JOURNAL OF DATA SCIENCE | Vol.2022:04
eISSN:2805-5160

Forecasting using K-means Clustering and RNN Methods with PCA Feature Selection

Ferna Marestiani[1*], Sugiyarto Surono[2]

[1,2]Ahmad Dahlan University, Yogyakarta

Email: ferna1800015008@webmail.uad.ac.id[1], Sugiyarto@math.uad.ac.id[2]

Abstract

Artificial Neural Networks is a computing system that is inspired by how the nervous system works in humans and continues to grow rapidly until now. Just like the nervous system in humans, artificial neural networks work through the process of studying existing data to formulate new data outputs. An artificial neural network using the Recurrent Neural Network (RNN) method is one of the popular models used today, especially in forecasting cases. In simple terms, the forecasting flow using the RNN method begins by dividing the test data and training data, the forward calculation process, the backward calculation process, the optimization calculation, and the evaluation calculation of the forecasting model. The main obstacle of the RNN method is the presence of a vanishing gradient which can cause poor forecasting results. In this study, the authors propose a Principal Component Analysis (PCA) dimension reduction method to obtain the most influential variables and become inputs for the prediction model that is built to minimize existing errors. The author also uses the K-means clustering method to divide the data with similar trend variations. To increase the clustering effect, the researcher used similarity calculation based on Euclidean distance. So that in an effort to build optimal prediction results, first time series data with the most influential variables will be selected using the PCA method. Furthermore, the data are grouped using the K-means method and will be included in the prediction model that is built. In the RNN prediction model, the data will be trained using the Backpropagation Through Time (BPTT) method and the optimization method used is Stochastic Gradient Descent (SGD). Forecasting with the RNN method with PCA produces an accuracy of 93%, while forecasting using the RNN method without PCA produces an accuracy of 82%. The experimental results show that the RNN method with PCA achieves higher predictive accuracy and flexibility than RNN without PCA.

Keywords

PCA, K-means Clustering, RNN, BPTT, SGD

Introduction

Forecasting is a field of science to predict events that will occur in the future using past and current data and projecting them into mathematical models (Ariyanto et al., 2017). In simple terms, there are 2 general methods of forecasting, namely qualitative methods and quantitative methods. Qualitative method is a method that is used when there is no historical data and is

Submission: 2 June 2022; **Acceptance:** 10 June 2022

intuitive, so that it is impossible to do mathematical calculations. Generally, qualitative methods are obtained from the opinions of an expert as a consideration for decision making. While the quantitative method is the method used when there is historical data, so that mathematical calculations can be carried outetode (Maricar, 2019).

The method that is often used for forecasting is the quantitative method by using time series data. Time series data is data collected based on a certain time sequence to describe the development of a situation (Desmonda et al., 2018). Several forecasting methods that use time series data are *Fuzzy* (Sumartini et al., 2017), *Markov Chain* (Rukhansah et al., 2016), ARIMA (Nur Hadiansyah, 2017), *Monte Carlo* (Moh. Jufriyanto, 2020), and *Neural Network* (SAWITRI et al., 2018). In recent times, the *Recurrent Neural Network* (RNN) has become one of the promising forecasting methods due to its wide application for forecasting and its very high generalization performance.

RNN was first developed by Jeff Elman in 1990 and is one of the architectures of the Neural Network (Novita, 2016). RNN is included in the category of Deep Learning because the data is processed through many layers (Firmansyah et al., 2020). The uniqueness of the RNN method is that it has a very good description and can overcome feedforward weaknesses. Another uniqueness of RNN is that there is a feedback connection that carries noise information at the time of the previous input which will be accommodated for the next input (Informasi, 2017).

