Journal of Information Technology and Computer Science
Volume 4, Number 3, Desember 2019, pp. 233-240
Journal Homepage: www.jitecs.ub.ac.id

Preprocessing Approach for Tuberculosis DNA Classification using Support Vector Machines (SVM)

Mochammad Anshori[1], Wayan, Firdaus Mahmudy[2], Ahmad Afif Supianto[3]

[1,2,3]Faculty of Computer Science, Brawijaya University, Malang, Indonesia
[1]aanshori.moch@gmail.com, {[2]wayanfm*, [3]afif.supianto*}@ub.ac.id

Received 21 May 2019; accepted 29 November 2019

Abstract. Tuberculosis is a disease that caused by the mycobacterium tuberculosis virus. Tuberculosis is very dangerous and it is one of the top 10 causes of the death in the world. In its detection, errors often occur because it is similar to the other lung disease. The challenge is how to get the best detection system for classification of Tuberculosis using Deoxyribo Nucleic Acid (DNA) sequence data from mycobacterium tuberculosis. One way to do the detection is using machine learning algorithm which is Support Vector Machines (SVM). Before making a detection for Tuberculosis DNA Classification, it is necessarry to preprocess the DNA datasets first. Preprocessing method that we used in this research are using k-Mer for feature extraction, then processed with TF-IDF to transform it into numerical value and uniform the data length. Not Only that, because the DNA datasets is very large so dimension reduction is really needed and we used Linear Discriminant Analysis (LDA). Classification of Tubercolosis DNA will be done using Support Vector Machine (SVM) method with the best preprocesing method. So, in this research to get the best detection for DNA classification will be tested several experiment of parameters value from the method that we used in this research. The overall result based on the experiment of this research, the best *k* of k-Mer value is 5 that produce accuracy, precision, recall, F Score, and MCC are 0.927, 0.927, 0.920, 0.875.

1 Introduction

Tuberculosis (TB) is a dangerous disease. Based on WHO (World Health Organization), TB is one of the top 10 causes of death in the world. TB ever occupied the second cause of death after HIV/AIDS. This means that TB has emerged as a global health threat in this century [1]. TB is caused by the mycobacterium tuberculosis virus. In addition, the TB virus has resistance to drugs and not all TB can be treated with the same drug. Like tuberculosis with lineage from Beijing which has the highest resistance to drugs so it requires different treatments [2]. Drug-resistant TB (DR-TB) is a major threat because in 2013 as many as 3.5% of patients with tuberculosis had a defense against the drugs given [3]. The current challenge is how to detect and treat this TB. In some case the disease is difficult to detected because it resembles other respiratory diseases [4]. Therefore, it is necessary to detect TB better by using DNA data from mycobacterium tuberculosis because each organism must have DNA that differentiates and characterizes the organism. In its detection, it can be implemented by machine learning algorithms and included in the branch of science of bioinformatics. The use of machine learning is very important for bioinformatics because it can learn and build predictive models from genomes, proteins, etc as an input then analyze it [5].

There are various variants of the mycobacterium tuberculosis virus also each DNA has a different length of data. Mycobacterium tuberculosis DNA contains a

sequence of nitrogen base codes {A, T, C, G} with that order reaching thousands [6]. The Iimportant aspects before classification is the selection features from DNA sequence. This research focuses on how to extract features from DNA sequence data and preprocess it before the classification. After that, dimension reduction wil be done to reduce large data dimensions so that the classification process will be faster. The main focus of this research is how to preprocess data optimally, so that this research will produce the best classification of TB DNA.

For feature extraction from DNA sequence data will be using k-Mer method. Features are substring of DNA and required to transform into numerical vector because the input of machine learning is in numerical form [7]. One approach to transform from string to numerical is using TF-IDF method. TF-IDF could give weight to each substring, and its weight used as input to machine learning algorithm. TF-IDF also used to get uniform data length because the substring data are not in the same length. The data used is a complete genome that has up to thousands of long data, it should be high dimension of data. Therefore, dimension reduction will be carried out by LDA.

