**Breast cancer classification using CNN model with Machine Learning**

Indra Cahya Ramdani1,1, Dyiyah Kresna1,2, Sugiyarto1,3\*

1Department Mathematics, Faculty of Applied and Technology, Ahmad Dahlan University, Yogyakarta, Indonesia

Email: [indra1800015053@webmail.uad.ac.id](mailto:indra1800015053@webmail.uad.ac.id)1, [dyiyah1900015004@webmail.uad.ac.id](mailto:dyiyah1900015004@webmail.uad.ac.id)2, [sugiyarto@math.uad.ac.id](mailto:sugiyarto@math.uad.ac.id)3

**Abstract**

…

*Keywords*: CNN

**Introduction**

Kanker merupakan tumor ganas yang dapat merusak sel pada tubuh. Kanker payudara merupakan penyakit yang paling sering terjadi dan penyebab kematian terbesar pada wanita[1]. Perkembangan tumor ganas ini memakan waktu yang relatif lama sehingga gejala infeksi tidak tampak mata, akan tetapi jika sudah dikategorikan level parah maka gejala ini akan muncul[2]. Kasus kanker payudara ditemukan pada wanita rentang usia 20 sampai 59 tahun. Pada tahun 2019 ada sekitar 268.000 kasus baru terindikasi kanker payudara dan 41.760 diantaranya meninggal[3]. Penangan kasus penyakit ini idealnya dilakukan sedini mungkin untuk mengurangi resiko kematian pada wanita. Penanganan ini dilakukan proses skrining dari waktu ke waktu. Jika terjadi indikasi penyakit kanker maka akan diteruskan kepada tenaga kesehatan. Penilaian resiko dari hasil skrining memberikan kesempatan untuk membuat pilihan yang tepat kepada pelayanan kesehatan untuk melakukan pencegahan pertumbuhan tumor dengan langkah efektif[4]. Proses skrining atau diagnosa kanker payudara dapat menggunakan citra medis yaitu histologi atau radiologi[5]. Akan tetapi, dalam kasus nyatanya pendeteksian kanker masih terkendala karena butuh ketelitian dalam menentukan hasilnya. Kendala ini dikarenakan kesulitan menentukan margin garis kanker dan ketepatan diagnosis kanker[6]. Penerapan teknologi yang banyak digunakan adalah pendeteksian penyakit dengan menggunakan kecerdasan buatan. Kecerdasan buatan yang dipakai adalah metode *machine learning*(ML)yang mencakup *deep learning.* Metode pendeteksian data citra dengan menggunakan *deep learning* adalah Convolution Neural Network(CNN). Aplikasi pendeteksian dengan metode CNN dapat mengurangi beban kerja tenaga kesehatan dan efisiensi waktu untuk mendiagnosa kanker payudara[7].

Data citra dikategorikan sebagai data yang tidak terstruktur. Pengolahan kasus ini tidak dapat diproses secara langsung, melainkan dengan melalui proses ekstraksi fitur. Salah satu proses ekstraksi fitur dapat menggunakan metode CNN. Metode ini dapat mengolah data citra untuk dilakukan prediksi pada kasus *suvervised learning.* CNN terdiri dari jaringan fitur ekstraksi dan artifisial kecerdasan buatan. Kinerja baik CNN pada kasus klasifikasi gambar telah terbukti pada arsitektur AlexNet dengan menghasilkan top-5 akurasi sebesar 83.6%[8]. Akan tetapi, data yang digunakan training belum tentu bisa mengeneralisasi data dalam kasus nyata. Oleh karena itu, perlu adanya pengembangan terkait arsitektur yang ada dengan metode atau arsitektur lainya.

ML memiliki banyak metode diantaranya Support Vector Machine, Naïve Bayes, Decicion Tree dan masih banyak lainya. Setiap metode memiliki kelebihanya masing-masing. Kelebihan ini dapat mempengaruhi dalam kemampuan generalisasi data tertentu. Suatu metode ML dapat juga digabungkan dengan metode ML lainya. Langkah ini disebut juga dengan *transfer learning.* Penggabungan metode yang dilakukan Sharma dan Mehra [9] menggunakan VGG16 dengan SVM menghasilkan capaian yang sangat baik pada kasus data citra hitologi. Merujuk pada penelitian sebelumnya memungkinkan untuk dilakukan percobaan dengan menggunakan arsitektur lainya.