In the research conducted (Cao et al., 2020), Cao et al use the GRU model in forecasting to get a better accuracy value. The suggested development is the use of different optimizations to obtain better accuracy from the research. So in this study the author uses the RNN prediction model with data training using *Backpropagation Through Time* (BPTT). Another development carried out in this research is the use of the *Stochastic Gradient Descent* (SGD) activation function during data training. Which before the data is processed into the RNN model, the time series data will be processed into the following 2 stages. First, the time series data will select the most influential factors using the *Principal Component Analysis* (PCA) method. The many factors used in forecasting are often accompanied by high complexity which will cause a large computational load. In addition, the influence of the main factors and supporting factors have an important role in forecasting so that only important factors will be used. Therefore, it is necessary to select features, one of which is by using PCA. The purpose of feature selection is to reduce factors or variables in order to produce less complexity in forecasting analysis.

Second, dividing time series data and grouping data with similar trends using the K-means Clustering method. If the time series data is predicted directly, the prediction accuracy obtained is likely to be low due to the characteristics and trends of different data variations, so it is necessary to group the data to get optimal prediction results. To increase the clustering effect, the researcher used similarity calculation based on Euclidean distance and Manhattan distance. Furthermore, the data will be processed into the RNN model.

This method will be applied to the case of forecasting river water quality and daily climate. The level of water quality and daily climate is currently one of the topics of conversation, especially in big cities. Humans as living beings cannot possibly live without water, therefore the river as a source of water needs to be maintained so that it can function sustainably. Several studies related to river water quality forecasting using the Neural Network method (Wu et al., 2021), Regression Analysis (Shakhari et al., 2019), Fuzzy Neural Network (Sun & He, 2018), to Adaptive Neuro-Fuzzy Inference System (ANFIS) (Evangelista et al., 2020).

Methodology

1. Covarian Matrix

Suppose $x_{11}, x_{21}, \ldots\ldots, x_{n1}$ is the measurement of n in the first column of a data. Average measurement of the sample $\bar{x}_1$ is

$$\bar{x}_1 = \frac{1}{n}\sum_{i=1}^{n} x_{j1} \tag{1}$$

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.

$$\bar{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij} \tag{2}$$

The sample variance for the *j*-th variable

$$\begin{aligned} s_j^2 &= s_{jj} \\ &= Var(X_j) \\ &= \frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j) \end{aligned} \tag{3}$$

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

$$\begin{aligned} s_{jh} &= Cov(X_j, X_h) \\ &= \frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)\frac{1}{n-1}\sum_{i=1}^{n}(x_{ih} - \bar{x}_j)^2 \end{aligned} \tag{4}$$

2. Eigenvalues and Eigenvectors

Definition 1 (sLuis & Moncayo, n.d.) If A is a matrix $n \times n$ then the nonzero vector x in R^n is called the eigenvector of A if Ax is a scalar multiple of x i.e.

$$Ax = \lambda x \tag{5}$$

For a scalar λ is called the eigenvalue of A and x is called the eigenvector of A which corresponds to λ.
To determine the eigenvalues of the matrix A sized $n \times n$ so Ax $= \lambda x$ can be written as follows

$$Ax = \lambda Ix \tag{6}$$

Equation (6) is equivalent to

$$(\lambda I - A)x = 0 \tag{7}$$

Where $x \neq 0$. Equation (7) will have a non-zero solution if

$$det(\lambda I - A) = 0 \tag{8}$$

Equation (8) is called the characteristic equation of A with a scalar that satisfies this equation is the eigenvalue of . When expanded, then *det* ($\lambda I -$ A) is a polynomial λ which is called the characteristic polynomial of A. If A is the size matrix $n \times n$, then the characteristic polynomial has degree n dan λ^n is 1. Characteristic polynomials $p(\lambda)$ from matrix $n \times n$ has the following form.

$$p(\lambda) = det(\lambda I - A) \tag{9}$$

$$= \lambda^n + c_1\lambda^{n-1} + \cdots . + c_n$$
$$= 0$$

3. Principal Component Analysis (PCA)

Definition 2 (Luis & Moncayo, n.d.) A vector ω is said to be a linear combination of vectors $v_1, v_2, \ldots, v_r$ if scalar $c_1, c_2, \ldots, c_r$ such that

$$\omega = c_1 v_1 + c_2 v_2 + \ldots . + c_r v_r \tag{10}$$

The principal components depend only on the covariance matrix S (correlation matrix R) dari $X_1, X_2, \ldots, X_n$, then the main components formed based on the linear combination are as follows.

