The early detection system of pulmonary tuberculosis disease using learning vector quantization 2 (lvq2)

L A Widyasari, P S Sasongko, Sutikno, Suhartono, and E Reynaldhi

Department of Informatics Faculty of Science and Mathematics, Diponegoro University Semarang, Indonesia.
Email: priyoss1234@gmail.com

Abstract. Tuberculosis is an infectious disease caused by the Mycobacterium tuberculosis virus and infects the pulmonary. While in Indonesia, Central Java province was on the third-ranked for the highest number of new case of pulmonary tuberculosis disease. This disease can cause dangerous complications until death if not immediately detected and not treated completely. To help the community do early detection of pulmonary tuberculosis disease easily, this research aims to make an early detection system of pulmonary tuberculosis disease using Artificial Neural Network algorithm Learning Vector Quantization 2 (LVQ2). The variable that was used consisted of 8 symptoms of pulmonary tuberculosis disease. The research data obtained from health record data of pulmonary tuberculosis patients at Puskesmas Karangawen II Kab. Demak as much as 80 data. The distribution of training data and testing data was obtained from the application of k-fold cross-validation with the value of k = 8. The results showed that the best LVQ2 architecture for early detection system was obtained in combination of parameters learning rate (α) 0,06; smallest learning rate 0,001; window (ε) 0,3; and maximum epoch 500. The best architecture in this research produced 87,5% accuracy, 12,5% error rate, 85% sensitivity, and 90% specificity with a processing time of 8-fold was 60,68 seconds.

1. Introduction
Tuberculosis (TB) is a public health problem that is one of the main causes of death in the Asia-Pacific region, including Indonesia. TB is an infection in humans caused by Mycobacterium tuberculosis. This is due to the lack of adequate diagnostic testing facilities. According to Fariz Nurwidya [1], the detection of TB disease uses molecular detection of TB which is used for faster and simpler treatment, but the constraints are the availability of supporting facilities. Based on WHO's Global Tuberculosis Report 2016, Indonesia is one of six countries in the world which contributes to 60% of new tuberculosis cases and ranked second after India. While within the scope of Indonesia, Central Java province was ranked third for the highest number of newly dominated pulmonary tuberculosis cases dominated by male patients that were 61% [2]. However, viral infections can still occur in women and even children.

Pulmonary tuberculosis disease is spread through sputum of pulmonary tuberculosis patients. Transmission is quite vulnerable to people who often make direct contact with patients with pulmonary tuberculosis: clinical symptoms which are experienced by people with pulmonary tuberculosis in the form of respiratory disorders such as a sputum cough, sputum cough with blood, shortness of breath, and chest pain, and systemic symptoms such as fever, lethargy, night sweats, and weight loss [3]. Therefore, if a person frequently communicates with tuberculosis sufferers and feel
some of these symptoms, it is necessary to immediately conduct self-examination to prevent the spread of the virus and also do the proper treatment.

People who feel the symptoms of pulmonary tuberculosis disease can go to the health center. Since 2000, Community Health Center (Puskesmas) has run tuberculosis control programs in conjunction with the implementation of Directly Observed Treatment Short-course (DOTS) strategies and integrated into primary health services. In order to help the government and community to control tuberculosis disease, especially in early detection of pulmonary tuberculosis through symptoms, existing technology and science can be utilized to handle the problem so the control and early detection of disease can be done more efficiently, quickly and accurately.

Technology with advanced computing systems can do the work resembling the work of the human brain to solve a problem — computing system that represents how the human brain works are called an artificial neural network. According to Kenneth C. Laudon [4], artificial neural networks work by "studying" patterns of large amounts of data by filtering out data, searching for relationships, building models, and correcting the model errors itself over and over again [4].

One of the artificial neural network algorithms that can be utilized for early detection of pulmonary tuberculosis is Learning Vector Quantization (LVQ). According to Purwanti and Widiyanti [5], LVQ is a method of pattern classification which each unit of output represents a particular category or group. The processing that occurs in each neuron is to find the closest distance between an input vector to the corresponding weight; they automatically identify TB bacilli in sputum using Learning Vectors Quantization (LVQ) with an accuracy of 91.33%. Leleury [6] conducted a study to diagnose internal disease using Backpropagation and Learning Vector Quantization artificial neural networks with an accuracy rate for Backpropagation method of 61.84% while using the LVQ method the diagnostic accuracy rate was 93.42%

The LVQ2 algorithm is the development algorithm of LVQ (LVQ1) where the codebook vector is updated with different strategies [2]. Some research using LVQ2 algorithm in the medical field such as detection of coronary heart disease that has a level of accuracy of 93.3% [7] and diagnosis of a psychiatric disease that has 90% accuracy[8]. Based on several studies that have been done, LVQ2 shows relatively good accuracy and can be applied in case of classification. Based on the issues and descriptions previously mentioned, it will be built an early detection system of pulmonary tuberculosis disease using Learning Vector Quantization 2 (LVQ2). The system is built based on the website so it is expected to help the community in early detection of pulmonary tuberculosis more easily and quickly, and can be done further examination and treatment to reduce the risk of patients and the risk of transmission.