Arsitektur yang dibuat oleh Mahmood et. al [10] menggunakan Faster R-CNN sebagai pendeteksi objek dan Deep CNNs sebagai metode pengklasifikasian. Pendeteksi objek berguna untuk menghilangkan objek yang kurang penting atau noise sehingga proses pelatihan dapat berjalan dengan ringan. Faster R-CNN menghasilkan kinerja yang sangat baik dan waktu training 10x lebih cepat dari R-CNN dan 1.6x lebih cepat dari Fast R-CNN[11]–[13]. Akurasi yang baik pada penelitian Mahmood et. al tidak terlepas pada peranan Deep CNNs yang merupakan gabungan dari dua arsitektur Densenet-201 dan Resnet-50. Penelitian Hira et. al [14] menyatakan bahwa terdapat arsitektur lain yaitu Resnet Xt-50 yang merupakan pembaharuan dari Resnet-50. Arsitektur Resnet Xt-50 mengunakan jaringan multicabang yang membawa kardinalitas dimensi baru dengan mempertimbangkan kedalaman dan lebar dimensi.

Penelitian ini akan mencoba untuk menggabungkan metode Faster R-CNN, Deep CNNs(Resnet Xt-50 dengan Densenet-201), dan pada tahap akhir proses klasifikasi akan menggunakan metode SVM. Metode dengan arsitektur ini akan dievaluasi menggunakan conffusion matrix dan metrics erornya.

**INSYG Net**

INSGY Net terdiri dari arsitektur Faster R-CNN dan Deep CNNs dengan ekstraksi fitur Densenet-201 dan Resnet Xt-50. Faster R-CNN memiliki kinerja lebih cepat dibandingkan dengan R-CNN dan Fast R-CNN. Memungkinkann dalam proses pendeteksian objek memiliki beban yang lebih rendah dan waktu komputasi yang cepat. Selain itu, INSGY Net menggunakan dua fitur ekstaktor. Setiap fitur ekstraktor memiliki nilai output pola yang berbeda. Resnet Xt-50 mengunakan jaringan multi cabang dengan membawa kardinalitas ke dimensi baru, mempertimbangkan kedalaman dan lebar dimensi. Sedangkan Densenet-201, telah terbukti dapat mengatasi masalah gradien dan meningkatnya kinerja jaringan. Menggabungkan dua arsitektur ini, memungkinkan mendapatkan output nilai pola yang dapat mengatasi masalah gradien, peningkatan kinerja, dan dapat memcahkan nilai pola dengan mempertimbangkan kedalaman dan lebar dimensi. Pada akhirnya, INSGY Net memiliki kecepatan dalam pendeteksian objek dan fitur ekstaktor yang dapat mengatasi masalah data yang besar.

**Methodology**

1. ***Convolution Neural Network***

*Convolution Neural Network* selama dekade terakhir memberika hasil trobosan di berbagai bidang yang mencakup pengenala pola seperti pemrosesan gambar hingga pengenalan suara[Albawi]. *Convolution Neural Network* sendiri dikembangkan dari *Multilayer Perceptron* (MLP) yang di desain untuk mengolah data dua dimensi. CNN adalah salah satu komponen dari deep learning yang tinggi dan banyak diaplikasikan dalam data citra atau gambar[28]. Nama CNN menunjukan kontruksi matematis yang terdiri dari 3 jenis *lapisan convolution, pooling, dan fully connected layers*[29]. CNN juga mampu mengekstraksi fitur dari data dengan konvolusi. Perbedaan CNN dengan metode ekstraksi fitur tradisional adalah CNN tidak perlu mengekstraksi fitur secara manual [30].

