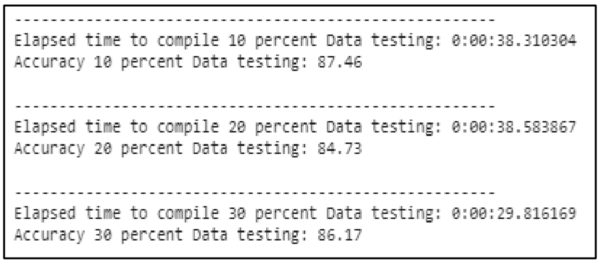
```
Elapsed time to compile 5 Kfold Data testing: 0:00:00.032043
Avg accuracy 5 Kfold : 77.08933314693137

Elapsed time to compile 10 Kfold Data testing: 0:00:00.060176
Avg accuracy 10 Kfold : 77.18670262420909

Elapsed time to compile 15 Kfold Data testing: 0:00:00.081845
Avg accuracy 15 Kfold : 77.31930509104423
```

Figure 11 Naive Bayes Second Scenario Kfold Cross-Validation

Artificial Neural Network Second Scenario

Like in the first scenario, the implementation of the algorithm in Artificial Neural Network the second scenario uses two data split methods, namely Random Split and Kfold Cross-Validation. In Random Split, the dataset is split into 3 comparisons, namely for testing data 10% of the dataset, 20% of the dataset, and 30% of the dataset. Figure 12 is the performance test result of the model Artificial Neural Network in the second scenario. From this experiment, the accuracy model is 87.46% for testing data 10%, 84.73% for testing data 20%, and 86.17%for testing data 30%.

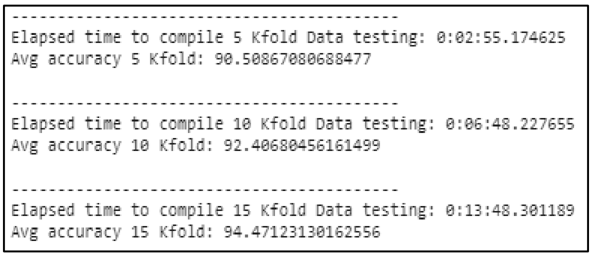
Figure 12 shows the results of the Artificial Neural Network second scenario random split method.



```
----------------------------------------------------
Elapsed time to compile 10 percent Data testing: 0:00:38.310304
Accuracy 10 percent Data testing: 87.46

----------------------------------------------------
Elapsed time to compile 20 percent Data testing: 0:00:38.583867
Accuracy 20 percent Data testing: 84.73

----------------------------------------------------
Elapsed time to compile 30 percent Data testing: 0:00:29.816169
Accuracy 30 percent Data testing: 86.17
```

Figure 12 NN Second Scenario Random Split

Implementation of Artificial Neural Network the second scenario of using the method Kfold Cross-Validation The dataset is divided into several K values. Split data is done 3 times that among others, with a value of 5 K, 10 K, and 15K. Figure 9 is the performance test result of the model Artificial Neural Network the first scenario with Kfold Cross-Validation. Of testing, with each value, Kfold showed performance accuracy models Artificial Neural Network of 90.5% for the 5 K, 92.4% to 10 K, and 94.47% for the15K. Figure 13 shows the results of the Artificial Neural Network second scenario Kfold Cross-Validation method.



```
------------------------------------------
Elapsed time to compile 5 Kfold Data testing: 0:02:55.174625
Avg accuracy 5 Kfold: 90.50867080688477

------------------------------------------
Elapsed time to compile 10 Kfold Data testing: 0:06:48.227655
Avg accuracy 10 Kfold: 92.40680456161499

------------------------------------------
Elapsed time to compile 15 Kfold Data testing: 0:13:48.301189
Avg accuracy 15 Kfold: 94.47123130162556
```

Figure 13 ANN Second Scenario Cross-Validation

Scenario Comparison

Scenario comparison is done to determine the effect of the model Natural Language Processing in performing representations medical abstract on the accuracy of the algorithm Naïve Bayes and Artificial Neural Network. In the first scenario, the representation processing time is medical abstract 3108 records the data becomes vector data for 44 minutes 50 seconds. The implementation of the algorithm with the Naïve Bayes random split method obtained an average performance accuracy model of 77.3% with an average test duration of 0.0073 seconds. Meanwhile, the Artificial Neural Network with the random split method obtained an average performance accuracy of 84.53% with an average test duration of 39.35 seconds. In the implementation of the algorithm **Naïve** Bayes using the Kfold cross-validation method, the average performance is accuracy model 77.37% with an average testing time of 0.063 seconds. Meanwhile, the Artificial Neural Network with the Kfold cross-validation method obtained an average performance accuracy of 93.04% with an average test duration of 7 minutes 28.33 seconds.

In the second scenario, the representation processing time is medical abstract 3108 records the data becomes vector data for 1 hour 10 minutes 57 seconds. The implementation of the algorithm

Naïve Bayes with the method random split obtained an average performance accuracy model of 84% with an average test duration of 0.007 seconds. Meanwhile, the Artificial Neural Network with the random split method obtained an average performance accuracy of 86.12% with an average test duration of 35.56 seconds. In the implementation of the algorithm, the Naïve Bayes Kfold cross-validation method obtained an average performance accuracy model of 77.19% with an average testing time of 0.057 seconds. Meanwhile, the Artificial Neural Network with the Kfold cross-validation method obtained an average performance accuracy of 92.45% with an average test duration of 7 minutes 50.56 seconds.