$$PC_m = \sum_{j=1}^{m} a_{jm} X_m$$
$$= a_{1m}X_1 + a_{2m}X_2 + \ldots + a_{mm}X_m \tag{11}$$

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

$$PC_{m\times 1} = a^T X^T m \times 1 \tag{12}$$

With a^T is the transpose of the eigenvector matrix, PC and X^T are the new variable matrix (principal component) and the data matrix. The principal component is a linear combination of $PC_1, PC_2, \ldots, PC_m$ which are uncorrelated and have the maximum variance. According to Breezin (Brézin, 2002) if PC=$a^T X$ and it is known that $S = [(X - \bar{x})(X - \bar{x})^T]$, then from the equation $PC_{m\times 1} = a^T X^T m \times 1$ The variance of each main component is obtained, namely

$$Var(PC) = aSa^T \tag{13}$$

To determine the first principal component weighting coefficient vector PC_1 This can be done using the Lagrange function as follows:

$$\frac{\partial L}{\partial x} = 0 \tag{14}$$

Maximizing variance $PC_1 = a_1^T S a_1$ with limitations $a_1^T a_1 = 1$ or $a_1^T a_1 - 1 = 0$. In order to obtain $Var(PC_i)$ the maximum then the limit is used $a_i^T a_i = 1$, using the Lagrange multiplier method, we get:

$$a_i^T a_i = 0 \tag{15}$$

If the equation $Sa_i - \lambda_i a_i = 0$ multiplied by a_i^T dan $a_i^T a_i = 1$, then the result is obtained

$$\lambda_i = a_i^T S a_i \tag{16}$$

Because $Var(PC_i) = a_i S a_i^T$ and $\lambda_i = a_i^T S a_i$ so

$$Var(PC_i) = a_i^T S a_i = \lambda_i \tag{17}$$

$PC_1 = a_i^T X$ is the first principal component which is a linear combination with the aim of maximizing (PC_1) with a constant $a_1^T a_1 = 1$ and is obtained that

$$Var(PC_i) = a_1^T S a_1 = \lambda_1 \tag{18}$$

Where λ_1 is the largest eigenvalue of the matrix $\boldsymbol{S}$. The first principal component can be written as follows.

$$PC_1 = a_{11}X_1 + a_{21}X_2 + \cdots + a_{m1}X_m = a_1^T X \quad (19)$$

The form of the main component of-p namely $PC_m = a_m^T X$ is a linear combination that aims to maximize variance (PC_m) andi s not correlated with other main components but is orthogonal to other main components. The contrains that must be met by (PC_m) are $a_m^T a_m = 1$ and Cov($a_m^T X, a_l^T X$) $= 0$ for l<m, so that the main components of the Cmp are as follow.

$$PC_m = a_{1m}X_1 + a_{2m}X_2 + \cdots + a_{mm}X_m = a_m^T X \quad (20)$$

$$Var(PC_m) = a_m^T S a_m = \lambda_m \quad (21)$$

4. 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 (Sibuea & Safta, 2017).

For example, given the data matrix X=$\{X_{ij}\}$ which size $n \times p$ 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

$$C_i = \frac{\sum_{i=1}^{n} x_i \in s_i}{n} \quad (23)$$

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 (Mustofa & Suasana, 2018).