Diagram

Description automatically generated

Gambar 1. Arsitektur Convolution Neural Network. [8CNN]

Sebelumnya sudah dijelaskan bahwa CNN terdisri dari 3 jenis lapisan yaitu *Convolution, Pooling dan fully connected layer.*

1. *Convolution*

*Convolutional layer* merupaka komponen *fundamental* dari arsitektur CNN yang berfungsi untuk ekstraksi fitur [29] dengan demikian mesin pembelajarann representasi fitur dari gambar yang di inputkan*. Neuron* yang terdapat pada lapisan *convolutional* di susun menjadi peta fitur. *Neuron-neuron* dimemiliki bidang reseptif, yang terhubung dengan lapisan neuran sebelumnya melalui serangkaian bobot yang dapat di latih [31].

Gambar yang di input pada CNN

1. *Pooling*

*Pooling (down-sampling layer*) merupakan bagian penting dari CNN. Bagian ini berfungsi untuk menurunkan beban komputasi dengan cara menggunangi jumlah koneksi antar klapisan convolutional[Gu] . Sederhananya *pooling layer* ini digunakan untuk menggunangi resolusi special dari peta fitur sehingga mencapai invariant special yang bergunak untuk memasukan distorsi dan terjemahan [Rawat] *Pooling* sendiri terbagi menjadi beberapa jenis seperti *max-pooling, average-pooling, stochastic pooling, overlapping pooling* [Gu]. secara umum rumusan dari *pooling* sebagai berikut :

Diasumsikan sebagai feature map pada *pooling layer*

*Pooling* (x) menunjukan aturan fungsi *down-sampling*, sebagai bobot pooling. Secara umum ada nilai tetap , adalah bias dan fungsi aktifasi tidak digunakan sehingga bentuk umum dari *pooling layer* [31] :

1. *Fully Connnected leyer*

Dalam *fully Connected leyer* peta fitur pada gambar digabungkan menjadi vektor fitur berdimensi satu sebagai masukan *fully Connected leyer*. Keluaran dari lapisan *fully Connected leyer* dapat diperoleh dengan menjumlahkan bobot pada pasukan selanjutnya ditanggapi oleh fungsi aktifasi [31], yang di perlihatkan pada rumus berikut :

1. ***Support Vector Machine Machine***

Support vector machine machine (SVM) adalah model klasifikasi biner, mekanismenya merupakan untuk menemukan hyperplane klasifikasi optimal yang dapat memenuhi persyaratan klasifikasi [37]. Tujuan dari SVM adalah untuk menemukan Hyperplane pemisah yang optimal dengan memaksimalkan margin antara hyperplane pemisah dan kumpulan data [38]. SVM dapat menjamin akurasi klasifikasi hyperplane, sekaligus memaksimalkan area kosong di kedua sisi hyperplane [37]..

Diberikan dataset training dalam feature space, hyperplane dinotasikan sebagai Untuk membuat klasifikasi hyperplane dengan benar semua sample dan memiliki margin klasifikasikan, sehingga batasan diperlukan:

Dengan demikian, masalah maksimalisasi margin dapat didefinisikan sebagai:

Masalah ganda dari masalah primal dapat diperoleh dengan membangun fungsi Lagrange:

Dimana dinotasikan Lagrange multiplier.

Salah satu keunggulan SVM ialah dapat diterapkan pada data non-linier dengan memodifikasi SVM menggunalan fungsi kernel [39]. Fungsi yang membawa barisan data nonlinier ke dimensi yang lebih tinggi didefinisikan sebagai fungsi pemetaan dan ditunjukkan oleh [40]. Dalam hal ini, proses regresi dibawa ke area berdimensi tinggi dengan fungsi kernel

Dimana p adalah derajat polinomial.

1. Covarian Matrix

Suppose .....,is the measurement of n in the first column of a data. Average measurement of the sample is

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

In general, for the *j*-th variable, if there are m variables and n is the number of data, the sample average is as follows.