Overall, from the accuracy comparison, it is found that in the first scenario and the second scenario ANN with the Cross-Validation method has the highest accuracy value, namely 93.04% in the first scenario and 92.45% in the second scenario. In the Naive Bayes Algorit hm, the highest accuracy value obtained is 77.37% in the first scenario using the Cross-Validation method and 84% in the second scenario using the Random Split method. However, in terms of model testing time, the ANN algorithm takes more than 7 minutes for the Cross-Validation method and 37 seconds for the Random Split method in both scenarios. Meanwhile, the Naive Bayes algorithm only takes less than 1 second of model testing time for all experiments.

4. Conclusion

This paper presents the results of a study applying a combination of NLP, Naive Bayes, and ANN Classifier to predict disease types based on a doctor's diagnosis dataset. The experimenta l results show that in general ANN provides better performance than Naive Bayes. In the random split dataset, the average accuracy performance of Naive Bayes is 84%, while the average accuracy performance of ANN is 86.12%. Both are generated from the second scenario, namely the involvement of the unstructured dataset in the word2vec training process. In the cross- validation dataset, the highest Naive Bayes average accuracy performance was 77.37%, while the ANN average accuracy performance was 93.04%. Both are generated from the first scenario. On average, ANN in the first scenario gave an average accuracy of 88.79%, while in the second scenario it was 89.29%. It can be concluded that the involvement of unstructured data from the dataset in the word2vec training process improves the performance of the ANN model even though it is not significant.

References

P. S. Roshanov *et al.*, (2011) ‘Computerized clinical decision support systems for chronic disease management: A decision-maker-researcher partnership systematic review’, *Implement. Sci.*, vol. 6, no. 1, pp. 1–17.

M. Ridwan, H. Suyono, and M. Sarosa, (2013) ‘Penerapan Data Mining Untuk Evaluasi Kinerja Akademik Mahasiswa Menggunakan Algoritma Naive Bayes Classifier’, *J. EECCIS*, vol. 7, no. 1, p. pp.59-64.

J. K. Abdul Aziz Priatna, Rani Megasari, (2018) ‘Penerapan Association Rules Menggunakan Algoritma Apriori Pada Sistem Rekomendasi Pemilihan Resep Obat Berdasarkan Data Rekam Medis’, *... J. Apl. dan ...*, vol. 1, no. 2, pp. 55–60.

Guardian Y. Sanjaya, S. Harry, L. Lazuardi, and N. Faizah, (2012) ‘Datamining peresepan elektronik di pelayanan kesehatan primer: potensi pengembangan sistem pendukukung keputusan klinis’, *Semin. Nas. Inform. Medis*, no. September, pp. 26–30.

N. Indrawati, (2010) ‘NATURAL LANGUAGE PROCESSING (NLP) BAHASA INDONESIA adalah’:, no. 1.

T. F. M. Raj and S. Prasanna, (2013) ‘Implementation of ML using naïve bayes algorithm for identifying disease-treatment relation in bio-science text’, *Res. J. Appl. Sci. Eng. Technol.*, vol. 5, no. 2, pp. 421–426.

J. Olamaei and S. Ashouri, (2015) ‘Demand response in the day-ahead operation of an isolated microgrid in the presence of uncertainty of wind power’, *Turkish J. Electr. Eng. Comput. Sci.*, vol. 23, no. 2, pp. 491–504.

B. S. Prakoso, D. Rosiyadi, H. S. Utama, and D. Aridarma, (2019) ‘Klasifikasi Berita Menggunakan Algoritma Naive Bayes Classifer Dengan Seleksi Fitur Dan Boosting’, *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 3, no. 2, pp. 227–232.

Y. D. Prabowo, T. L. Marselino, and M. Suryawiguna, (2019), ‘Pembentukan Vector Space Model Bahasa Indonesia Menggunakan Metode Word to Vector’, *J. Buana Inform.*, vol. 10, no. 1, p. 29.

W. I and S. S. U. Rahman S, (2015),‘Treatment of Missing Values in Data Mining’, *J. Comput. Sci. Syst. Biol.*, vol. 09, no. 02, pp. 51–53.

Irwan budiman, M. R. Faisal, and D. T. Nugrahadi, (2020), ‘Studi Ekstraksi Fitur Berbasis Vektor Word2Vec pada Pembentukan Fitur Berdimensi Rendah’, *J. Komputasi*, vol. 8, no. 1, pp. 62–69.

B. Jang, I. Kim, and J. W. Kim, (2019), ‘Word2vec convolutional neural networks for classification of news articles and tweets’, *PLoS One*, vol. 14, no. 8, pp. 1–20.

F. Tempola, M. Muhammad, and A. Khairan, (2018), ‘Perbandingan Klasifikasi Antara KNN dan Naive Bayes pada Penentuan Status Gunung Berapi dengan K-Fold Cross Validation’, *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 5, p. 577.

F. E. Prabowo and A. Kodar, (2019), ‘Analisis Prediksi Masa Studi Mahasiswa Menggunakan Algoritma Naïve Bayes’, *J. Ilmu Tek. dan Komput.*, vol. 3, no. 2, p. 147.

A. S. Kurniawansyah, (2018), ‘Implementasi Metode Artificial Neural Network dalam Memprediksi Hasil Ujian Kompetensi Kebidanan’, vol. V.