International Journal of Innovative Research and Scientific Studies, 6(2) 2023, pages: 330-343

IJIRSS
International Journal of Innovative Research & Scientific Studies

ISSN: 2617-6548

URL: www.ijirss.com

Developing an optimized recurrent neural network model for air quality prediction using K-means clustering and PCA dimension reduction

Sugiyarto Surono[1*], Khang Wen Goh[2], Choo Wou Onn[3], Ferna Marestiani[4]

[1,4]*Department of Mathematics, Ahmad Dahlan University, Yogyakarta, Indonesia.*
[2,3]*Faculty of Data Science and Information Technology, INTI International University, Nilai, Malaysia.*

Corresponding author: Sugiyarto Surono (*Email: sugiyarto@math.uad.ac.id*)

Abstract

Prediction is a means of forecasting a future value by using and analyzing historical or current data. A popular neural network architecture used as a prediction model is the Recurrent Neural Network (RNN) because of its wide application and very high generalization performance. This study aims to improve the RNN prediction model's accuracy using k-means grouping and PCA dimension reduction methods by comparing the five distance functions. Data were processed using Python software and the results obtained from the PCA calculation yielded three new variables or principal components out of the five examined. This study used an optimized RNN prediction model with k-means clustering by comparing the *Euclidean, Manhattan, Canberra, Average,* and *Chebyshev* distance functions as a measure of data grouping similarity to avoid being trapped in the local optimal solution. In addition, PCA dimension reduction was also used in facilitating multivariate data analysis. The k-means grouping showed that the most optimal distance is the average function producing a DBI value of 0.60855 and converging at the 9th iteration. The RNN prediction model results evaluated based on the number of RMSE errors which was 0.83, while that of MAPE was 8.62%. Therefore, it was concluded that the K-means and PCA methods generated a more optimal prediction model for the RNN method.

Keywords: Dimension reduction, Forecasting, K-means, Metrics space, Principal component analysis, Recurrent neural network.

DOI: 10.53894/ijirss.v6i2.1427
Funding: This study received no specific financial support.
History: Received: 4 October 2022/**Revised:** 24 February 2023/**Accepted:** 6 March 2023/**Published:** 21 March 2023

Authors' Contributions: All authors contributed equally to the conception and design of the study.
Competing Interests: The authors declare that they have no competing interests.
Transparency: The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained.
Ethical Statement: This study followed all ethical practices during writing.
Publisher: Innovative Research Publishing

1. Introduction

The problem associated with dimensionality reduction is realizing smaller variables while still retaining most of the important information contained in the data. Principal Component Analysis (PCA) is a multivariate analysis method often used for dimension reduction [1]. PCA is designed for reducing data space dimensions and selecting correlated data into new

uncorrelated ones using linear algebra techniques. Even though the dimensions become smaller, the information contained in the dataset is retained since the variance is maintained at 70-80 % [2, 3].

Cluster analysis is as a useful technique for determining patterns in data sets by grouping variables [4]. One Popular method for partitioning data into multiple groups is K-means clustering. This non-hierarchical classification method groups similar data points together and separates those with different characteristics into distinct clusters [5, 6].

Prediction is an important field that entails future events estimation using past and current data, as well as interpreting it into a mathematical model [7]. This field has a wide range of practical applications, including financial forecasting, weather prediction, and trend analysis. There are two general methods for making prediction: qualitative and quantitative. The qualitative technique is inherently intuitive and employs experts' opinions, thereby making mathematical calculations impossible. In contrast, the quantitative technique is based on historical data to make prediction and is suitable for mathematical calculations. The quantitative method is often used for predictions through time series analysis.

Time series data consists of one object covering several periods. Predictions with time series data need to consider the patterns used [8, 9]. One of the most promising prediction methods that are recently utilized is Recurrent Neural Network (RNN) due to its wide application for prediction and very high generalization performance. The basic idea is to create a network topology capable of representing sequential or time series data [10, 11].

The prediction method used by Cao, et al. [12] has also been implemented on a number of reviews [12] utilized the k-means clustering and Gate Recurrent Unit (GRU) methods for prediction. The most influential variables in their study were selected using principal component analysis (PCA), then time series data were grouped based on the same variance using k-means clustering with GRU as the prediction model. The results showed that the method achieved better predictive accuracy and flexibility as compared to the PCA-Long Short-Term Memory(LSTM), PCA-Extreme Learning Machine (ELM), and PCA-Convolutional Neural Network(CNN) models.

The study conducted by He and Zhang [1] proposed a prediction model based on PCA and *Backpropagation* (BP). PCA is used to lessen the variable dimension on data and eradicate the correlation between variables. The obtained main component is used as input vector on BP neural network. Neural network model PCA-BP results in the prediction with highest accuracy, and PCA gives the ability of generalization better than *Multiple Linear Regression* (MLR) method.

Based on the aforementioned studies, the PCA method and k-means grouping are able to optimize the generalization performance of predictions. Therefore, this current study proposes an RNN prediction model as one of the neural network architectures. The purpose is to predict with an optimized RNN method using the k-means grouping and the PCA dimension reduction method by comparing the Euclidean, Manhattan, Canberra, Average, and Chebyshev distance functions.

Air quality data was used because pollution in big cities is a threat to the environment and to the public health. The World Health Organization (WHO) presents the negative effects on health, stating that around 4.2 million people die yearly. Furthermore, air pollution also detrimental effects on the environment, such as smog, acid rain, ozone layer depletion, and global warming. Therefore, an effective solution is required for monitoring and predicting air pollution efficiently [13, 14].

2. Materials and Methods

The process of data analysis used in this study is explained below.