2. Materials and methods
2.1. Study Area and dataset
The data used in this study came from health records data of patients with pulmonary tuberculosis at Puskesmas Karangawen II Demak Regency. The data used amounted to 80 data consisting of 40 data of patients suspected pulmonary tuberculosis and 40 data of a patient that didn’t suspect pulmonary tuberculosis. Eight data variables used are cough type symptoms (no cough, dry cough, and sputum cough), duration of a cough (more than 2 weeks or not), cough with blood, weight loss, cold sweat at night without activity, fever, shortness of breath, and chest pain — the results of the identification of these symptoms in the form of information whether the suspect or not pulmonary tuberculosis disease in accordance with known symptoms.

2.2. LVQ2 Algorithm
LVQ2 is a developmental method from basic LVQ algorithm (LVQ1). The LVQ Algorithm serves to conduct training on the supervised competitive layer [9]. This algorithm concept finds the closest distance between the input vectors to the corresponding weight vector. During the learning process, the output unit will be positioned by regulating and updating the weights through supervised learning to estimate the decision of classification. In LVQ2, two vectors which have the closest distance (winner vector and runner-up vector) are updated if they satisfy several conditions [9]
The condition that must be met in the LVQ2 algorithm to update the weight vectors based on the two closest vectors are [8]

1. The winner unit and the runner-up (the next closest vector) represent different classes.
2. The input vector belongs to the same class as the runner-up.
3. The distances from the input vector to the winner and from the input vector to the runner-up are approximately equal. This condition is expressed regarding a window.

The window condition expressed using the following notation:

- \( X \) = Current input vector
- \( Y_c \) = Reference/weight vector that is closest to \( X \)
- \( Y_r \) = Reference/weight vector that is next to closest to \( X \) (runner-up)
- \( d_c \) = Euclidean distance from \( X \) to \( Y_c \)
- \( d_r \) = Euclidean distance from \( X \) to \( Y_r \)

The Euclidean distance equation is known by the following equation as in equation (1).

\[
\begin{align*}
\bar{d}_j &= \sqrt{\sum_{i=1}^{n}(X_i - W_{ij}^2)} \\
\bar{d}_j &= \text{Euclidean distance between the input vector and weight vector for } j \text{-th output unit, } j = 1,2,3, ..., m. \\
\bar{x}_i &= \text{Input vector on } i - \text{th variable, } i = 1,2,3, ..., n. \\
W_{ij} &= \text{Weight vector on } i - \text{th variable which goes to } j - \text{th class, } i = 1,2,3, ..., n \text{ and } j = 1,2,3, ..., m. \\
t &= \text{Index of input data, } t = 1,2,3, ..., k.
\end{align*}
\]

The two closest vectors will do the learning as far as vector input meets the window conditions. The equation which is used to know whether the input vector \( X \) fall in the window is shown in equation (2) [9]

\[
\begin{align*}
\frac{\bar{d}_c}{\bar{d}_r} > 1 - \varepsilon \text{ and } \frac{\bar{d}_r}{\bar{d}_c} < 1 + \varepsilon
\end{align*}
\]

The window values which is used are between 0.1 to 0.5, and at LVQ2 the window value (\( \varepsilon \)) is usually expressed as 0.35 [9].

If the three conditions of weight update in LVQ2 are met, then the weight vector will be updated according to the equations (3) and (4). Equation (3) is the calculation for updating the weight of winning vector \( Y_c \).

\[
Y_c(t+1) = Y_c(t) - \alpha(t)[X(t) - Y_c(t)]
\]

While equation 4 is the calculation for updating the weight of runner-up vector \( Y_r \).

\[
Y_r(t+1) = Y_r(t) + \alpha(t)[X(t) - Y_r(t)]
\]

\( Y_c \) moreover, \( Y_r \) are the variables to recognize the weight vectors which will be updated and then the fixed weight vector will be referred to as \( W_j \).