The use of k-Mer has been successfully applied to similar studies that use sequence-based data such as DNA [8]. With k-Mer can provide stable and sometimes low accuracy depending on the selection of the appropriate *k* value because each value *k* contain different information [9]. This study we used TF-IDF to change string data into numerical value. TF-IDF can convert data from DNA substring to matrix. TF-IDF converts data based on the frequency of word occurrence. In this study we used LDA to reduce the data dimension. These techniques applied for feature extraction from large dimension data to lower dimensions. LDA process is based on supervised learning [10]. In previous research have used these methods but with different objects. So, the output of the research is to prove and get the right *k* value from k-Mer to extract the feature. After that, the machine learning algorithm will be used in this paper which is Support Vector Machine (SVM) to classify TB DNA, because its good performance in classification and has successfully applied to many classification problems, and its advantage that SVM could avoid overfitting and being able to generalize data properly[11] [12].

2 Method

2.1 K-Mer

Figure 1. Sample of k-Mer

In terms of biological sequences, k-mer could be defined as all possible subsequences with length of *k* [13]. In other words, the substring generated from k-mer can represent the entire length of the data sequence as shown in Fig.1. In the field of bioinformatics, k-Mer is used as feature extraction especially for metagenome analysis. Extraction of features from k-Mer is based on the frequency of the occurrence of the combination forming the DNA.

How k-Mer works is quite simple, k-Mer will take the substring based on the specified *k* value. Increasing the value of k will produce data with large dimensions and requires high time. The *k* value of k-Mer greatly affects the features produced. In this study we will try feature extraction with different *k* values, starting from 2 to 9.

2.2 Term Frequency – Inverse Document Frequency (TF-IDF)

After extracting features using k-Mer, then changing the data into numbers and standardizing the length of the data before entering into machine learning. One method used is TF-IDF. Term frequency - inverse document frequency (TF-IDF) is

used to extract features from a document. With TF-IDF can convert text data into a matrix of numbers. The way TF-IDF works is to calculate how often or the frequency of occurrence of a particular word in the document [7]. Inverse document frequency measures the occurrence of any word in all documents. Then give the weight to each word.

TF-IDF is widely used for sentiment analysis, NLP and also text classification. Because the output of k-Mer is a substring and contain text data. So, TF-IDF can be implemented in this DNA classification case to transform text data into numerical data and get uniform data length.

2.3 Linear Discriminant Analysis (LDA)

LDA is the method used to reduce data dimensions. LDA is based on supervised learning which means it requires knowledge in it [10]. LDA work by calculating the linear discriminant between different classes and maximize the separation. The technique is compute inter and intra class distances [14]. Then homogeneous data will be close together, and heterogeneous data will be separated as far as possible. So, the data with its own class will gather together. To do reduction, LDA find its eigenvalue and eigenvector that is not zero that explain covered original data.

2.4 Support Vector Machine (SVM)

SVM is a machine learning method that can be used for classification and regression problems. In learning, SVM creates a hyperplane by maximizing data boundary margins between classes. Support vector is a predictor value that is closest to the border that separates the class. This support vector is used in calculating margin creation from SVM [10]. With this, SVM is able to generalize data to data that will be data. SVM can be applied to linear and non-linear problems using the kernel function. There are 4 kernels commonly used in SVM, namely linear, polynomial, sigmoid, and radial basis function (RBF). Linear kernels are used for data with a class that is liner separated. With this kernel it can run faster than other kernels, but when confronted with separate data that is not linear it will result in poor evaluation performance. The polynomial, sigmoid, and RBF kernels can be used for data with classes that are not linearly separated. For detailed about SVM is described in [15]. In this study the kernel used is RBF because it tends to run faster than the other kernels and sometimes gives better evaluation performance. In addition, cross validation will also be used with a fold = 10. Use cross validation to maximize classification performance by avoid overfitting [16].

