Comparison of the Performance of Random Forest and K-Nearest Neighbor in Classifying Leukemia Using Principal Component Analysis

Sriani[1], Muhammad Ikhsan[2], Lailan Sofinah Harahap[3]
Program Studi Ilmu Komputer, Fakultas Sains dan Teknologi[1], [2], [3]
Universitas Islam Negeri Sumatera Utara, Medan, Indonesia
sriani@uinsu.ac.id[1], mhdikhsan@uinsu.ac.id[2], lailansofinahharahap@gmail.com[3]

***Abstract*— Leukemia is the most common blood cancer in Asia, one of which is Indonesia. Leukemia can affect blood cells, bone marrow, lymph nodes and other parts of the lymphatic system. One way to detect leukemia is to use microarray technology by applying gene expression. Microarrays have a very large number of genes so it is necessary to reduce the number of genes in order to eliminate irrelevant features and increase the accuracy of the classification process. The leukemia feature/gene reduction process was carried out using PCA and the classification process was carried out using RF and KNN. The accuracy results from the RF classification method using 100 n_estimators were 78.57%, while using the KNN method the accuracy results with K=1 were 78.57%, K=3 and 5 were 85.71%, and K=7 and 9 were 71.42%. The best accuracy results use KNN with K=3 and 5.**

***Keywords*— Leukimia, Microarray Data, Principal Component Analysis, Random Forest, K-Nearest Neighbor.**

I. INTRODUCTION

Leukemia is the most common cancer found in Asia, one of which is Indonesia. Based on GLOBOCAN data in 2020, leukemia in Indonesia is the 9th most common disease with 14,979 new cases and 11,530 deaths. [1]. Leukemia can affect blood cells, bone marrow, lymph nodes and other parts of the lymphatic system [2]. One way to detect leukemia is to use microarray technology by applying gene expression [3]. Microarray data has a large number of gene dimensions, so it is necessary to reduce the gene dimensions in order to eliminate features from irrelevant dimensions with the aim of increasing the accuracy of the microarray data classification process for leukemia disease data. [4].

The process of reducing leukemia microarray data uses Principal Component Analytic (PCA) reduction. Where PCA is a data transformation method that can produce a new, more important set of variables by reducing computational complexity so that it has smaller dimensions and can reduce the occurrence of deletion of important features which results in the model not being able to understand the complexity of the problem. [5]. Leukemia can cause illness and even death, so it is necessary to diagnose and detect leukemia in order to cure it early [6]. The detection process can be carried out using a classification process.

The reduced leukemia data will be classified using the Random Forest (RF) and K-Nearest Neighbor (KNN) methods. Where the RF method is carried out by building many trees to become a forest, while KNN works by using Euclidean distance. This was done with the aim of knowing which classification method is more appropriate in classifying leukemia data that has been reduced using PCA.

Research on dimension reduction using PCA and classification using RF and KNN has been carried out several times, one of which is research conducted by [6] using K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree (DT), RF, and Gradient Boosting Decision Trees to achieve accuracy of 99.20%, 98.78%, and 98.41% respectively on the microarray data that has been reduced with the DESeq technique. Meanwhile, research conducted by [7] on Predicting Complete Remission of Acute Myeloid Leukemia: Machine Learning Applied to Gene Expression using KNN, SVM and RF classification had an accuracy of 84%, 74% and 81%. In research conducted by [5] on reducing high-dimensional data using PCA and classification using RF, it was able to provide accuracy results of 90.13%.

II. RESEARCH METHOD

This research is carried out by searching or collecting data on laukimia disease first, then the data will be processed by preprocessing the data using the min-max normalization equation, after that the data attribute feature reduction will be carried out using PCA, then the data will be classified using RF and KNN, after that it will the accuracy of the RF and KNN classification was tested using a confusion matrix. Finally, the accuracy results of RF and KNN classification will be compared to find out a better and more precise classification method for classifying leukemia data. The stages of this research can be seen in Figure 1.