$$d(x,y) = |x - y| = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (24)$$

5. 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 (Al Kindhi et al., 2019). The training process for RNN is the same as the training process for neural networks in general. There are three main steps of the training process in RNN. First, carry out the forward pass process and make predictions (Babaee et al., 2018). In this process, calculations are carried out for each hidden state (h_t) based on each input (x_t) and the specified weight. After finding the value of the hidden state, then the next step is to calculate the output or prediction results (z_t) . The second step, compare the prediction results (z_t) 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:

$$h_t = f(W_{hh}\, h_{t-1} + W_{xh}\, x_t + b_h \tag{25}$$

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

$$z_t = f(W_{hz}\, h_t + b_z) \tag{26}$$

6. 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 copies (Juanda et al., 2018). 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 W_{hz}

$$\frac{\partial E}{\partial W_{hz}} = \frac{\partial E}{\partial z}\,\frac{\partial z}{\partial W_{hz}} \tag{27}$$

$$\frac{\partial E}{\partial z} = -\,(y - z) \tag{28}$$

$$\frac{\partial z}{\partial W_{hz}} = (z)\ (1-z)(h) \tag{29}$$

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

$$\frac{\partial E}{\partial W_{hh}} = \sum_{k=1}^{t-1} \frac{\partial E}{\partial z}\ \frac{\partial z}{\partial h_t}\ \frac{\partial h_t}{\partial h_k}\ \frac{\partial h_{t-1}}{\partial W_{hh}} \tag{30}$$

$$\frac{\partial z}{\partial h_t} = (z)\ (1-z)(W_{hz}) \tag{31}$$

$$\frac{\partial h_t}{\partial h_k} = (h)\ (1-)(\ W_{hh}) \tag{32}$$

$$\frac{\partial h_{t-1}}{\partial W_{hh}} = (h)\ (1-h)(h_{t-1}) \tag{33}$$

The formula (35) is used in the chain rule (34), which is to calculate the weight gradient $W_{\mathrm{x}h}$

$$\frac{\partial E}{\partial W_{xh}} = \sum_{k=1}^{t-1} \frac{\partial E}{\partial z}\ \frac{\partial z}{\partial h_t}\ \frac{\partial h_t}{\partial h_k}\ \frac{\partial h_{t-1}}{\partial W_{xh}} \tag{34}$$

$$\frac{\partial h_{t-1}}{\partial W_{xh}} = (h)\ (1-h)(x) \tag{35}$$

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

$$\frac{\partial E}{\partial b_z} = \frac{\partial E}{\partial z}\ \frac{\partial z}{\partial b_z} \tag{36}$$

$$\frac{\partial z}{\partial b_z} = (z)\ (1-z)(1) \tag{37}$$

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

$$\frac{\partial E}{\partial W_{xh}} = \sum_{k=1}^{t-1} \frac{\partial E}{\partial z}\ \frac{\partial z}{\partial h_t}\ \frac{\partial h_t}{\partial h_k}\ \frac{\partial h_{t-1}}{\partial b_h} \tag{38}$$

$$\frac{\partial h_{t-1}}{\partial b_h} = (h)\ (1-h)(1) \tag{39}$$

7. 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 training (Haqqi & Kusumoputro, 2022). SGD can overcome high loss values by updating parameters or weights and bias after backpropagation, or in RNN-BPTT.

$$\theta' = \theta - \alpha \frac{\partial E}{\partial \theta} \tag{40}$$

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 BSK_5, SO_4, CL, NH_4 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.