2.5 Performance Measure

The performance of the machine learning classifier can be seen by its evaluation, such as accuracy, precision, recall, F score and Matthews Correlation Coefficient as outline below:

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F\ score = 2 \cdot \frac{Precision \cdot Recall}{Precision+Recall} \quad (4)$$

$$MCC = \frac{TPxTN-FPxFN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (5)$$

Where:

TP is True Positive, TN is True Negative

FP is False Positive, FN is False Negative

3 Result and Analysis

The used datasets in this research is a complete genome of mycobacterium tuberculosis which the total number are 233 data. Datasets obtained from National Center for Biotechnology Information (NCBI) website. There are 6 classes in this datasets based on main lineage of the organism [17]. The classes based on lineage because each of the lineage have a different drug resistance [2]. Detail dataset shown as Table 1 below:

Table 1. Detail the datasets

Class	Lineage	Total data
1	Indo-Oceanic	7
2	East Asian	74
3	Central Asia	12
4	Euro-America	130
5	West African 1	7
6	West African 2	2

3.1 Feature Extraction

In this section, will be explained the result after doing the feature extraction using k-Mer and TF-IDF to give weight on each substring based on k-Mer. To compare performance of k-mer and TF-IDF will be conducted with The classification using SVM with radial basis function kernel, because it can handle data that are not linearly separated. To get better result, we implement cross validation with 10 fold. With cross validation, it can avoid overfitting so the data can be used to be more general.

In this research, best k of k-Mer is 6 and produces accuracy = 0.734, precision = 0.699, recall = 0.734, F score = 0.693 and MCC = 0.515. This is the best result rather than the other k. The lowest result shown on $k = 2$. When $k > 6$, the evaluation tends to be decrease. It means by using k with value between 2 to 9, it is able to get the best k value.

Table 2. Classification result

k	2	3	4	5	**6**	7	8	9
Accuracy	0.562	0.601	0.670	0.648	**0.734**	0.717	0.567	0.567
Precision	0.474	0.543	0.617	0.560	**0.699**	0.696	0.528	0.528
Recall	0.562	0.601	0.670	0.648	**0.734**	0.717	0.567	0.567
F score	0.430	0.505	0.605	0.592	**0.693**	0.685	0.421	0.421
MCC	0.064	0.209	0.378	0.321	**0.515**	0.541	0.099	0.099
Time (s)	0.312	0.0781	0.176	0.698	**3.49**	20.4	110	439

Based on the table, we know that computational time increase for each k value. It is caused by data dimension that always be bigger for each k value. The lowest computational time on $k = 2$ with 0.312 second and the highest on $k = 9$ with 439 second, equals to 7 minute 19 second.

Data dimension shown on Table 3. With $k = 6$, the data dimension is 233, 4096. It is so huge that contain 4096 features for each data. The data dimension always increase following the k value. Consider the dimension, it is product of 4^k. As there are four characters A, C, T and G inside the DNA and k is k-Mer value. Number of features are got from DNA structure combination. Because the dimensions are huge for higher k, it needs to be reduced to decrease computational time and its dimension.

Table 3. Data dimension result using TF-IDF

k	Dimension
2	233, 16
3	233, 64
4	233, 256
5	233, 1024
6	233, 4096
7	233, 16384
8	233, 65521
9	233, 258031

3.2 Dimension Reduction

After feature extraction, dimension reduction is necessary because the previous data has a large dimension for the higher *k* values. This dimension reduction aims to prove that dimensional reduction can transform into lower dimension, reduce computational time and improve classification evaluation performance. Method that was used is Linear Discriminant Analysis (LDA). LDA is supervised learning that consider the data class. The result of dimension reduction using LDA shown as Table 3 below:

Table 3. Evaluation classification using LDA

k	2	3	4	**5**	6	7	8	9
Accuracy	0.721	0.764	0.918	**0.927**	0.923	0.918	0.914	0.906
Precision	0.665	0.708	0.910	**0.920**	0.915	0.912	0.907	0.900
Recall	0.721	0.764	0.918	**0.927**	0.923	0.918	0.914	0.906
F score	0.683	0.725	0.909	**0.920**	0.915	0.910	0.905	0.896
MCC	0.518	0.597	0.859	**0.875**	0.867	0.859	0.852	0.837
Time (s)	0.0399	0.0408	0.0538	**0.0519**	0.0479	0.0419	0.0449	0.0378

Based on Table 3, now we get the best *k* is 5 with accuracy = 0.927, precision = 0.920, recall = 0.927, F score = 0.875, and MCC = 0.875. For $k > 5$ the evaluation tends to be decrease, and when $k > 2$ the evaluation tends to be increase and reach the maximal evaluation on $k = 5$. If we see, the computational time for all of *k* are the same, with average time equals to 0.045 second. It is faster than before LDA applied.