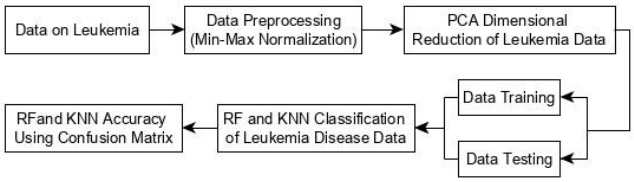

Figure 1. Data analysis process

The processes and methods used in this research are as

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v13i2.2165, Copyright ©2024
Submitted : June 11, 2024, Revised : May 16, 2024, Accepted : June 16, 2024, Published : June 30, 2024

follows:

A. Data on Leukimia

This research data was obtained from the Bioinformatics Laboratory on the website*:* www.biolab.si/supp/bi-cancer/projections/info/ using leukemia microarray data of 70 patient data with a total of 256 genes/dimensions consisting of 2 classes, namely Acute Myeloid Leukemia (AML) and Acute Lymphoblastic Leukemia (ALL).

B. Data Preprocessing

Data preprocessing is the process of correcting errors in original data such as incomplete data and irregular data formats by eliminating unimportant data in order to improve data quality [8]. The data preprocessing process can be carried out using the min-max normalization equation so that the data becomes normal by scaling the value range from 0 to 1, as follows [9].

$$v_n' = \frac{v_n - Min}{Max - Min} \quad (1)$$

Where:
v_n' = Data normalized to n.
v_n = Actual data with original data range to n.
v = Actual data with original range.

C. Principal Component Analysis (PCA)

Data that has been preprocessed will reduce the number of features or attributes using PCA so that the data is not oversimplified by carrying out the process of transforming the original data into smaller data dimensions. PCA has two main functions, namely the reduction function which is the process of reducing the number of variables to be fewer with the aim of simplifying the data mining process and the transformation function which is the process of changing initial variables that are correlated to become uncorrelated. The PCA stages are as follows [5].

a. Calculate the correlation matrix with the following equation.
   - Attribute variants

$$var(A_1) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1} \quad (2)$$

$$var(A_2) = \frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{n-1} \quad (3)$$

   - Covariance of two attributes

$$var(A_1) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})-(y_i - \bar{y})}{(n-1)} \quad (4)$$

   Where:
   x_i : data to i
   $\bar{x}$: average value x
   y_i : data to i
   $\bar{y}$: average value of y
   n : amount of data

   - Covariance matrix

$$Covariance = \begin{pmatrix} cov(A_1, A_1) & cov(A_1, A_2) \\ cov(A_2, A_1) & cov(A_2, A_2) \end{pmatrix} \quad (5)$$

b. Calculate the eigenvalue which states how much variation the PC variable can explain. If M is an m×m matrix, then each λ satisfies the following equation:

$$Mv = \lambda v \quad (6)$$

So each eigenvalue must satisfy the determinant equation:

$$|M - \lambda I| = 0 \quad (7)$$

Where:
M : covariance matrix
v : eigen vector
λ : eigen value
I : identity matrix

c. Calculating the principal component (PC), after knowing the eigen value and eigen vector, the next step is to calculate the PC value by ordering the eigen values from largest to smallest.

d. Dimensionality reduction, after sorting the eigenvalues, the next process is to select the PC variable which has a percentage value > 80% or you can make assumptions about the value you want to use in the data dimensionality reduction process. The reduction formula can be seen in equation 8 below:

$$Transformed\ Data = Row\ Data \times Row\ Feature\ Vector \quad (8)$$

Where:
Row Data : Preliminary data
Row Feature Vector : Selected eigenvectors

D. Random Forest (RF)

RF is a development algorithm from the Classification and Regression Tree (CART) method which can be used in the classification process by determining the root node of a random tree that is built. In the Decision Tree process there are several algorithms, such as CART, ID3, C45 and so on. In this research, a classification process will be carried out so that the appropriate algorithm for determining the decision tree is the CART algorithm using the following equation [10]:

$$Entropy\ (S) = -\sum_{i}^{c} P_i \log_2 P_i \quad (9)$$

$$IG(S, a) = Entropy(S) - \Sigma_v \frac{|s_v|}{|S|} Entropy(s_v) \quad (10)$$

Where:
IG = Information Gain.
v = attribute data set value a.
s_v = the amount of data that has attribute a.