Based on the above LVQ2 explanation, the LVQ2 algorithm is (5)

0. Initialize training data \( X \) and the target \( T \)
1. Initialize weight vectors \( W_j \), learning rate (\( \alpha \)), the smallest learning rate, window (\( \varepsilon \))
2. While stopping condition is false, do steps 3-7
3. For each training input vector \( X \), do steps 4-5
   4. Find \( j \) so that \( \|X-W_j\| \) is a minimum
   5. Update \( W_j \) as follows:
      - If \( T = C_j \) then
\[ W_j(t+1) = W_j(t) + \alpha(t)[X(t) - W_j(t)] \]  
(5)

- If \( T \neq C_j \), then check the window condition using equation (2).
  - If the window condition is True, then
    - \( W_j \) which does not represent the class of input vector \( X \) (winning vector) is called \( Y_c \), will update the weight using equation (6).
      \[ Y_c(t+1) = Y_c(t) - \alpha(t)[X(t) - Y_c(t)] \]  
(6)
    - Whereas, \( W_j \) which represent the class of the input vector \( X \) (runner-up vector) is called \( Y_r \), will update the weight using equation (7).
      \[ Y_r(t+1) = Y_r(t) + \alpha(t)[X(t) - Y_r(t)] \]  
(7)
  - If the window condition is False, then do weight update using equation 8.
      \[ W_j(t+1) = W_j(t) - \alpha(t)[X(t) - W_j(t)] \]  
(8)

6. Reduce learning rate (\( \alpha \))

\[ \alpha(t+1) = \alpha(t) - (0.1 \times \alpha(t)) \]  
(9)

7. Test stop condition.

The stop condition is expressed by reaching a certain number of iterations (executing step 2), or the learning rate reaches a small enough value.

3. Methodology

3.1. Mapping Data

The data mapping stage serves to identify the data obtained into the LVQ2 network architecture into input vectors and output units. The data used consist of 9 data attributes, where 8 attributes consist from 8 variable symptoms of pulmonary tuberculosis disease which will become input vector (\( X \)) and 1 other attribute will become unit of output (\( Y_j \)). The variable of output unit was obtained from the identification data of pulmonary tuberculosis disease (\( T \)) which is known to have 2 kinds of identification result that is not suspect (\( Y_1 \)) and suspect (\( Y_2 \)) pulmonary tuberculosis disease.

3.2. Normalization

If the data used is not numerical data it is necessary to normalize the data where the category value of the data variable must be within the range \([0, 1]\) [3]. This normalization is necessary because the consistency of data in the same dynamic range can increase the accuracy level [10]. The process of normalizing the data to a numerical value in the range \([0, 1]\) is done according to equation (10) [11].

The Normalization Table is shown in table 1.

\[ X = \frac{r - 1}{R - 1} \]  
(10)

Information:
- \( X \) = value of the result of normalization
- \( r \) = scaling rank value
- \( R \) = largest scaling rank value
Table 1. Normalization Table

| Symptom Variable | Variable Description | Category | Rating Value (r) | Value of \( r \) moreover, \( R \) | Normalization Result |
|------------------|----------------------|----------|----------------|----------------|---------------------|
| \( x_1 \)       | Cough type           | No Cough | 1               | \( r = 1, R = 3 \) | 0                   |
|                  |                      | Dry Cough| 2               | \( r = 2, R = 3 \) | 0.5                 |
|                  |                      | Sputum Cough | 3   | \( r = 3, R = 3 \) | 1                   |
| \( x_2 \)       | Duration of cough (more than 2 weeks) | No | 1 | \( r = 1, R = 2 \) | 0 |
|                  |                      | Yes     | 2               | \( r = 2, R = 2 \) | 1                   |
| \( x_3 \)       | Cough with blood     | No      | 1               | \( r = 1, R = 2 \) | 0                   |
|                  |                      | Yes     | 2               | \( r = 2, R = 2 \) | 1                   |
| \( x_4 \)       | Weight loss          | No      | 1               | \( r = 1, R = 2 \) | 0                   |
|                  |                      | Yes     | 2               | \( r = 2, R = 2 \) | 1                   |
| \( x_5 \)       | Cold sweat at night without activity | No | 1 | \( r = 1, R = 2 \) | 0 |
|                  |                      | Yes     | 2               | \( r = 2, R = 2 \) | 1                   |
| \( x_6 \)       | Fever                | No      | 1               | \( r = 1, R = 2 \) | 0                   |
|                  |                      | Yes     | 2               | \( r = 2, R = 2 \) | 1                   |
| \( x_7 \)       | Shortness of breath  | No      | 1               | \( r = 1, R = 2 \) | 0                   |
|                  |                      | Yes     | 2               | \( r = 2, R = 2 \) | 1                   |
| \( x_8 \)       | Chest pain           | No      | 1               | \( r = 1, R = 2 \) | 0                   |
|                  |                      | Yes     | 2               | \( r = 2, R = 2 \) | 1                   |