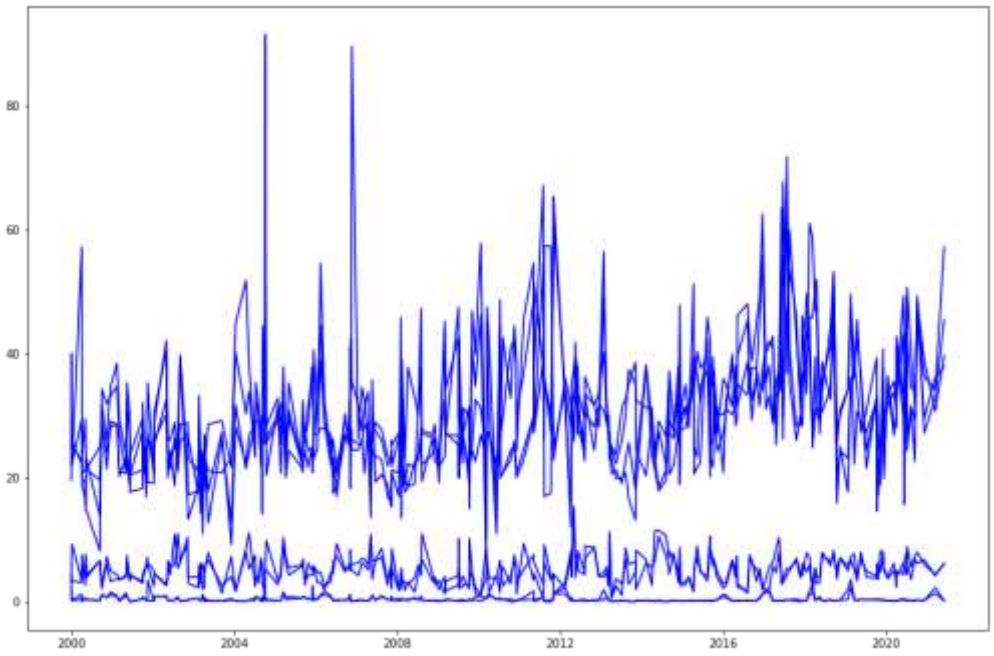

Figure 1. River water quality data

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.

$$\begin{bmatrix} -0{,}3812 & 0{,}4233 & 0{,}1793 & \cdots & 1{,}0519 & 1{,}0826 & 0{,}3521 \\ -0{,}8302 & 2{,}0277 & -0{,}3581 & \cdots & -0{,}7549 & -0{,}9065 & -0{,}7001 \\ 0{,}0228 & 1{,}2012 & -0{,}8231 & \cdots & -0{,}7549 & -1{,}1200 & -1,-369 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0{,}7412 & 1{,}1040 & 1{,}1440 & \cdots & -0{,}6646 & -0{,}1198 & -0{,}6370 \\ 0{,}6963 & 0{,}4233 & 1{,}9845 & \cdots & -0{,}4839 & -0{,}6668 & -0{,}1529 \\ 0{,}4269 & 0{,}4719 & 2{,}1276 & \cdots & -0{,}6646 & 3{,}0122 & 1{,}7729 \end{bmatrix}$$

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.

$$\begin{bmatrix} 1{,}0053 & 0{,}5921 & 0{,}0241 & -0{,}0063 & -0{,}1194 & -0{,}0967 & -0{,}0095 & -0{,}1068 \\ 0{,}5921 & 1{,}0053 & 0{,}0652 & 0{,}0500 & -0{,}0535 & -0{,}0843 & -0{,}0394 & -0{,}0365 \\ 0{,}0241 & 0{,}0652 & 1{,}0053 & 0{,}7930 & -0{,}1196 & -0{,}0516 & 0{,}3596 & 0{,}2808 \\ -0{,}0063 & 0{,}0500 & 0{,}7930 & 1{,}0053 & -0{,}1207 & -0{,}0593 & 0{,}3349 & 0{,}3237 \\ -0{,}1194 & -0{,}0535 & -0{,}1196 & -0{,}1207 & 1{,}0053 & 1{,}8185 & -0{,}1687 & -0{,}1568 \\ -0{,}0967 & -0{,}0843 & -0{,}0516 & -0{,}0593 & 0{,}8185 & 1{,}0053 & -0{,}0595 & -0{,}0281 \\ -0{,}0953 & -0{,}0394 & 0{,}3596 & 0{,}3349 & -0{,}1687 & -0{,}0595 & 1{,}0053 & 0{,}8207 \\ -0{,}1068 & -0{,}0365 & 0{,}2808 & 0{,}3237 & -0{,}1568 & -0{,}0281 & 0{,}8207 & 1{,}0053 \end{bmatrix}$$

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.