E. K-Nearest Neighbor (KNN)

The results of attribute feature reduction using PCA will proceed to the classification stage using KNN. KNN is a non-parametric algorithm for predicting the value of new data points where a new class will be assigned to new data points based on the closeness of their similarity to the training set points. The steps in the k-nearest neighbor algorithm include the following [11].

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v13i2.2165, Copyright ©2024
Submitted : June 11, 2024, Revised : May 16, 2024, Accepted : June 16, 2024, Published : June 30, 2024

a. Provides datasets in the form of training data and test data.
b. Determine the neighborhood value K (K is a positive integer and is usually small).
c. Perform distance calculations on the training data rows for all points in the test data using the Euclidean distance calculation in equation 11 below:

$$d(x_i x_j) = \sqrt{\sum_{r=1}^{n}(x_{ir} - x_{jr})^2} \quad (11)$$

Where:
x_i: Sample data
x_j: Data training
r : Data variables
d : Euclidean distance
n : Data dimensions

d. Sort the resulting distance values from smallest to largest.
e. The top K rows must be selected from the sorted rows.
f. Test points will be assigned a class.

F. *Confusion Matrix*

The classification results of the Random Forest (RF) and K-Nearest Neighbor (KNN) methods will be tested for accuracy using a confusion matrix with the following equation [12].

$$\text{Accuracy (x)} = \frac{TP+TN}{TP+FN+FP+TN} \times 100\% \quad (12)$$

$$\text{precision (x)} = \frac{TP}{TP+FP} \times 100\% \quad (13)$$

$$\text{Recall (x)} = \frac{TP}{TP+FN} \times 100\% \quad (14)$$

$$\text{f1} - \text{score (x)} = \frac{2\times TP}{(2\times TP)+FN+FP} \times 100\% \quad (15)$$

Where:
TP (*True Positive*) = The number of correct positive data detected.
TN (*True Negative*) = The number of negative data detected is correct.
FP (*False Positive*) = The number of negative data detected as positive.
FN (*False Negative*) = The number of positive data detected is negative.

III. RESULTS AND ANALYSIS

This testing stage was carried out to determine the results of leukemia data analysis carried out using PCA, RF and KNN using a system that has been designed. The system is designed using the Python programming language. Where the system that has been designed will reduce leukemia disease data using the PCA method (n_components = 10), then classification will be carried out on the reduced data using RF (n_estimators = 100) and KNN (K=1, 3, 5, 7 and 9). After the classification is carried.

```
model_PCA_RF = RandomForestClassifier(n_estimators = 100, random_state = 0)
model_PCA_RF.fit(X_PCA_train, y_PCA_train)
y_PCA_RF = model_PCA_RF.predict(X_PCA_test)

print("Confusion Matrix of a random forest dengan Reduksi PCA :")
print(confusion_matrix(y_PCA_test, y_PCA_RF))
ac = accuracy_score(y_PCA_test,y_PCA_RF)
pre = precision_score(y_PCA_test, y_PCA_RF, average ='weighted')
re = recall_score(y_PCA_test, y_PCA_RF, average ='weighted')
f1 = f1_score(y_PCA_test, y_PCA_RF, average ='weighted')
np.set_printoptions()
print("Accuracy =", ac)
print("Precision =", pre)
print("Recall =", re)
print("f1-score =", f1)
```

```
Confusion Matrix of a random forest dengan Reduksi PCA :
[[5 0]
 [3 6]]
Accuracy = 0.7857142857142857
Precision = 0.8660714285714286
Recall = 0.7857142857142857
f1-score = 0.789010989010989
```