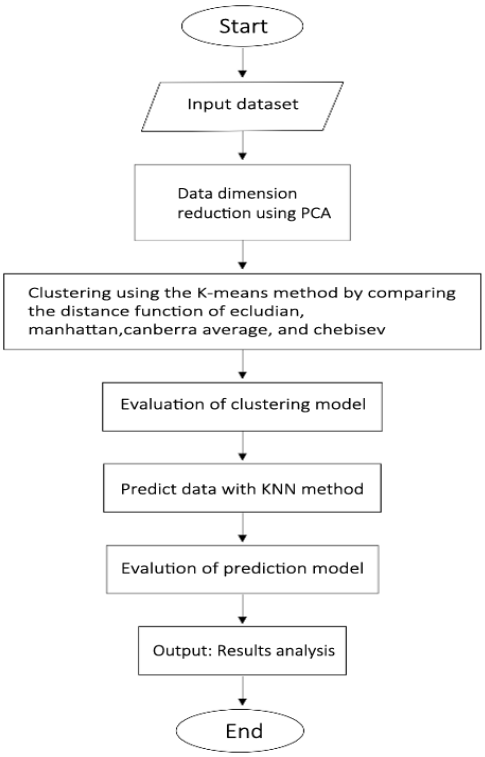

Figure 1. Flowchart of methodology research.

Whereas the method that is used is explained Figure 1.

2.1. Principal Component Analysis (PCA)

The PCA is an unsupervised multivariate analysis method used for lowering high-dimensional data without significantly reducing its characteristics [15, 16]. In addition, the analysis was performed to obtain the main component that was able to maintain 70-80% of the variance in the dataset. Its computation was based on eigenvalues and eigenvectors' calculations showing the data spread magnitude in a dataset [17]. The general stages of PCA dimension reduction are as follows.

2.1.1. Covariance Matrix

Covariant matrix $(\boldsymbol{S})$ is a matrix contained with variant between variables. Each variable's average value (mean) in the form of mean matrix $(\boldsymbol{\mu})$ using order of $\boldsymbol{m \times 1}$. The mean matrix $(\boldsymbol{\mu})$ was obtained by dividing the transpose of the $\boldsymbol{X^T}$ number of data $\boldsymbol{n}$ is given by Equation 1:

$$\boldsymbol{\mu} = \frac{1}{n}\boldsymbol{X}^T\boldsymbol{1} \tag{1}$$

The next step is to calculate the variance of $(\boldsymbol{s})$ matrix data $\boldsymbol{X}$ by changing matrix $(\boldsymbol{\mu})$ into matrix $\boldsymbol{x}$ with order of $\boldsymbol{m \times 1}$ by the equation stated below.

$$\boldsymbol{x} = \boldsymbol{\mu}\boldsymbol{1}^T$$

The initial step in finding the variance is by determining the summed squared deviation **(SS)** of diagonal matrix just like written in the following Equation 2.

$$\mathbf{SS} = [\boldsymbol{X}^T - \boldsymbol{x}][\boldsymbol{X}^T - \boldsymbol{x}]^T \tag{2}$$

Equation 3 presents **SS** that will be used in calculating variance by the following formula:

$$s = \frac{1}{n-1}\,\mathbf{SS} \tag{3}$$

2.1.2. Eigenvalues and Eigenvectors

If $(\boldsymbol{S} - \boldsymbol{\lambda I})$ is a singular matrix and $\boldsymbol{S}$ is a covariance matrix, then the eigenvalues are obtained with the following Equation 4:

$$det(\boldsymbol{S} - \boldsymbol{\lambda} I) = 0 \tag{4}$$

Also, the eigenvector is obtained using the following Equation 5.

$$(\boldsymbol{S} - \boldsymbol{\lambda} I)\boldsymbol{a} = 0 \tag{5}$$

2.1.3. Principal Component Analysis

The Principal Components (PC) were obtained by Equation 6 and formed based on a linear combination of the matrix $\boldsymbol{a}$, eigenvector, and data matrix $\boldsymbol{X}$ as follows:

$$PC_m = \sum_{j=1}^{m} \boldsymbol{a}^T\boldsymbol{X}^T = a_{11}X_1 + a_{21}X_2 + \cdots + a_{m1}X_m \tag{6}$$

2.2. K-means Clustering

The k-means is a fairly simple clustering algorithm that partitions data into one cluster to minimize similarities. According to Sinaga and Yang [18]; Yuan and Yang [19] the similarity measure used in the cluster is the distance function. The k-means algorithm principle was to minimize an objective function by determining the initial cluster center and then iteratively improving it until no significant change occurs. The cluster center c_{ij} was obtained by dividing the data's number x_i in a class by the number of classes n and given by Equation 7:

$$c_{ij} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{7}$$

For example, if p is the last iteration, then the above equation is expected to satisfy the following conditions.

$$c_{ij} = c_{ij-1}; 1 \leq n \leq p$$

The distance functions used for measuring the clusters' similarity include Euclidean, Manhattan, Canberra, Average, and Chebyshev.

2.2.1. Euclidean Distance

The Euclidean was calculated to measure the data point's distance x_i with the cluster center b_i in Euclidean space, examining the relationship between angles and distances [20]. The Euclidean distance is obtained by Equation 8.

$$d(x,b) = \sqrt{\sum_{i=1}^{n}(x_i - b_i)^2} \tag{8}$$

2.2.2. Manhattan Distance

The Manhattan is a method of calculating distance in space by applying the absolute difference concept to the data x_i and cluster center b_i [21]. The Manhattan distance is calculated using Equation 9.

$$d(x,b) = \sum_{i=1}^{n} |x_i - b_i| \tag{9}$$

2.2.3. Canberra Distance

For each value of the two vectors to be matched, the Canberra distance divides the absolute difference between the data x_i and the cluster center b_i by the absolute sum of the two values. Furthermore, the results of the two matched values were added to obtain the Canberra distance [22] and is presented below in Equation 10.

$$d(x,b) = \sum_{i=1}^{n} \frac{|x_i - b_i|}{|x_i| - |b_i|} \tag{10}$$

2.2.4. Average Distance

The average simply represents the mean distance calculation between the data x_i and the cluster center b_i which is shown in Equation 11.

$$d(x,b) = \sqrt{\frac{1}{p}\sum_{i=1}^{n}(x_i - b_i)} \; ; \; p = sum\ of\ cluster \tag{11}$$

2.2.5. Chebyshev Distance

The Chebyshev distance Equation 12 is used to determine the maximum distance value that measures the absolute magnitude of the difference between the data x_i and the cluster center b_i [23].

$$d(x,b) = max_c|x_i - b_i| \; ; for\ i = 1,2,..,n \tag{12}$$

The clustering data was then evaluated using the Davies Bouldin Index (DBI) model in order to maximize the inter-cluster distance known as separation value and minimize that of intra-cluster, called compactness value [24, 25].