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

The sample variance for the *j*-th variable

|  |  |  |
| --- | --- | --- |
|  | = | (3) |

The sample covariance for the j and *h*-th variables

|  |  |  |
| --- | --- | --- |
|  | = | (4) |

1. Eigenvalues and Eigenvectors

Definition If is a matrix then the nonzero vector x in is called the eigenvector of if x is a scalar multiple of x i.e.

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

For a scalar is called the eigenvalue of and x is called the eigenvector of which corresponds to .

To determine the eigenvalues of the matrix sized so can be written as follows

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

Equation (6) is equivalent to

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

Where . Equation (7) will have a non-zero solution if

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

Equation (8) is called the characteristic equation of with a scalar that satisfies this equation is the eigenvalue of . When expanded, then *det* is a polynomial which is called the characteristic polynomial of If is the size matrix , then the characteristic polynomial has degree n dan is 1. Characteristic polynomials from matrix has the following form.

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

1. Principal Component Analysis (PCA)

Definition 2 A vector is said to be a linear combination of vectors if scalar such that

|  |  |  |
| --- | --- | --- |
|  |  | (10) |

The principal components depend only on the covariance matrix (correlation matrix ) dari , then the main components formed based on the linear combination are as follows.

|  |  |  |
| --- | --- | --- |
|  |  | (11) |

It can also be written in the matrix equation as follows.

|  |  |  |
| --- | --- | --- |
|  |  | (12) |

With is the transpose of the eigenvector matrix, PC and are the new variable matrix (principal component) and the data matrix. The principal component is a linear combination of which are uncorrelated and have the maximum variance. According to Breezin if PC= and it is known that , then from the equation The variance of each main component is obtained, namely

|  |  |  |
| --- | --- | --- |
|  |  | (13) |

To determine the first principal component weighting coefficient vector This can be done using the Lagrange function as follows:

|  |  |  |
| --- | --- | --- |
|  |  | (14) |

Maximizing variance with limitations or In order to obtain the maximum then the limit is used , using the Lagrange multiplier method, we get:

|  |  |  |
| --- | --- | --- |
|  |  | (15) |

If the equation multiplied by dan , then the result is obtained

|  |  |  |
| --- | --- | --- |
|  |  | (16) |

Because and so

|  |  |  |
| --- | --- | --- |
|  |  | (17) |

is the first principal component which is a linear combination with the aim of maximizing with a constant and is obtained that

|  |  |  |
| --- | --- | --- |
|  |  | (18) |

Where is the largest eigenvalue of the matrix ***S***.The first principal component can be written as follows.

|  |  |  |
| --- | --- | --- |
|  |  | (19) |

The form of the main component of-*p* namely is a linear combination that aims to maximize variance ( andi s not correlated with other main components but is orthogonal to other main components. The contrains that must be met by ( are and Cov( for *l<m*, so that the main components of the Cmp are as follow.

|  |  |  |
| --- | --- | --- |
|  |  | (20) |

|  |  |  |
| --- | --- | --- |
|  |  | (21) |

1. K-means Clustering

The K-means algorithm is a fairly simple clustering algorithm that partitions the best data into several k clusters. The purpose of k-means is to group data by maximizing the similarity of data in one cluster and minimizing the similarity of data between clusters. The measure of similarity used in the cluster is a function of distance. So that maximizing the similarity of the data is obtained based on the shortest distance between the data and the centroid point. The k-means algorithm starts with the formation of a cluster partition at the beginning, then iteratively improves the cluster partition until there is no significant change in the cluster partition

For example, given the data matrix X=which size with i=1,2,...,n, j=1,2,...,p, and assume the number of initial clusters is K. Next, to calculate the next *i*-th cluster centroid, the formula is used

|  |  |  |
| --- | --- | --- |
|  |  | (23) |

Euclidean Distance

Euclidean distance is a calculation to measure the distance of two points in Euclidean space which studies the relationship between angles and distances.