Figure 2. PCA (n_components = 10) and RF (n_estimators = 100) classification results

```
neigh = KNeighborsClassifier(n_neighbors=1)
neigh.fit(X_PCA_train, y_PCA_train)

y_PCA_KNN = neigh.predict(X_PCA_test)

print("Confusion Matrix of a random forest dengan Reduksi PCA :")
print(confusion_matrix(y_PCA_test, y_PCA_KNN))


ac = accuracy_score(y_PCA_test,y_PCA_KNN)
pre = precision_score(y_PCA_test, y_PCA_KNN, average ='weighted')
re = recall_score(y_PCA_test, y_PCA_KNN, average ='weighted')
f1 = f1_score(y_PCA_test, y_PCA_KNN, average ='weighted')

np.set_printoptions()
print("Accuracy =", ac)
print("Precision =", pre)
print("Recall =", re)
print("f1-score =", f1)
```

```
Confusion Matrix of a random forest dengan Reduksi PCA :
[[5 0]
 [3 6]]
Accuracy = 0.7857142857142857
Precision = 0.8660714285714286
Recall = 0.7857142857142857
f1-score = 0.789010989010989
```

Figure 3. PCA (n_components = 10) and KNN (K = 1) classification results

```
neigh = KNeighborsClassifier(n_neighbors=3)
neigh.fit(X_PCA_train, y_PCA_train)

y_PCA_KNN = neigh.predict(X_PCA_test)

print("Confusion Matrix of a random forest dengan Reduksi PCA :")
print(confusion_matrix(y_PCA_test, y_PCA_KNN))


ac = accuracy_score(y_PCA_test,y_PCA_KNN)
pre = precision_score(y_PCA_test, y_PCA_KNN, average ='weighted')
re = recall_score(y_PCA_test, y_PCA_KNN, average ='weighted')
f1 = f1_score(y_PCA_test, y_PCA_KNN, average ='weighted')

np.set_printoptions()
print("Accuracy =", ac)
print("Precision =", pre)
print("Recall =", re)
print("f1-score =", f1)
```

```
Confusion Matrix of a random forest dengan Reduksi PCA :
[[5 0]
 [2 7]]
Accuracy = 0.8571428571428571
Precision = 0.8979591836734694
Recall = 0.8571428571428571
f1-score = 0.8601190476190477
```

Figure 4. PCA (n_components = 10) and KNN (K = 3) classification results

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v13i2.2165, Copyright ©2024
Submitted : June 11, 2024, Revised : May 16, 2024, Accepted : June 16, 2024, Published : June 30, 2024

```
neigh = KNeighborsClassifier(n_neighbors=5)
neigh.fit(X_PCA_train, y_PCA_train)

y_PCA_KNN = neigh.predict(X_PCA_test)

print("Confusion Matrix of a random forest dengan Reduksi PCA :")
print(confusion_matrix(y_PCA_test, y_PCA_KNN))


ac = accuracy_score(y_PCA_test,y_PCA_KNN)
pre = precision_score(y_PCA_test, y_PCA_KNN, average ='weighted')
re = recall_score(y_PCA_test, y_PCA_KNN, average ='weighted')
f1 = f1_score(y_PCA_test, y_PCA_KNN, average ='weighted')

np.set_printoptions()
print("Accuracy =", ac)
print("Precision =", pre)
print("Recall =", re)
print("f1-score =", f1)
```

```
Confusion Matrix of a random forest dengan Reduksi PCA :
[[5 0]
 [2 7]]
Accuracy = 0.8571428571428571
Precision = 0.8979591836734694
Recall = 0.8571428571428571
f1-score = 0.8601190476190477
```