If m is the cluster center and S is the distance, then the similarities between clusters R_{ij} were calculated using the following equation.

$$R_{ij} = \frac{S_i + S_j}{|m_i - m_j|} \tag{13}$$

In which, R_{ij} is a measure of similarity between cluster C_i and C_j. DBI value indicates the amount of optimum cluster through the following notion.

$$R_i = max_{i \neq j}\, R_{ij} \tag{14}$$

Hence, DBI was calculated using the equation below.

$$DB = \frac{1}{n_c}\sum_{i=1}^{n_c} \cdot max_{i \neq j}\, R_{ij} \tag{15}$$

2.3. Recurrent Neural Network (RNN)

A Recurrent Neural Network (RNN) is part of the NN for sequential data processing. This is included in deep learning because the data goes through several layer of processing [26]. In general, the training process is similar to that of neural networks and has three main steps. The first step is to make forward passes and predictions. Each of hidden state (h_t) is calculated based on each input (x_t) and a weight that has been predetermined. After the hidden state value is obtained, the second step is to compare the actual output value (target) with the predicted result (z_t) using a loss function. The loss function produces an error value that indicates whether the prediction results are accurate or not, thereby permitting a decision to be made to evaluate the RNN performance. In the final step, the Backpropagation Through Time (BPTT) process is conducted on the error value obtained from loss function which is then used to calculate the gradient for each time step in the network. The goal of BPTT is to determine better weights and biases as compared to the previous process. The weight and bias are then updated using the Stochastic Gradient Descent (SGD) method.

2.3.1. Feedforward

In feedforward training, the hidden state and output values were calculated through the following equations.

$$h_t = f(w_{hh}\, h_{t-1} + w_{xh}\, x_t + b_h) \tag{16}$$

$$z_t = w_{hz}\, h_t + b_z \tag{17}$$

$$\hat{y}_t = f(z_t) \tag{18}$$

Where:

x_t : Input value.

h_t : Hidden state value at t.

h_{t-1} : The previous hidden state value.
z_t : Output layer.
$\hat{y}_t$: Output result.
f : Non-linear activation function.
b_h : Bias value at hidden layer.
b_z : Bias value at output layer.
w_{xh} : Weight value on input to hidden.
w_{hh} : Weight value on hidden to hidden.
w_{hz} : Weight value on hidden to output.

When training the feedforward model, a non-linear activation function is needed to convert an input into output. Mathematically, the activation function is the sum of the input and the weighted bias. The activation function used is as follows.

2.3.2. Rectified Linear Unit (ReLU)

ReLU is an activation function whose value has a range $[0, \infty]$ [27]. The equation for the ReLU function includes.

$$f(x) = \begin{cases} 0 \; for \; x \leq 0 \\ x \; for \; x > 0 \end{cases} \tag{19}$$

2.3.3. Softmax

Another form of Logistic Regression algorithm used for classifying more than two classes is Softmax [28]. The form of the softmax activation equation is expressed as follows.

$$f(\hat{y}_i) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}; for \; j = 1,2, \ldots, K \tag{20}$$

The derivation for $i = n$ is:

$$\frac{\partial \hat{y}_i}{\partial z_n} = \frac{e^{z_j} \sum_{k=1}^{K} e^{z_k} - e^{z_j} e^{z_n}}{(\sum_{k=1}^{K} e^{z_k})^2}$$

$$= \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \left(1 - \frac{e^{z_n}}{\sum_{k=1}^{K} e^{z_k}}\right)$$

$$= \hat{y}_i (1 - \hat{y}_n)$$

$$= \hat{y}_n (1 - \hat{y}_n)$$

The derivation for $i \neq n$ is:

$$\frac{\partial \hat{y}_i}{\partial z_n} = \frac{0 \sum_{k=1}^{K} e^{z_k} - e^{z_j} e^{z_n}}{(\sum_{k=1}^{K} e^{z_k})^2}$$

$$= -\frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \frac{e^{z_n}}{\sum_{k=1}^{K} e^{z_k}}$$

$$= -\hat{y}_i \hat{y}_n$$

In which e is exponential.

The output obtained during the feedforward training was then compared with the target value via a loss function, which is a categorical cross-entropy used in multi-class classification problems [29]. If y_i is the target value and $\hat{y}_k$ is the output result, then the categorical cross-entropy is calculated using the following equation.

$$L(y_i, \hat{y}_t) = -\left(\sum_{i=1}^{n} y_i \log \hat{y}_t\right); for \; k = 1,2, \ldots, c \tag{21}$$

$$\frac{\partial L}{\partial z_n} = -\sum_{i=1}^{n} y_i \frac{\partial \log \hat{y}_t}{\partial z_n}$$

$$= -\sum_{i=1}^{n} y_i \frac{\partial \log \hat{y}_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial z_n}$$

$$= -\sum_{i=1}^{n} y_i \frac{1}{\hat{y}_t} \frac{\partial \hat{y}_t}{\partial z_n}$$

$$= -\sum_{i=1}^{n} \frac{y_i}{\hat{y}_t} \frac{\partial \hat{y}_t}{\partial z_n}$$

2.3.4. Backpropagation Throught Time (BPTT)

Backpropagation Through Time (BPTT) is an algorithm used for changing the RNN weight value. Backpropagation refers to two things: the first is a mathematical method used for calculating derivatives and applying chain rules, while the second is a training algorithm for updating network weights in order to minimize errors. According to Lillicrap and Santoro [30] , BPTT is calculated by obtaining the gradients from the error or loss value until the weight and bias parameters change. The calculation of weight gradient of w_{hz}, w_{hh}, w_{xh} can be done by using the following equation.

$$\frac{\partial L}{\partial w_{hz}} = \frac{\partial L}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial z_t} \frac{\partial z_t}{\partial w_{hz}} \tag{22}$$

$$\frac{\partial L}{\partial w_{hh}} = \sum_{k=1}^{t-1} \frac{\partial L}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial z_t} \frac{\partial z_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial h_{t-1}} \frac{\partial h_{t-1}}{\partial w_{hh}} \tag{23}$$

$$\frac{\partial L}{\partial w_{xh}} = \sum_{k=1}^{t-1} \frac{\partial L}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial z_t} \frac{\partial z_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial h_{t-1}} \frac{\partial h_{t-1}}{\partial w_{xh}} \tag{24}$$