|  |  |  |
| --- | --- | --- |
|  |  | (24) |

1. Recurrent Neural Network (RNN)

RNN is one of the architectures of the Neural Network to process sequential data. The main difference with the Neural Network is that the signal can flow forward and backward repeatedly hidden state () based on each input () and the specified weight. After finding the value of the hidden state, then the next step is to calculate the output or prediction results () . The second step, compare the prediction results () dengan nilai keluaran yang sebenarnya atau disebut juga *target*, menggunakan *Loss Function*. *Loss Function* with the actual output value or also called the target, using the Loss Function. The Loss Function generates an error value that can indicate whether the prediction results are on target or even far from the target so that they can conclude how good or bad the performance of the RNN is. The last step, from the error value generated by the Loss Function, then the Backpropagation Through Time (BPTT) process is carried out to calculate the gradient for each time step in the network. The BPTT process is carried out to find better weights and biases than the previous process. After the BPTT process is complete, the weight and bias are updated using the Stochastic Gradient Descent (SGD) method. To calculate the hidden state value for time *t,* used formula:

|  |  |  |
| --- | --- | --- |
|  |  | (25) |

To calculate the output as a prediction, the formula is used

|  |  |  |
| --- | --- | --- |
|  |  | (26) |

1. Backpropagation Throught Time (BPTT)

BPTT is an algorithm used to change the value of the weights on the RNN. The BPTT training algorithm is usually used for time-series data. The main concept of BPTT is to expand the network at each time step by laying out the same copy of the RNN, and rearranging the network connections to get connections between subsequent. To produce accurate predictions, the parameters in the RNN such as learning rate, number of neurons, and amount of data will be tested.

Gradient is a value that is used to adjust the appropriate parameter or weight in a network, so that the network can learn. It is in this BPTT process that the gradient calculation process occurs. The bigger the gradient, the bigger the adjustment, and vice versa. BPTT uses the chain rule concept, namely the relationship between several derivatives (such as chains). The following is the BPTT chain rule.

The formulas (28) and (29) are used in the chain rule (27), which is to calculate the weight gradient

|  |  |  |
| --- | --- | --- |
|  |  | (27) |

|  |  |  |
| --- | --- | --- |
|  |  | (28) |

|  |  |  |
| --- | --- | --- |
|  |  | (29) |

The formulas (31), (32), (33) are used in the chain rule (30), which is to calculate the weight gradient

|  |  |  |
| --- | --- | --- |
|  |  | (30) |

|  |  |  |
| --- | --- | --- |
|  |  | (31) |

|  |  |  |
| --- | --- | --- |
|  |  | (32) |

|  |  |  |
| --- | --- | --- |
|  |  | (33) |

The formula (35) is used in the chain rule (34), which is to calculate the weight gradient

|  |  |  |
| --- | --- | --- |
|  |  | (34) |

|  |  |  |
| --- | --- | --- |
|  |  | (35) |

The formula (37) is used in the chain rule (36), which is to calculate the gradient bias

|  |  |  |
| --- | --- | --- |
|  |  | (36) |

|  |  |  |
| --- | --- | --- |
|  |  | (37) |

The formula (39) is used in the chain rule (38), which is to calculate the weight gradient

|  |  |  |
| --- | --- | --- |
|  |  | (38) |

|  |  |  |
| --- | --- | --- |
|  |  | (39) |

1. Stochastic Gradient Descent (SGD)

The use of Stochastic Gradient Descent in a neural network is motivated by a high cost or loss value and requires running backpropagation after. SGD can overcome high loss values by updating parameters or weights and bias after backpropagation, or in RNN-BPTT.

|  |  |  |
| --- | --- | --- |
|  |  | (40) |
|  |  |  |

**Results and Discussion**

The object used in this study is secondary data, namely river water quality data obtained from the official Kaggle website. The dataset was obtained from the official Kaggle website on the page <https://www.kaggle.com>. The dataset consists of variables taken from 2 stations. The variables of the water quality data can be seen in Table 1.