Figure 5. PCA (n_components = 10) and KNN (K = 5) classification results

```
neigh = KNeighborsClassifier(n_neighbors=7)
neigh.fit(X_PCA_train, y_PCA_train)

y_PCA_KNN = neigh.predict(X_PCA_test)

print("Confusion Matrix of a random forest dengan Reduksi PCA :")
print(confusion_matrix(y_PCA_test, y_PCA_KNN))


ac = accuracy_score(y_PCA_test,y_PCA_KNN)
pre = precision_score(y_PCA_test, y_PCA_KNN, average ='weighted')
re = recall_score(y_PCA_test, y_PCA_KNN, average ='weighted')
f1 = f1_score(y_PCA_test, y_PCA_KNN, average ='weighted')

np.set_printoptions()
print("Accuracy =", ac)
print("Precision =", pre)
print("Recall =", re)
print("f1-score =", f1)
```

```
Confusion Matrix of a random forest dengan Reduksi PCA :
[[5 0]
 [4 5]]
Accuracy = 0.7142857142857143
Precision = 0.8412698412698413
Recall = 0.7142857142857143
f1-score = 0.7142857142857143
```

Figure 6. PCA (n_components = 10) and KNN (K = 7) classification results

```
neigh = KNeighborsClassifier(n_neighbors=9)
neigh.fit(X_PCA_train, y_PCA_train)

y_PCA_KNN = neigh.predict(X_PCA_test)

print("Confusion Matrix of a random forest dengan Reduksi PCA :")
print(confusion_matrix(y_PCA_test, y_PCA_KNN))


ac = accuracy_score(y_PCA_test,y_PCA_KNN)
pre = precision_score(y_PCA_test, y_PCA_KNN, average ='weighted')
re = recall_score(y_PCA_test, y_PCA_KNN, average ='weighted')
f1 = f1_score(y_PCA_test, y_PCA_KNN, average ='weighted')

np.set_printoptions()
print("Accuracy =", ac)
print("Precision =", pre)
print("Recall =", re)
print("f1-score =", f1)
```

```
Confusion Matrix of a random forest dengan Reduksi PCA :
[[5 0]
 [4 5]]
Accuracy = 0.7142857142857143
Precision = 0.8412698412698413
Recall = 0.7142857142857143
f1-score = 0.7142857142857143
```

Figure 7. PCA (n_components = 10) and KNN (K = 9) classification results

Based on Figures 2 to Figure 7, it can be seen that the accuracy results of the RF classification method with a value of n_estimators = 100 is 78.57%, while using the KNN classification method with a value of K = 1 is 78.57%, K = 3 and 5 is 85.71%, K = 7 and 9 as much as 71.42%. The accuracy results are obtained from data that has been reduced using the PCA method with a value of n_components = 10. The comparison of accuracy results from the RF and KNN methods that have been reduced using PCA can be seen in table 2 below.

Table 1. Comparison of RF and KNN accuracy results

PCA	RF (*n_estimators* = 100)	K	KNN
n_components = 10	78.57%	1	78.57%
		3	**85.71%**
		5	**85.71%**
		7	71.42%
		9	71.42%

Based on table 4.12 above, it can be concluded that the best accuracy for classifying leukemia disease data that has been reduced using PCA is using the KNN classification method with K values = 3 and 5.

IV. CONCLUSION

The results of the analysis carried out on leukemia disease data in the classification of reduced data can be concluded that the Principal Component Analysis (PCA) method is able to reduce the attributes/features of leukemia disease data with a total of 10 n_components attributes/features. The number of attributes (n_components) in the PCA being built cannot exceed the number of M×N matrices of the original data matrix. The process of classifying leukemia is carried out using the Random Forest (RF) method by building 100 trees (n_estimators) and the K-Nearest Neighbor (KNN) method using values (K = 1, 3, 5, 7 and 9). The best accuracy results were in the classification of leukemia using KNN with values (K = 3 and 5) namely 85.71%. while using Random Forest (RF) by building 100 trees (n_estimators) the accuracy results were 78.57%.