$$\begin{bmatrix} -0{,}0121 & -0{,}4495 & 0{,}4939 & -0{,}2155 & 0{,}7036 & 0{,}0418 & -0{,}0908 & 0{,}0453 \\ 0{,}0268 & -0{,}3997 & 0{,}5380 & -0{,}2456 & -0{,}6931 & -0{,}0236 & 0{,}0761 & -0{,}0532 \\ 0{,}4594 & 0{,}0764 & 0{,}2760 & 0{,}4567 & 0{,}0439 & -0{,}5706 & 0{,}2272 & 0{,}3449 \\ 0{,}4635 & 0{,}0879 & 0{,}2543 & 0{,}4577 & -0{,}0200 & 0{,}5991 & -0{,}2298 & -0{,}3015 \\ -0{,}2776 & 0{,}5075 & 0{,}3931 & -0{,}0536 & -0{,}0718 & -0{,}0515 & -0{,}6176 & 0{,}3443 \\ -0{,}2055 & 0{,}5421 & 0{,}4034 & -0{,}1104 & 0{,}1218 & 0{,}0480 & 0{,}5916 & -0{,}3487 \\ 0{,}4869 & 0{,}1849 & -0{,}0599 & -0{,}4555 & 0{,}0445 & -0{,}4033 & -0{,}3170 & -0{,}5024 \\ 0{,}4655 & 0{,}1968 & -0{,}0684 & -0{,}5024 & -0{,}0008 & 0{,}3813 & 0{,}2223 & 0{,}5408 \end{bmatrix}$$

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 (λ)	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.