Table 1. River Water Quality Dataset

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15\_BSK5 | 15\_SO4 | 15\_CL | 15\_NH4 | 16\_BSK5 | | 16\_SO4 | 16\_CL | 16\_NH4 |
| 4,4 | 6,0 | 26,8 | 25,0 | | 0,87 | 0,54 | 40,0 | 32,0 |
| 3,4 | 9,3 | 25,8 | 25,8 | | 0,25 | 0,14 | 22,30 | 22,0 |
| 5,3 | 7,6 | 23,2 | 20,50 | | 0,07 | 0,14 | 20,40 | 18,8 |
|  |  |  |  | |  |  |  |  |
| 6,9 | 7,4 | 34,2 | 31,70 | | 0,25 | 0,16 | 29,30 | 22,6 |
| 6,8 | 6,0 | 38,9 | 29,61 | | 0,28 | 0,20 | 36,30 | 27,2 |
| 6,2 | 6,1 | 39,7 | 37,90 | | 0,17 | 0,16 | 57,17 | 45,5 |

Before the data is processed using the PCA method, EDA is first performed on the dataset. Figure 1 shows a dataset that has been preprocessed.

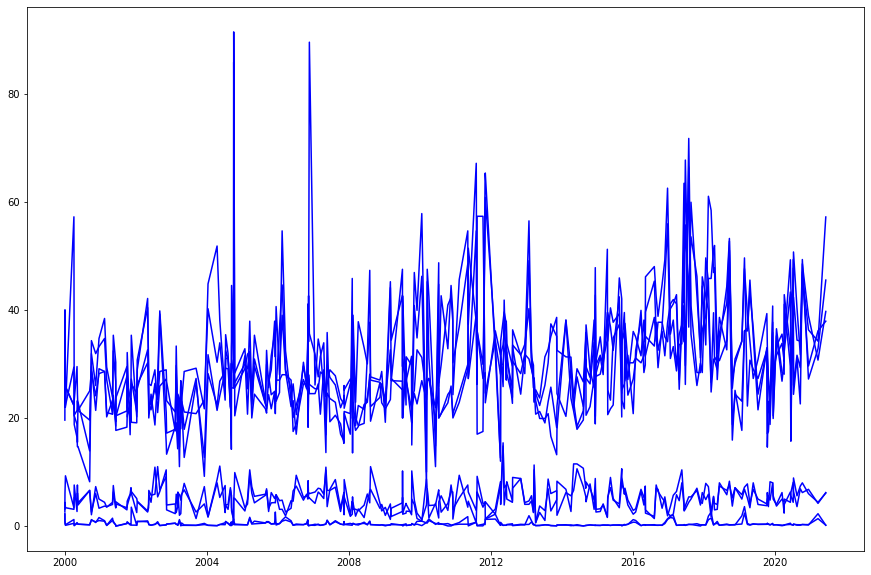


Figure 1. River water quality data

The first step in processing this data is to choose the most influential factor using the PCA method. The PCA calculation process begins with normalizing the data so that the data lies within a certain range. Furthermore, the calculation of the covariance value between variables formed into the matrix. Then the eigenvector values are calculated, and the principal components are determined.

The results of data normalization using softmax are written in matrix form as follows.

The covariance matrix shows the relationship between variables to be analyzed using PCA.

The results of the calculation of the covariance matrix are presented in the matrix below.

Based on the matrix above, it can be seen that each variable is interconnected.

Next, look for the values and eigenvectors of the covariance matrix. The results of the calculation of the eigenvectors are presented in the matrix below.

While the results of the calculation of the eigenvalue matrix are: [2,500 1,8665 1,1125 0,4132 0,2337 0,1793 0,1545].

After obtaining the eigenvalues, the value of the Cumulative Proportion of Variance (PKV) is calculated to determine the number of main components to be selected. It can be seen that the cumulative proportion of the eigenvalues is in the following table.