V. ACKNOWLEDGMENT

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g.” Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

VI. REFERENCE

[1] D. Prasetya, “Leukemia, Penyakit Kanker Darah, yuk simak selengkapnya,” *Hermina Pateur*, 2023. https://herminahospitals.com/id/articles/leukemia-penyakit-kanker-darah-yuk-simak-selengkapnya (accessed Jan. 20, 2024).

[2] V. Rupapara, F. Rustam, W. Aljedaani, H. F. Shahzad, E. Lee, and I. Ashraf, “Blood cancer prediction using

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v13i2.2165, Copyright ©2024
Submitted : June 11, 2024, Revised : May 16, 2024, Accepted : June 16, 2024, Published : June 30, 2024

leukemia microarray gene data and hybrid logistic vector trees model," *Sci. Rep.*, vol. 12, no. 1, pp. 1–15, 2022, doi: 10.1038/s41598-022-04835-6.

[3] B. Pradana and A. Aditsania, "Implementasi Minimum Redudancy Maksimum Relevance (MRMR) dan Genetic Algorithm (GA) untuk Reduksi Dimensi pada Klasifikasi Data Micorarray Menggunakan Functional Link Neural Network (FLNN)," vol. 6, no. 2, pp. 8966–8977, 2019.

[4] I. G. N. P. V. Geramona and W. Astuti, "Implementasi Minimum Redundancy Maximum Relevance sebagai Teknik Reduksi Dimensi pada Klasifikasi Kanker Usus Besar Menggunakan Random Forest," vol. 7, no. 1, pp. 2490–2497, 2020.

[5] M. S. L. & P. S. Farah Diba, "Analisis Random Forest Menggunakan Principal Component Analysis Pada Data Berdimensi Tinggi Farah," *Indones. J. Comput. Sci.*, vol. 12, no. 4, pp. 2152–2160, 2023.

[6] Y. Xiao, J. Wu, Z. Lin, and X. Zhao, "A deep learning-based multi-model ensemble method for cancer prediction," *Comput. Methods Programs Biomed.*, vol. 153, pp. 1–9, 2018, doi: 10.1016/j.cmpb.2017.09.005.

[7] O. Gal, N. Auslander, Y. Fan, and D. Meerzaman, "Predicting Complete Remission of Acute Myeloid Leukemia: Machine Learning Applied to Gene Expression," *Cancer Inform.*, vol. 18, 2019, doi: 10.1177/1176935119835544.

[8] R. M. Awangga and N. H. Khonsa', "Analisis Performa Algoritma Random Forest dan Naive Bayes Multinomial pada Dataset Ulasan Obat dan Ulasan Film," *InComTech J. Telekomun. dan Komput.*, vol. 12, no. 1, p. 60, 2022, doi: 10.22441/incomtech.v12i1.14770.

[9] M. Arhami and M. Nasir, *Data Mining Algoritma dan Implementasi*. Yogyakarta: Penerbit Andi, 2020.

[10] A. U. Zailani and N. L. Hanun, "Penerapan Algoritma Klasifikasi Random Forest Untuk Penentuan Kelayakan Pemberian Kredit Di Koperasi Mitra Sejahtera," *Infotech J. Technol. Inf.*, vol. 6, no. 1, pp. 7–14, 2020, doi: 10.37365/jti.v6i1.61.

[11] C. Patgiri and A. Ganguly, "Adaptive thresholding technique based classification of red blood cell and sickle cell using Naïve Bayes Classifier and K-nearest neighbor classifier," *Biomed. Signal Process. Control*, vol. 68, no. April, p. 102745, 2021, doi: 10.1016/j.bspc.2021.102745.

[12] M. S. Santos, P. H. Abreu, S. Wilk, and J. Santos, "How distance metrics influence missing data imputation with k-nearest neighbours," *Pattern Recognit. Lett.*, vol. 136, pp. 111–119, 2020, doi: 10.1016/j.patrec.2020.05.032.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v13i2.2165, Copyright ©2024
Submitted : June 11, 2024, Revised : May 16, 2024, Accepted : June 16, 2024, Published : June 30, 2024