Bias gradient of b_z and b_h by the following equation.

$$\frac{\partial L}{\partial b_z} = \frac{\partial L}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial z_t} \frac{\partial z_t}{\partial b_z} \tag{25}$$

$$\frac{\partial L}{\partial b_h} = \sum_{k=1}^{t-1} \frac{\partial L}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial z_t} \frac{\partial z_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial h_{t-1}} \frac{\partial h_{t-1}}{\partial w_{xh}} \tag{26}$$

2.3.5. Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (SGD) is used for reducing high loss values and requires a running backpropagation after training. In addition, the SGD helps to overcome high loss values by updating the weight and bias parameters after backpropagation [31]. The SGD equation is expressed as follows.

$$w_{new} = w_{old} - \alpha \frac{\partial L}{\partial w} \tag{27}$$

Where:

w_{new} : New weight.

w_{old} : Old weight.

α : *Learning rate.*

$\frac{\partial L}{\partial w}$: *Loss function* gradient on weight and bias.

2.4. Prediction Model Evaluation

The built prediction model is expected to identify the number of errors. Several methods for evaluating the prediction model are as follows.

2.4.1. Mean Absolute Percentage Error (MAPE)

MAPE is a method used to calculate prediction model's accuracy. The results are categorized as very good when they have a MAPE value of less than 10% [32]. If y_i is the actual data and $\hat{y}_i$ is a prediction result, then MAPE is calculated using the equation below.

$$MAPE = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\% \tag{28}$$

2.4.2. Root Mean Square Error (RMSE)

The RMSE value is determined by squaring the difference between the actual data and the predicted results, divided by the number of data. When the RMSE is close to zero, then the prediction results are more accurate [33].

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \quad (29)$$

3. Results and Discussion

This study used air quality data from October 2019 to July 2022 in Sakurakoiji-Yurihonjo, Akita, Japan. The dataset in Table 1 is consisted of 889 data with variables PM_{25}, PM_{10}, O_3, NO_2, CO obtained from the Air Quality Historical Data Platform's (aqicn.org) official website.

Table 1.
Air quality at cc, Akita, Japan.

Date	PM_{25}	PM_{10}	O_3	NO_2	CO
2022-07-01	36.0	17.0	27.0	1.00	1.00
2022-07-02	31.0	NaN	23.0	1.00	1.00
2022-07-03	18.0	NaN	26.0	1.00	1.00
2022-07-04	18.0	17.0	27.0	1.00	1.00
2022-07-05	24.0	4.00	23.0	1.00	1.00
⋮	⋮	⋮	⋮	⋮	⋮
2019-10-31	43.0	20.0	46.0	2.00	3.00

Whereas the input variables that are used are:

PM_{25} : Air particle smaller than 2.5 micrometer.

PM_{10} : Air particle smaller than 10 micrometer.

O_3 : One of the secondary polutant formed by photochemistry reaction.

NO_2 : Polutant formed by vehicle's fuel, burning trash, coal, and industry waste.

CO : Carbon monoxide contained in air.

This part explains the calculation process of PCA that results in output in the form of main component, in which the main component is used as input in the calculation process of k-*means*. The obtained output or supervised data is used as input in the calculation of RNN prediction.

3.1. Principal Component Analysis (PCA)

The initial step of data processing using PCA dimension reduction is by doing normalization on the data purposed to place the value on each variable ranged from 0 until 1. The normalization results then being changed into matrix form $\boldsymbol{X}$ are stated below.

$$\boldsymbol{X} = \begin{bmatrix} -0.13273 & 1.52317 & -1.06672 & -0.76750 & -0.42666 \\ 0.56705 & 0.10002 & -0.09415 & -0.76750 & 2.34378 \\ -0.74505 & -0.75387 & -1.87720 & -0.76750 & -0.42666 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -0.48263 & -1.32313 & -0.41834 & 2.07545 & -0.42666 \\ -0.22021 & 1.52317 & 0.39212 & 0.65397 & -0.42666 \\ 0.82947 & 2.37707 & 1.36469 & 2.07545 & -0.42666 \end{bmatrix}$$

The next step is to calculate the covariant matrix ($\boldsymbol{S}$) based on the matrix $\boldsymbol{X}$ presented above. The result of covariant matrix ($\boldsymbol{S}$) is obtained by equation 4, the result is written as follows.

$$\boldsymbol{S} = \begin{bmatrix} 1.00205 & 0.45584 & 0.35257 & 0.07057 & -0.03381 \\ 0.45584 & 1.00205 & 0.40885 & -0.01653 & 0.03443 \\ 0.35257 & 0.40885 & 1.00205 & 0.05194 & -0.02465 \\ 0.07057 & -0.01653 & 0.05194 & 1.00205 & -0.04209 \\ -0.03381 & 0.03443 & -0.02465 & -0.04209 & 1.00205 \end{bmatrix}$$

Covariant matrix ($\boldsymbol{S}$) shows the relation between variables that is analyzed using PCA. Based on matrix S on the above, it can be observed that each variable is related to each other. The eigen value then is obtained from covariant matrix ($\boldsymbol{S}$), and sorted from the highest value ($\lambda_1 > \lambda_2 > \lambda_3 > \lambda_4 > \lambda_5$) stated below.

$$\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \end{bmatrix} = \begin{bmatrix} 1.81967 \\ 1.05446 \\ 0.95891 \\ 0.65525 \\ 0.52198 \end{bmatrix}$$

Based on the above eigen value, there are five eigen vectors obtained that are compatible with each other are presented below.

$$\boldsymbol{a} = \begin{bmatrix} -0.57769 & -0.56540 & -0.58700 & -0.04447 & -0.00525 \\ -0.59602 & 0.76704 & -0.16448 & 0.16644 & -0.04052 \\ -0.55243 & -0.25691 & 0.79253 & -0.01611 & -0.02079 \\ -0.07389 & 0.12885 & -0.00531 & -0.69164 & 0.70677 \\ 0.01924 & -0.09673 & 0.01478 & 0.70120 & 0.70594 \end{bmatrix}$$

The obtained eigen value is used to calculate the value of Variance Proportion Cumulative (VPC) which has a purpose to determine the amount of main component that is chosen. VPC is capable in explaining the data's total variance up to 70-80%.