To prove the performance of LDA in Table 4 shows the dimension comparison between without LDA and using LDA. We can see that using LDA the dimension data reduced into 5 for all of *k*. It is proven that LDA can reduced data dimension. This dimension contains information about the original data. It means the data is essential form than before the extracted data.

Table 4. Dimension comparison without LDA and with LDA

k	Without LDA	With LDA
2	233, 16	233, 5
3	233, 64	233, 5
4	233, 256	233, 5
5	233, 1024	233, 5
6	233, 4096	233, 5
7	233, 16384	233, 5
8	233, 65521	233, 5
9	233, 258031	233, 5

Now let we see comparison of computational time between using LDA and without LDA with visualization shown as Fig. 1 below. We can see that the computational time more stable by LDA for each of *k* value, it is about 0.045 second. Without LDA, computational time increase according to the *k* because its dimension

always becomes higher. Since $k >= 6$, computational time increase significantly. It is required so much time to classify the data. Based on all of this result, it is proven that using LDA can reduce computational time because the feature has been reduced and no need more time to classify.

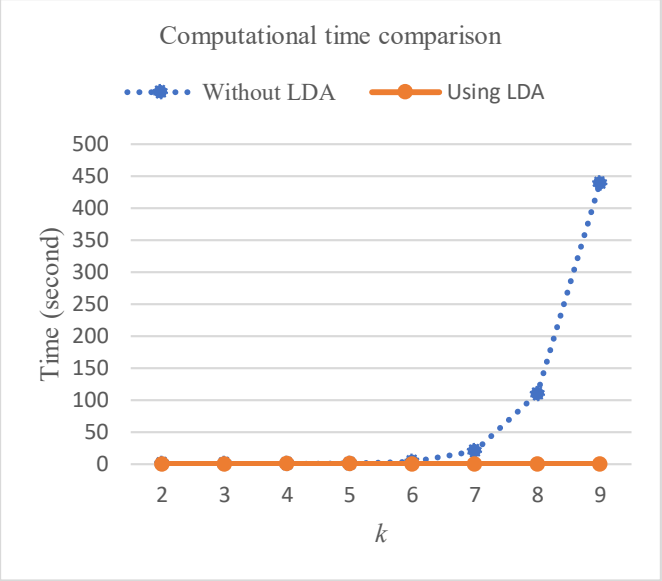

Figure 1. Comparison of computational time between LDA and without LDA

To compare SVM with the preprocessing approach to see how performance of LDA, in the Table 5 will be shown the result between SVM with before and after LDA Applied. In The table 5 shows the best k from the test that we done. The best k while LDA didn't applied is $k = 6$ and the best k for LDA Applied is $k = 5$. The k value is how many substrings from DNA will be produce. It means by capturing by 5 characters sequentially is the best for this research.

Table 5. Comparison between before and after LDA applied

	Without LDA	LDA Applied
Best k	6	5
Accuracy	0.734	0.927
Precision	0.699	0.920
Recall	0.734	0.927
F score	0.693	0.920
MCC	0.515	0.875

It can be seen from Table 5 that LDA Applied in SVM give the higher performance result of classifier. This has happened because of LDA maximize the separation between different class. So, the data will be distributed to its own class. It is also cased of LDA minimize scatter within class and maximize between the classes. It is also proven that by dimension reduction using LDA can improve the performance of classifier, in this case is SVM. The accuracy = 0.927, precision = 0.920, recall = 0.927, F score = 0.920, and MCC = 0.875.

4 Conclusion

It can be concluded that the feature extraction using k-Mer and TF-IDF is success applied. K-Mer to extract the substring of DNA and TF-IDF to transform substring into numerical vector using weight from TF-IDF. Classification algorithm that we used is SVM with radial basis function kernel. Based on the results, it can be seen that k-Mer has an influence on the extraction of Tuberculosis DNA data. The extraction features can determine the performance evaluation of classification. To extract data from DNA substring, we used k-Mer and TF-IDF. Likewise, with the selection of methods to reduce data dimensions. In this case LDA is the best because the evaluation results are very good. It is happened because LDA in reduction dimension process consider class of data. By dimension reduction, computational time being faster than before that without dimension reduction. The final result of this study is, the best k value is k = 5 based on the experiment using k-Mer and TF-IDF that used for extraction feature and classified using SVM with LDA Applied. With performance evaluation accuracy = 0.927, precision = 0.930, recall = 0.927, F score = 0.924, and MCC = 0.875.