3.3. K-Fold Cross Validation
The previous normalization process produces data with numerical values. The data then will be processed using k-fold cross-validation to generate training data and test data. The data is divided into \( k \) subset, where 1 subset will be the test data and the other subset become the training data. LVQ2 architecture can be seen in figure 1.

![Figure 1. LVQ2 architecture 8 Input Variable](image)

3.4. Training
The training process will be made based on the LVQ2 training algorithm which will decide the training weight.

3.5. Testing & evaluation
The testing process will use the testing data and the last weight vectors which are taken from the earlier result. The testing result is done by searching the smallest gap between training weights vector
to the input vector by using the Euclidean distance. The training weight vectors which has the smallest distance will be entered according to the input vector class. The class is based on the classification result from the testing process. After finishing the testing calculation, we can continue with the evaluation calculation based on the confusion matrix. The aim for this testing is to find the accurate, error, sensitivity, and specificity. Evaluation of the performance of a classification model is based on the object count (testing) that is predicted correctly and incorrectly. This number is tabulated in a table known as a confusion matrix. Each cell in the confusion matrix contains a number indicating how many cases occur from the actual class to the prediction class [11]. The Confusion Matrix table is shown in table 2.

| Actual Class | Predicted Class |
|--------------|-----------------|
|              | Class = 1       | Class = 0 |
| Class = 1    | TP              | FN        |
| Class = 0    | FP              | TN        |

Based on table 2, it can be seen that there are four possibilities from the predicted results, namely True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). TP is the number of correct predictions, which the data is in a positive class. TN is the number of correct predictions, which the data is in a negative class. FP is the number of wrong predictions, which the data is in a definite class. FN is the number of wrong predictions, which the data is in a negative class [12]. Information from the confusion matrix is needed to determine the performance of the classification model. Performance values of the classification model that can be known from the calculation of confusion matrix information are accuracy, error rate, sensitivity, and specificity with the following equation (11), (12) and (13):

\[
\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}
\]

\[
\text{Error rate} = \frac{TP + FN + FP + TN}{FP + FN}
\]

\[
\text{Sensitivity} = \frac{TP + TN + FP + FN}{TP + FN}
\]

\[
\text{Specificity} = \frac{TN}{TN + FP}
\]

4. Results and discussions
The testing scenario can determine the accuracy level, error rate, sensitivity, and specificity of the LVQ2 algorithm implementation of the Pulmonary Tuberculosis Disease Early Detection System. The testing scenario is displayed in the table based on the test of the k-fold variation, learning rate (α) value 0.01 to 0.09, the smallest learning rate limit is 0.01 to 0.00001, window value 0.3, and epoch maximum 500.

Based on this scenario testing, we can find seven parameter combination which can give the highest accuracy value, which is 87.5%. The result of the testing scenario which has the highest accuracy is displayed in table 3.

| K-Fold | Learning rate | Least Learning rate | Window (ε) |
|--------|---------------|---------------------|-------------|
| 2, 4   | 0.01, 0.02, 0.03, 0.00001 | 0.3 |
| 5, 8   | 0.04, 0.05, 0.06, 0.0001 | 0.01 |
| 10     | 0.07, 0.08, 0.09, 0.001 | 0.001 |

Table 4 displays the best evaluation result which can be gained by each k-fold value variety.
Table 4. Best Evaluation Result on K-Fold Variation

| K-Fold | Accuracy | Error rate | Sensitivity | Specificity |
|--------|----------|------------|-------------|-------------|
| 2-Fold | 77.5%    | 22.5%      | 72.5%       | 90%         |
| 4-Fold | 82.5%    | 17.5%      | 77.5%       | 90%         |
| 5-Fold | 83.75%   | 16.25%     | 72.5%       | 90%         |
| 8-Fold | 87.5%    | 12.5%      | 87.5%       | 90%         |
| 10-Fold| 87.5%    | 12.5%      | 87.5%       | 87.5%       |

Based on table 4, we can see that k-fold which has k=8 value can results in higher evaluation result compared to the other k-fold variation. The testing result which has k-fold in k=8 value result is displayed in table 5. The result of this testing shows the highest accuracy can be reached 87.5% if the learning rate is started from 0.03. The accuracy chart obtained from the testing scenario which is displayed in figure 2.