The result of value calculation and the proportion of total variance is shown below.

Table 2.
Eigen value and the variance amount.

Component	Eigen value	Variance amount (%)	VPC (%)
1	1.82	36.3	36.3
2	1.06	21.1	57.4
3	0.96	19.1	76.5
4	0.66	13.1	89.6
5	0.52	10.4	100

Table 2 presents three new variables or main components that are derived from the analyzed variables. The ability of each component in representing analyzed variables is shown by the variance value and explained VPC. Variance is represented as eigen value. The eigen value shows the contribution of main component on the variance of all analyzed original variable. Table 2 also shows the three new variables or main components with the highest eigen value. The first main component has an eigenvalue of 1.82 (representing 36.3% of the variance); the second main component has an eigenvalue of 1.06 (representing 21.0% of the variance), and the third main component has an eigenvalue of 0.96 (representing 19.1% of the variance). Together, those three main components are able to withstand data variance as big as 76.5%, as shown by the cumulative value. The eigenvalue graph is depicted in the following figure.

The figure below represents the eigenvalues' graph.

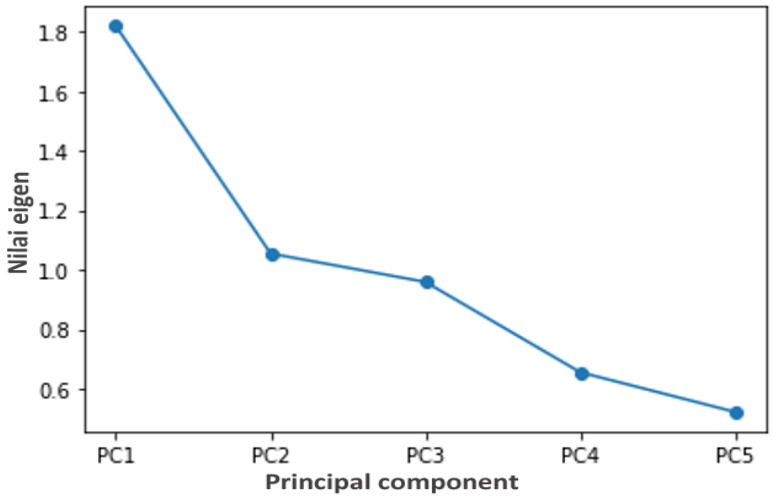

Figure 2.
Eigenvalue.

Figure 2 shows the sorted eigen value that is used in determining the amount of main component. According to the chart, there are three largest eigen value ($\lambda_1 > \lambda_2 > \lambda_3 > 0$) that shows the amount of main component which are two components.

The determination of main component can also be achieved by using VPC which can explain the total amount of data variance up to 70-80%. VPC can be calculated by using arranged variance on eigen value matrix or by summing up the variance proportion of each main component. VPC value can be calculated by following formula.

$$VPC = \left(\frac{\sum_{j=1}^{5} \lambda_j}{5.01027} \times 100\% \geq 70\%, \qquad j = 1,2,3 \right)$$

$$= \frac{\lambda_1 + \lambda_2}{5.01027} \times 100\% = 76{,}50\%$$

Based on the said calculation, it is found that VPC ≥ 70% corresponds to a value of 74.06%, indicating that the number of main components formed are two. The amount of variance significantly influences the number of main components that can be based on either the amount of variance proportion or eigen value.

The previously obtained eigen vector is then used to determine which variable are included in the result of three main components. The eigen vector obtained from three largest eigenvalues shows the *loading*/coefficient value of the three formed components. *Loading*/coefficient represents the variance a variable can be explained by a main component. A large variance indicates that the variable has a significant correlation (influence) on the main component, which represents the data variables that influences air quality. The *loading*/coefficient table of the three main components are:

Table 3.
Loading/Coefficient.

Variable	Component 2	Component 2	Component 3
1	-0.58	-0.05	-0.01
2	-0.60	0.17	-0.04
3	-0.55	-0.02	-0.02
4	-0.07	-0.69	0.71
5	0.02	0.70	0.71

Table 3 explains the relation between original variable with the new variable (main component) formed by PCA. Among the formed three main components, the one with highest *loading*/coefficient is selected (highlighted in bold) in which these values can explain the variable that affects the air quality, while variables with smaller *loading*/coefficient value are regarded as less influential in the formation of the main component. The two new variables (main component) that have been formed represent the five variables from the original data. The results of PC (Principal Component) from the data above are being presented below in Table 4.

Based on the obtained eigen value, the three main components are:

Table 4.
Principal component value.

PC1	PC2	PC3
-0.19	0.51	-0.88
-0.23	2.17	1.11
1.97	0.17	-0.77
:	:	:
-2.81	-1.40	1.04

The previously obtained eigen vector is used to determine which variable that is included in the three main components. The obtained three main components are then used as *input* in the clustering of k-*means.*

3.2. K-Means

After reducing the dimension with PCA, the next step is to do a clustering by using k-*means*.

A total of six cluster classes were used in this study. The clusters' quantity selected was based on the number of air quality indicators shown in Table 5.

Table 5.
Air quality indicator.

Class	Indicator
0	Safe
1	Moderate
2	Unhealthy
3	Unhealthy sensitive group
4	Dangerous
5	Very dangerous

The next step was to form the center of early cluster which would mark the average location for each cluster. The center of early cluster is written as below.

$$c_{ij} = \begin{bmatrix} -2.16618 & 0.46946 & 3.01787 \\ 0.43883 & 0.22002 & -0.83214 \\ -0.03964 & 1.01618 & 2.14419 \\ 0.34085 & -0.74118 & 0.15986 \\ 0.43303 & 2.27043 & 1.10821 \\ -0.06086 & -1.81189 & 1.17437 \end{bmatrix}$$

It can be observed that the cluster center obtained is still less accurate in the initial conditions, hence cluster center formation was calculated until its value remain the same. The obtained cluster center from the calculation of the Euclidean, Manhattan, Canberra, Average, and Chebyshev distance functions until it converges (i.e., cluster center does not change) is as follows.