Table 2.Eigenvalues and Total Variance

|  |  |  |  |
| --- | --- | --- | --- |
| Component | Eigen value ( | Number of variances (%) | Cumulative (%) |
| 1 | 2,5700 | 31,95 | 31,95 |
| 2 | 1,8665 | 23,20 | 55,16 |
| 3 | 1,5129 | 18,81 | 73,97 |
| 4 | 1,1125 | 13,83 | 87,80 |
| 5 | 0,4132 | 5,13 | 92,94 |
| 6 | 0,2337 | 2,90 | 95,84 |
| 7 | 0,1793 | 2,22 | 98,07 |
| 8 | 0,1545 | 1,92 | 100 |

In Table 2, obtained 3 new variables or main components that have represented the analyzed variables. The ability of each component to represent the analyzed variables is indicated by the magnitude of the variance described, which is called the eigenvalue. The magnitude of the eigenvalues indicates the contribution of the principal component to the variance of all the original variables analyzed.

After reducing the dimensions and getting the most influential indicators using PCA, the next step is to cluster using K-means. The results of grouping on k-means are presented in Table 3 below.

Table 3.Water quality Dataset After Clustering

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X1 | X2 | X3 | Labels | Cluster Distance 1 | | Cluster Distance 2 | Cluster Distance 3 | |
| -0,5956 | 2,3740 | 1,4649 | 0 | | 0,800 | 3,66 | 3,54 |
| -0,7177 | -1,3372 | 0,2241 | 2 | | 3,500 | 2,72 | 1,04 |
| -1,3524 | -2,1054 | -0,4967 | 2 | | 4,419 | 3,71 | 1,58 |
|  |  |  |  | |  |  |  |
| 0,7676 | -1,2681 | 1,1475 | 1 | | 3,837 | 1,78 | 2,43 |
| 1,5042 | -0,5197 | -0,9615 | 1 | | 3,707 | 0,92 | 2,81 |
| 4,4425 | 0,1223 | 0,6564 | 1 | | 5,951 | 2,77 | 5,52 |

This clustering technique is also used to change the type of unsupervised data into supervised data. Furthermore, the data obtained from the results of grouping k-means will enter the RNN forecasting process. The data will first be divided into two for the training and testing process. The distribution of data is divided into 80% training data and 20% test data. The next step is the training process for the RNN model with predetermined input and output variables. Then make predictions on the test data and the last step is the process of evaluating the model using the mean squared error, root mean squared error, and mean absolute deviation that have been discussed previously. Analysis of the prediction results from the RNN model can be seen with the help of graphs so that it can be easily understood even by ordinary people. Actual data labels and predicted results are compared using a line plot. If the line plot of the prediction results is close to the line plot of the actual data label, then the prediction model built will be better. In addition, the analysis of a model that is categorized as good can also be seen by comparing the error results. If the error rate is getting smaller, the better the model that has been tested will be.

Forecasting results using RNN with PCA and RNN without PCA are presented in Figure below.

Figure 2. RNN with PCA

Figure 3. RNN without PCA

Figure 2 is the prediction result using RNN with PCA method, showing the predicted sample for 12 day. Figure 3 is the prediction result RNN without PCA method, showing the predicted sample for 30 day. Next, we will analyze the error metrics of this method, which are shown in Table 4.

Table 4.MSE, RMSE, and MAD Accuracy Values

|  |  |  |
| --- | --- | --- |
|  | RNN with PCA | RNN without PCA |
| MSE | 0,014921 | 21,00179 |
| RMSE | 0,12215 | 4,582772 |
| MAD | 0,098413 | 1,141026 |
| Accuracy | 93% | 82% |

The results of the measurement of the MSE, RMSE, and MAD error matrices are obtained from the prediction model that has been built andi s presented in the table above. Based on the analysis of Figure 1, Figure 2, and Table 4, the result show that the forecasting model using the RNN method with PCA dimension reduction is very good for processing this data.

**Conclusions**

This paper has described a model for predicting river water quality. First, the factor with the greatest influence on river water quality data was selected using PCA. The time series data are then grouped with similar trend variations using k-means clustering. After getting the supervised data, then the prediction model is built based on the RNN method. The built model produces error metrics in river water quality data with MSE 0.014921, RMSE 0.12215, MAD 0.098413, and the accuracy results obtained are 93%. The proposed model produces more accurate prediction results than the RNN prediction model without PCA feature selection. For the RNN prediction model without PCA on river water quality data, the MSE error metric value is 21,00179, RMSE 4,582772, MAD 1,141026, and the accuracy results obtained are 82%.