References

[1] S. Asia, W. Paci, I. Congress, T. Evolution, and T. B. E. Meeting, "Tuberculosis in evolution," no. April, pp. 3–5, 2015.

[2] S. A. Yimer, G. Norheim, A. Namouchi, E. D. Zegeye, W. Kinander, and T. Tønjum, "Mycobacterium tuberculosis Lineage 7 Strains Are Associated with Prolonged Patient Delay in Seeking Treatment for Pulmonary Tuberculosis in Amhara Region , Ethiopia," *J. Clin. Microbiol.*, vol. 53, no. 4, pp. 1301–1309, 2015.

[3] R. De Janeiro, "Artificial Neural Network Models for Diagnosis Support of Drug and Multidrug Resistant Tuberculosis," *Lat. Am. Congr. Comput. Intell.*, pp. 1–5, 2015.

[4] Y. Zhan, B. Li, Y. Huo, A. Lin, and H. Wu, "A case of multiple organ tuberculosis," *Radiol. Infect. Dis.*, pp. 0–4, 2018.

[5] J. T. Wassan, H. Wang, and H. Zheng, "Machine Learning in Bioinformatics," *Encycl. Bioinforma. Comput. Biol.*, pp. 300–308, 2019.

[6] W. Ashlock and S. Datta, "Evolved features for DNA sequence classification and their fitness landscapes," *IEEE Trans. Evol. Comput.*, vol. 17, no. 2, pp. 185–197, 2013.

[7] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," *Expert Syst. Appl.*, vol. 57, pp. 117–126, 2016.

[8] M. Martínez-porchas and F. Vargas-albores, "An efficient strategy using k-mers to analyse 16S rRNA sequences," *Heliyon*, no. May, p. e00370, 2017.

[9] G. Han and D. Cho, "Genomics Genome classification improvements based on k-mer intervals in sequences," *Genomics*, no. October, pp. 0–1, 2018.

[10] S. Ilias, N. Tahir, R. Jailani, and S. Alam, "Feature Extraction of Autism Gait Data Using Principal Component Analysis and Linear Discriminant Analysis," *2016 IEEE Ind. Electron. Appl. Conf.*, pp. 275–279, 2016.

[11] D. Novitasari, I. Cholissodin, and W. F. Mahmudy, "Optimizing SVR using Local Best PSO for Software Effort Estimation," *J. Inf. Technol. Comput. Sci.*, vol. 1, no. 1, pp. 28–37, 2016.

[12] D. Novitasari, I. Cholissodin, and W. F. Mahmudy, "Hybridizing PSO with

SA for Optimizing SVR Applied to Software Effort Estimation," *TELKOMNIKA*, vol. 14, no. 1, pp. 245–253, 2016.

[13] D. Phan, N. G. Nguyen, F. R. Lumbanraja, and M. R. Faisal, "Combined Use of k-Mer Numerical Features and Position-Specific Categorical Features in Fixed-Length DNA Sequence Classification," *J. Biomed. Sci. Eng.*, vol. 10, no. 8, pp. 390–401, 2017.

[14] Y. Wang and Y. Chen, "A New Feature Extraction Algorithm Based on Fisher Linear Discriminant Analysis," *2017 3rd Int. Conf. Control. Autom. Robot.*, no. 1, pp. 414–417, 2017.

[15] V. N. Boser, Bernhard E. and Guyon, Isabelle M. and Vapnik, "Training Algorithm Margin for Optimal Classifiers," *COLT '92 Proc. fifth Annu. Work. Comput. Learn. theory*, pp. 144–152, 1992.

[16] C. Lameiro and P. J. Schreier, "Cross-validation techniques for determining the number of correlated components between two data sets when the number of samples is very small," *2016 50th Asilomar Conf. Signals, Syst. Comput.*, pp. 601–605, 2016.

[17] M. B. Reed *et al.*, "Major Mycobacterium tuberculosis Lineages Associate with Patient Country of Origin □ †," *J. Clin. Microbiol.*, vol. 47, no. 4, pp. 1119–1128, 2009.