3.2.1. Euclidean

The clustering by Euclidean distance function reaches the convergent on the 18th iteration. The change of cluster center is stated as below.

The 1st iteration

$$c_{ij} = \begin{bmatrix} 0.23891 & -0.89053 & 0.24332 \\ 0.32566 & 0.59473 & 0.48852 \\ -2.51717 & 0.28712 & -0.24308 \\ -1.46614 & -0.02140 & -0.13171 \\ -0.21103 & -1.77450 & 1.17092 \\ 0.93541 & 0.17998 & -0.80419 \end{bmatrix}, i = 1,2,..,6; j = 1,2,3$$

The 18th iteration

$$c_{ij} = \begin{bmatrix} 0.46325 & -0.79027 & 0.16971 \\ 0.08192 & 1.72942 & 1.58335 \\ -2.21486 & -0.33482 & -0.20689 \\ -0.30742 & 0.26469 & -0.85840 \\ 0.31567 & -1.80208 & 1.19093 \\ 1.45926 & 0.12579 & -0.76815 \end{bmatrix}, i = 1,2,..,6; j = 1,2,3$$

By using the same process, the clustering is done for Manhattan, Canberra, Average, and Chebyshev by using distance function until reaches the convergent written in the following lines.

3.2.2. Manhattan

The 12th iteration

$$c_{ij} = \begin{bmatrix} 0.10764 & 1.81359 & 0.49755 \\ -1.43314 & -1.02799 & 0.51301 \\ 1.43655 & 0.14684 & -0.79062 \\ 0.80731 & -1.07429 & 0.52761 \\ -0.28655 & 0.26214 & -0.85751 \\ -2.43947 & 0.35409 & -0.77278 \end{bmatrix}, i = 1,2,..,6; j = 1,2,3$$

3.2.3. Canbera

The 12th iteration

$$c_{ij} = \begin{bmatrix} 1.08847 & 0.15834 & -0.80238 \\ 1.11641 & 0.78097 & 1.52072 \\ -0.99388 & 1.81442 & 1.54473 \\ 0.72295 & -0.90876 & 0.29051 \\ -1.37932 & -1.09250 & 0.51851 \\ -1.07853 & 0.32762 & -0.89023 \end{bmatrix}, i = 1,2,..,6; j = 1,2,3$$

3.2.4. Average

The 9th iteration

$$c_{ij} = \begin{bmatrix} -1.41132 & -1.05029 & 0.48275 \\ -0.23700 & 0.25992 & -0.85572 \\ -2.37960 & 0.31183 & -0.85945 \\ 1.48006 & 0.13582 & -0.77787 \\ 0.78602 & -1.10015 & 0.47754 \\ 0.08192 & 1.72942 & 1.58335 \end{bmatrix}, i = 1,2,..,6; j = 1,2,3$$

3.2.5. Chebyshev

The 22nd iteration

$$c_{ij} = \begin{bmatrix} -0.33573 & 0.12100 & -0.71487 \\ -1.06439 & 1.79892 & 1.57165 \\ 1.50661 & 0.01173 & -0.64892 \\ 0.52437 & -1.12980 & 0.50352 \\ 0.88956 & 1.68045 & 1.59159 \\ -2.17509 & -0.41070 & -0.13254 \end{bmatrix}, i = 1,2,..,6; j = 1,2,3$$

Based on the calculation of the distance functions of Euclidean, Manhattan, Canberra, Average, and Chebyshev, the DBI is then calculated to observe which distance function can cluster distance between objects minimally under one cluster. Using Python software, the DBI value on each distance function is obtained as follows.

Table 6.
DBI value in distance function.

Distance	DBI	Number of convergent iterations
Euclidean	0.62	18
Manhattan	0.61	12
Canbera	0.92	12
Average	0.61	9
Chebysev	0.61	22

Table 6 shows the DBI values obtained for each distance function. The DBI value for grouping using the average function produces the lowest value of 0.61 and with only 9 iterations reaches convergence.Thus, the most optimal distance for data grouping is the average function. The k-means grouping results using the average function are then used as input to the RNN prediction model.

The result of k-*means* clustering by average *distance* function is shown on the following Table 7:

Table 7.
Principal component (PC).

PC1	PC2	PC3	Lable
-0.19	0.51	-0.88	3
-0.23	2.17	1.11	1
1.96	0.17	-0.77	5
-0.12	1.28	2.06	4
⋮	⋮	⋮	⋮
-2.81	-1.40	1.04	0

3.3. Recurrent Neural Network (RNN)

After clustering using k-*means*, the next step is to form a prediction model using RNN. The parameter initialization which is used includes three parameters under these following scenarios:

Epoch : 100

Neuron hidden : 100, 70, 50, 20, 6

Batch size : 32

Learning rate : 0.01

Loss function : Categorical crossentropy

The values obtained from each parameter are based on prior studies that utilized the same method. The parameter of *max epoch*, *hidden neurons* , and *batch size* refer to a study by Jeong, et al. [34]. While the *learning rate* value and the *loss binary cross entropy function refer to other studies such as* [35] and a result of the writer's own elaboration. There is no set rule for determining parameter, but by experiments using data and model that to be used are necessary. The parameter of previous studies can serve as reference but a preliminary experiment is still needed to identify the optimal conditions.

Prior to training, the dataset was divided into 80% training and 20% test data, resulting in 389 and 98 instances, respectively. The training data were used to adjust the machine learning model, while the result was evaluated with the test data.

The model was trained using python software with 100 epochs. Table 5 shows the changes in the loss value and training accuracy.

Table 8.
Loss value and accuracy.

Epoch	Loss	Accuracy (%)
1	1.83	13.3
2	1.79	16.7
3	1.78	21.1
4	1.76	23.4
5	1.74	31.6
6	1.80	37.8
7	1.69	40.4
8	1.66	44.5
⋮	⋮	⋮
100	0.28	91.0

Table 8 shows that the minimum loss value was 0.28 in the 100th iteration with 91% accuracy. Figure 3 shows the loss value graph and the accuracy results.