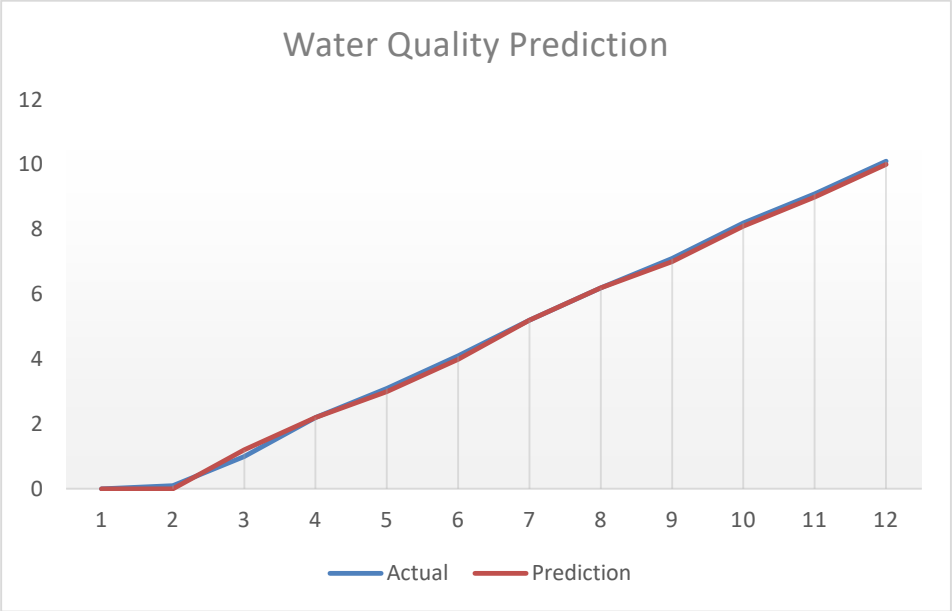

Figure 2. RNN with PCA

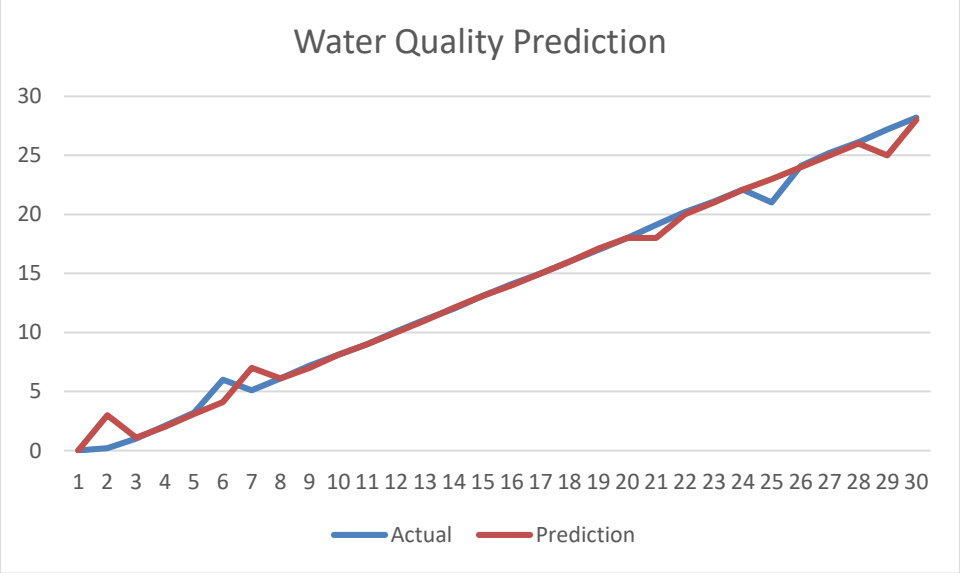

Figure 3. RNN without PCA

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

Al Kindhi, B., Sardjono, T. A., & Hery Purnomo, M. (2019). Prediction of DNA Hepatitis C Virus based on Recurrent Neural Network-Back Propagation through Time (RNN-BPTT). *2019 International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation, ICAMIMIA 2019 - Proceeding*, 208–214. https://doi.org/10.1109/ICAMIMIA47173.2019.9223395

Ariyanto, R., Puspitasari, D., & Ericawati, F. (2017). Penerapan Metode Double Exponential Smoothing Pada Peramalan Produksi Tanaman Pangan. *Jurnal Informatika Polinema*, *4*(1), 57. https://doi.org/10.33795/jip.v4i1.145

Babaee, M., Li, Z., & Rigoll, G. (2018). Occlusion Handling in Tracking Multiple People Using RNN. *Proceedings - International Conference on Image Processing, ICIP*, 2715–2719. https://doi.org/10.1109/ICIP.2018.8451140

Brézin, E. (2002). Introduction to Matrix Models. *Asymptotic Combinatorics with Application to Mathematical Physics*, 23–50. https://doi.org/10.1007/978-94-010-0575-3_2

Cao, X., Liu, Y., Wang, J., Liu, C., & Duan, Q. (2020). Prediction of dissolved oxygen in pond culture water based on K-means clustering and gated recurrent unit neural network. *Aquacultural Engineering*, *91*(17), 102122. https://doi.org/10.1016/j.aquaeng.2020.102122

Desmonda, D., Tursina, T., & Irwansyah, M. A. (2018). Prediksi Besaran Curah Hujan Menggunakan Metode Fuzzy Time Series. *Jurnal Sistem Dan Teknologi Informasi (JUSTIN)*, *6*(4), 141. https://doi.org/10.26418/justin.v6i4.27036

Evangelista, D. G. D., Bedruz, R. A. R., Vicerra, R. R. P., & Bandala, A. A. (2020). Design of Adaptive Neuro- Fuzzy Inference System (ANFIS) Model on Assessment of Effluent to Class C Fresh Surface Waters. *2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management, HNICEM 2020*. https://doi.org/10.1109/HNICEM51456.2020.9399993

Firmansyah, M. R., Ilyas, R., & Kasyidi, F. (2020). Klasifikasi Kalimat Ilmiah Menggunakan Recurrent Neural Network. *Prosiding The 11th Industrial Research Workshop and National Seminar*, *11*(1), 488–495.

Haqqi, M. S., & Kusumoputro, B. (2022). *Komparasi Metode Optimasi Adam dan SGD dalam Skema Direct Inverse Control untuk Sistem Kendali Data Sikap dan Ketinggian Quadcopter*. *10*(2), 458–469.