![Testing Scenario Accuracy Graphic](image)

Figure 2. Testing Scenario Accuracy Graphic

If we are calculating the most optimum time, the parameter which results in the highest optimum time is the utility of learning rate 0.06, epsilon 0.001, window 0.3 and maximum epoch 500 parameters. The total runtime process is 60.68 second. The best parameter can give the different accuracy level to other k-fold variation.

![K-Fold Effect to Accuracy Graphic](image)

Figure 3. K-Fold Effect to Accuracy Graphic

The best parameter testing to 8-fold can result in the best accuracy if it is implemented to other k-fold variation. The testing scenario with the best parameter is the 0.06 learning rate, epsilon 0.01, and window 0.3 into other k-fold variation which is displayed in table 5.

Table 5. Testing Result with Best Parameter to K-Fold Variation

| K-Fold | Runtime (seconds) | Accuracy | Error Rate |
|--------|-------------------|----------|------------|
| 2-Fold | 15.30             | 77.5%    | 22.5%      |
| 4-Fold | 36.24             | 82.5%    | 17.5%      |
| 5-Fold | 39.42             | 83.75%   | 16.25%     |
| 8-Fold | 60.68             | 87.5%    | 12.5%      |
| 10-Fold| 118.59            | 86.25%   | 13.75%     |
Based on the testing scenario which is done, we can see that the testing scenario result has parameter combination which has the best evaluation and the fastest runtime process. This is the parameter combination:

- \( K\)-fold = 8-fold
- Learning rate = 0.06
- Epsilon = 0.001
- Window = 0.3
- Maximum Epoch = 500

This parameter can give 87.5% of accuracy, 12.5% of error rate, 85% of sensitivity, and 90% of specification.

5. Conclusion
The conclusion we can take from this research of Pulmonary Tuberculosis Disease Early Detection using Learning Vector Quantization 2 is the early detection system of Pulmonary Tuberculosis Disease using Learning Vector Quantization 2 was built according to the functional needs and can be used to do early detection by looking at the 8 symptoms of pulmonary tuberculosis disease based on training and testing process to find particular weight vectors. The web-based system makes it easy for people to do early detection and take measures to prevent the spread of the virus if stated as suspected pulmonary tuberculosis. The testing algorithm of Pulmonary Tuberculosis disease early detection using Learning Vector Quantization 2 shows the best accuracy if we use the k-fold implementation which has \( k \) value = 8, learning rate (\( \alpha \)) parameter 0.06, epsilon 0.001, window 0.3, and epoch maximum 500. This model results in 87.5% accuracy level, 12.5% error rate, 85% sensitivity, 90% specification, and total runtime of 8-fold is 60.68 seconds.

References
[1] Nurwidya F Handayani D and Burhan E 2018 Chonnam Medical Journal 54 1.
[2] Ariani M and Budianita E 2015 Penerapan Jaringan Syaraf Tiruan Untuk Deteksi Penyakit Jantung Koroner (PJK) Menggunakan Metode Learning Vector Quantization 2 (LVQ2) Riau : UIN Suska Riau.
[3] Tuffery S 2011 Data Mining And Statistics For Decision Making (France: Wiley)
[4] Budianita E and Prijodiprodjo W 2013 IJCCS 7 155
[5] Purwanti E Widiyanti P 2012 ITJID Indonesian Journal of Tropical and Infectious Disease 3 26.
[6] Han J Kamber M 2006 Data Mining : Concepts and Techniques (San Francisco: Morgan Kaufmann Publisher).
[7] Kemenkes 2017 Data dan Informasi Profil Kesehatan Indonesia 2016 Jakarta Pusat Data dan Informasi Kementerian Kesehatan RI pp. 1-9.
[8] Kenneth C and Laundon J 2008 Sistem Informasi Manajemen Mengelola Perusahaan Digital Edisi 10 Jakarta Salemba Empat (Jakarta Salemba Empat).
[9] Laily D Lampus BS & Rombot D 2015 Jurnal Kedokteran Komunitas dan Tropik 3 2
[10] Gorunescu F 2011 Intelligent Systems Reference Library 12 Chennai (India Springer).
[11] Leleury A, Lesnussa A, and Madiuw J 2016 Jurnal Matematika Integratif 12 89
[12] Saptawati L, Mardiastuti, Karuniawati A, and Rume 2012 Jurnal Tuberkulosis Indonesia 8 1