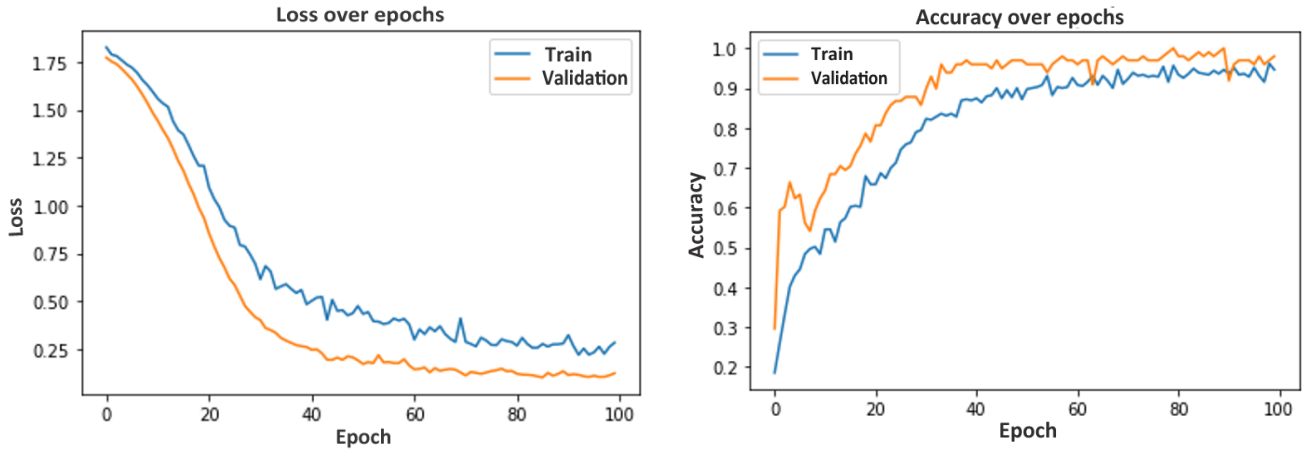

Figure 3.
Loss value and accuracy.

It was observed that the average accuracy for 100 epochs was 95.67%. Furthermore, the predicted data obtained were compared with the actual data to obtain the prediction model accuracy. Figure 4 shows the comparison of the actual data and the predicted results.

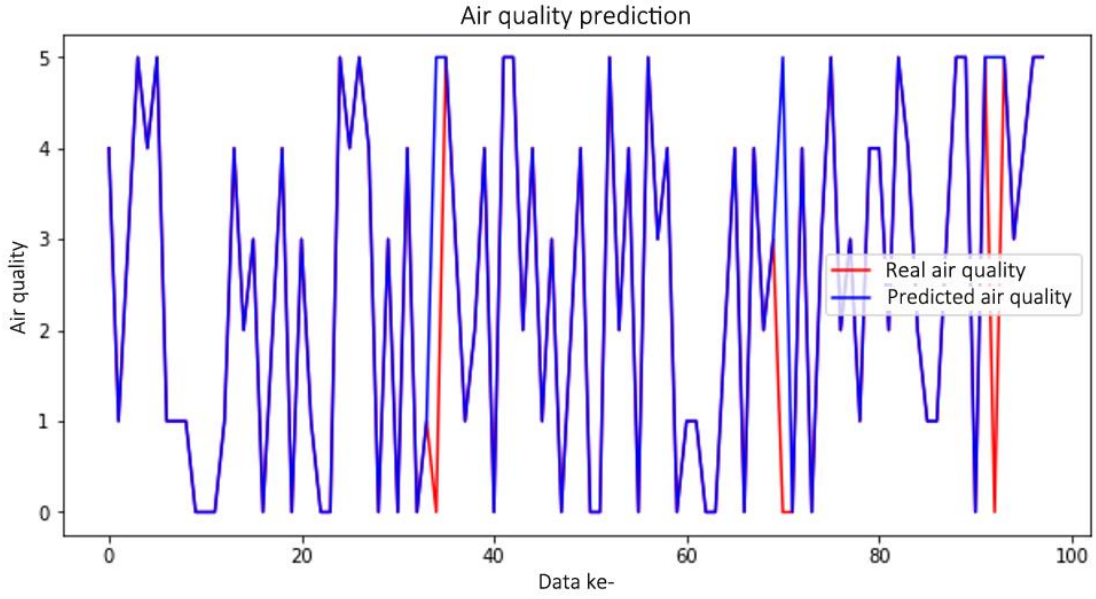

Figure 4.
Comparison of actual and predicted data.

Based on the figure, the red color represents the actual data while the blue color represents the predicted results. The prediction model results were analyzed by evaluating the RMSE and MAPE error metric models. The calculation result for the number of errors in the prediction model using MAPE was 8.62%, while that of RMSE was 0.83.

4. Conclusion

The results obtained from the PCA calculation are three new variables out of the five examined. The first, second, and third principal components have variances of 36.31%, 21.04%, and 19.13% respectively. Consequently, their cumulative variance was 76.50%, indicating that only three variables are necessary to explain the factors affecting air quality. The clustering of k-*means using distance function of Euclidean*, *Manhattan*, *Canberra*, *average*, and *Chebyshev* is evaluated using DBI method. The obtained DBI value for each distance function are as follows: *Euclidean* with 0.62 and reached convergent state on the 18th iteration, *Manhattan* with 0.61 and reached convergent state on the 12th iteration, *Canberra* with 0.92 and reached convergent state on the 12th iteration, *average* with 0.61 and reached convergent state on the 9th iteration, and *Chebyshev* with 0.61 and reached convergent state on the 22nd iteration. The smaller the obtained DBI value (*non*-negative = 0), will result in better cluster obtained from k-*means* clustering. The *Average* distance function had the lowest DBI value and was the fastest in reaching the convergent state as compared with *Euclidean*, *Manhattan*, *Canberra*, and *Chebyshev*. Thus, it can be concluded that k-*means* clustering using average distance function is better than *Euclidean*, *Manhattan*, *Canberra*, and *Chebyshev*. The prediction model results, evaluated using the RMSE error metric produced 0.83 total error, while that of MAPE was 8.62%. Therefore, the optimized RNN prediction method using k-means clustering and PCA dimension reduction yielded good predictive model capabilities.