Informasi, F. T. (2017). *Peramalan Laju Inflasi Di Indonesia Menggunakan Back Propagation Neural Network Forecasting of Inflation Rate in Indonesia Using Back Propagation Neural Network Menggunakan Back Propagation Neural*.

Juanda, R. A., Jondri, & Rohmawati, A. A. (2018). Prediksi Harga Bitcoin Dengan Menggunakan Recurrent Neural Network. *E-Proceeding of Engineering*, *5*(2), 3682–3690.

Luis, F., & Moncayo, G. (n.d.). *Elementary Linear Algebra: Applications Version, 11th Edition*.

Maricar, M. A. (2019). Analisa Perbandingan Nilai Akurasi Moving Average Dan Exponential Smoothing Untuk Sistem Peramalan Pendapatan Pada Perusahaan XYZ. *Jurnal Sistem Dan Informatika*, *13*(2), 36–45.

Moh. Jufriyanto. (2020). Peramalan Permintaan Keripik Singkong dengan Simulasi Monte Carlo Forecasting Demand for Cassava Chips with Monte Carlo Simulation. *Jurnal Teknik Industr*, *6*(2), 107–113.

Mustofa, Z., & Suasana, I. S. (2018). Algoritma Clustering K-Medoids Pada E-Government Bidang Information And Communication. *Jurnal Teknologi Dan Komunikasi*, *9*, 1–10.

Novita, A. (2016). Prediksi Pergerakan Harga Saham Pada Bank Terbesar Di Indonesia Dengan Metode Backpropagation Neural Network. *Jutisi*, *05*(01), 965–972.

Nur Hadiansyah, F. (2017). Prediksi Harga Cabai dengan Menggunakan pemodelan Time Series ARIMA. *Indonesian Journal on Computing (Indo-JC)*, *2*(1), 71. https://doi.org/10.21108/indojc.2017.2.1.144

Series (Studi Kasus: PT. Telekomunikasi Indonesia, Tbk Kandatel Sukabumi). *Jurnal Ilmiah SANTIKA*, *8*(2), 1–17.

Rukhansah, N., Muslim, M. A., & Arifudin, R. (2016). Peramalan Harga Emas Menggunakan Fuzzy Time Series Markov Chain Model. *Komputaki*, *1*(1), 56–74. https://www.unaki.ac.id/ejournal/index.php/komputaki/article/view/113

Sawitri, M. N. D., Sumarjaya, I. W., & Tastrawati, N. K. T. (2018). Peramalan Menggunakan Metode Backpropagation Neural Network. *E-Jurnal Matematika*, *7*(3), 264. https://doi.org/10.24843/mtk.2018.v07.i03.p213

Shakhari, S., Verma, A. K., & Banerjee, I. (2019). Remote location water quality prediction of the indian river ganga: Regression and error analysis. *International Conference on ICT and Knowledge Engineering*, *2019-Novem*. https://doi.org/10.1109/ICTKE47035.2019.8966796

Sibuea, M. L., & Safta, A. (2017). Pemetaan Siswa Berprestasi Menggunakan Metode K-Means Clustring. *Jurteksi*, *4*(1), 85–92. https://doi.org/10.33330/jurteksi.v4i1.28

Sumartini, Hayati, M. N., & Wahyuningsih, S. (2017). Peramalan Menggunakan Metode Fuzzy Time Series Cheng. *Jurnal EKSPONENSIAL*, *8*, 51–56.

Sun, H., & He, Y. (2018). Research and Application of Water Quality Evaluation of a Certain Section of Yangtze River Based on Fuzzy Neural Network. *Proceedings - 2017 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration, ICIICII 2017*, *2017-Decem*, 301–304. https://doi.org/10.1109/ICIICII.2017.39

Wu, Y., Ling, R., Zhou, J., Zhang, M., & Gao, W. (2021). Prediction of Water Quality based on artificial neural network. *Journal of Physics: Conference Series*, *1738*(1), 1–5. https://doi.org/10.1088/1742-6596/1738/1/012066