In future work, use different optimization algorithms to optimize the parameter selection of the RNN model in an effort to improve experimental accuracy.

**Acknowledgements**

Departments Mathematics, Faculty of Applied Science and Technology, Ahmad Dahlan University, Indonesia

**References**

[1] M. C. Chun, “Breast Cancer: Symptoms, Risk Factors, and Treatment, Medical News Today.” https://www.medicalnewstoday.com/articles/37136.php (accessed Jun. 24, 2022).

[2] T. S. P. Et. al., “Resnet Based Feature Extraction with Decision Tree Classifier for Classificaton of Mammogram Images,” *Turkish J. Comput. Math. Educ.*, vol. 12, no. 2, pp. 1147–1153, 2021, doi: 10.17762/turcomat.v12i2.1136.

[3] S. Boumaraf, X. Liu, Z. Zheng, X. Ma, and C. Ferkous, “A new transfer learning based approach to magnification dependent and independent classification of breast cancer in histopathological images,” *Biomed. Signal Process. Control*, vol. 63, no. August 2020, p. 102192, 2021, doi: 10.1016/j.bspc.2020.102192.

[4] J. D. Brooks *et al.*, “Personalized risk assessment for prevention and early detection of breast cancer: Integration and implementation (PERSPECTIVE I&I),” *J. Pers. Med.*, vol. 11, no. 6, pp. 1–18, 2021, doi: 10.3390/jpm11060511.

[5] D. Bardou, K. Zhang, and S. M. Ahmad, “Classification of Breast Cancer Based on Histology Images Using Convolutional Neural Networks,” *IEEE Access*, vol. 6, pp. 24680–24693, 2018, doi: 10.1109/ACCESS.2018.2831280.

[6] A. Saka, K. Yagi, and S. Nimura, “Endoscopic and histological features of gastric cancers after successful Helicobacter pylori eradication therapy,” *Gastric Cancer*, vol. 19, no. 2, pp. 524–530, 2016, doi: 10.1007/s10120-015-0479-y.

[7] M. Desai and M. Shah, “An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (MLP) and Convolutional neural network (CNN),” *Clin. eHealth*, vol. 4, no. 2021, pp. 1–11, 2021, doi: 10.1016/j.ceh.2020.11.002.

[8] S. Guan and M. Loew, “Breast cancer detection using transfer learning in convolutional neural networks,” *Proc. - Appl. Imag. Pattern Recognit. Work.*, vol. 2017-October, no. October 2017, 2018, doi: 10.1109/AIPR.2017.8457948.

[9] S. Sharma and R. Mehra, “Conventional Machine Learning and Deep Learning Approach for Multi-Classification of Breast Cancer Histopathology Images—a Comparative Insight,” *J. Digit. Imaging*, vol. 33, no. 3, pp. 632–654, 2020, doi: 10.1007/s10278-019-00307-y.

[10] Y. Wang, Z. Fang, H. Hong, and L. Peng, “Flood susceptibility mapping using convolutional neural network frameworks,” *J. Hydrol.*, vol. 582, p. 124482, 2020, doi: 10.1016/j.jhydrol.2019.124482.

[11] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[12] R. Girshick, “Fast R-CNN,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 1440–1448, 2015, doi: 10.1109/ICCV.2015.169.

[13] H. Huang, H. Zhou, X. Yang, L. Zhang, L. Qi, and A. Y. Zang, “Faster R-CNN for marine organisms detection and recognition using data augmentation,” *Neurocomputing*, vol. 337, pp. 372–384, 2019, doi: 10.1016/j.neucom.2019.01.084.

[14] S. Hira, A. Bai, and S. Hira, “An automatic approach based on CNN architecture to detect Covid-19 disease from chest X-ray images,” *Appl. Intell.*, vol. 51, no. 5, pp. 2864–2889, 2021, doi: 10.1007/s10489-020-02010-w.