Based on the analysis and conclusion, it is recommended to conduct further studies need using different dimensional reduction methods, such as Fisher Discriminant Analysis (FDA) or Rough-Set, and grouping methods, e.g. Fuzzy C-Means (FCM). By using different methods, it may be possible to obtain better result in terms of clustering and predictive capabilities. Additionally, it is important to consider the availability and quality of data when selecting a suitable method. Therefore, future studies should explore various techniques to improve the accuracy and efficiency of air quality prediction models

References

[1] F. He and L. Zhang, "Prediction model of end-point phosphorus content in BOF steelmaking process based on PCA and BP neural network," *Journal of Process Control*, vol. 66, pp. 51-58, 2018. https://doi.org/10.1016/j.jprocont.2018.03.005

[2] Y. Aït-Sahalia and D. Xiu, "Principal component analysis of high-frequency data," *Journal of the American Statistical Association*, vol. 114, no. 525, pp. 287-303, 2019.

[3] D. Granato, J. S. Santos, G. B. Escher, B. L. Ferreira, and R. M. Maggio, "Use of principal component analysis (PCA) and hierarchical cluster analysis (HCA) for multivariate association between bioactive compounds and functional properties in foods: A critical perspective," *Trends in Food Science & Technology*, vol. 72, pp. 83-90, 2018. https://doi.org/10.1016/j.tifs.2017.12.006

[4] P. Govender and V. Sivakumar, "Application of k-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019)," *Atmospheric Pollution Research*, vol. 11, no. 1, pp. 40-56, 2020. https://doi.org/10.1016/j.apr.2019.09.009

[5] F. Wang, H. H. Franco-Penya, J. D. Kelleher, J. Pugh, and R. Ross, "An analysis of the application of simplified silhouette to the evaluation of k-means clustering validity," in *International Conference on Machine Learning and Data Mining in Pattern Recognition. Springer, Cham*, 2017, pp. 291-305.

[6] D. Q. Zeebaree, H. Haron, A. M. Abdulazeez, and S. Zeebaree, "Combination of K-means clustering with genetic algorithm: A review," *International Journal of Applied Engineering Research*, vol. 12, no. 24, pp. 14238-14245, 2017.

[7] R. Fildes, S. Ma, and S. Kolassa, "Retail forecasting: Research and practice," *International Journal of Forecasting*, vol. 38, no. 4, pp. 1283-1318, 2022. https://doi.org/10.1016/j.ijforecast.2019.06.004

[8] J. F. Torres, A. Galicia, A. Troncoso, and F. Martínez-Álvarez, "A scalable approach based on deep learning for big data time series forecasting," *Integrated Computer-Aided Engineering*, vol. 25, no. 4, pp. 335-348, 2018. https://doi.org/10.3233/ica-180580

[9] F. J. Baldán and J. M. Benítez, "Distributed FastShapelet Transform: A Big Data time series classification algorithm," *Information Sciences*, vol. 496, pp. 451-463, 2019. https://doi.org/10.1016/j.ins.2018.10.028

[10] T. Donkers, B. Loepp, and J. Ziegler, "Sequential user-based recurrent neural network recommendations," in *RecSys 2017 - Proceedings of the 11th ACM Conference on Recommender Systems*, 2017, pp. 152–160.

[11] X.-Y. Zhang, F. Yin, Y.-M. Zhang, C.-L. Liu, and Y. Bengio, "Drawing and recognizing chinese characters with recurrent neural network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 849-862, 2017. https://doi.org/10.1109/tpami.2017.2695539

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

[13] D. Glencross, T. Ho, N. Camiña, C. Hawrylowicz, and P. Pfeffer, "Air pollution and its effects on the immune system," *Free Radical Biology & Medicine*, vol. 151, pp. 56-68, 2020.

[14] P. J. Landrigan, "Air pollution and health," *The Lancet Public Health*, vol. 2, no. 1, pp. e4-e5, 2017.

[15] H. Zou and L. Xue, "A selective overview of sparse principal component analysis," *Proceedings of the IEEE*, vol. 106, no. 8, pp. 1311-1320, 2018. https://doi.org/10.1109/jproc.2018.2846588

[16] K. K. Vasan and B. Surendiran, "Dimensionality reduction using principal component analysis for network intrusion detection," *Perspectives in Science*, vol. 8, pp. 510-512, 2016. https://doi.org/10.1016/j.pisc.2016.05.010

[17] S. Surono and R. D. A. Putri, "Optimization of fuzzy c-means clustering algorithm with combination of minkowski and chebyshev distance using principal component analysis," *International Journal of Fuzzy Systems*, vol. 23, no. 1, pp. 139-144, 2021. https://doi.org/10.1007/s40815-020-00997-5

[18] K. P. Sinaga and M.-S. Yang, "Unsupervised K-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80716-80727, 2020. https://doi.org/10.1109/access.2020.2988796

[19] C. Yuan and H. Yang, "Research on K-value selection method of K-means clustering algorithm," *Multidisciplinary Scientific Journal*, vol. 2, no. 2, pp. 226-235, 2019. https://doi.org/10.3390/j2020016

[20] B. Wang, X. Liu, B. Yu, R. Jia, and X. Gan, "An improved WiFi positioning method based on fingerprint clustering and signal weighted euclidean distance," *Sensors*, vol. 19, no. 10, p. 2300, 2019. https://doi.org/10.3390/s19102300

[21] L. Greche, M. Jazouli, N. Es-Sbai, A. Majda, and A. Zarghili, "Comparison between euclidean and Manhattan distance measure for facial expressions classification," in *2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems, WITS 2017*, 2017, pp. 2–5.

[22] M. Kocher and J. Savoy, "Distance measures in author profiling," *Information Processing & Management*, vol. 53, no. 5, pp. 1103-1119, 2017. https://doi.org/10.1016/j.ipm.2017.04.004

[23] É. O. Rodrigues, "Combining Minkowski and Chebyshev: New distance proposal and survey of distance metrics using k-nearest neighbours classifier," *Pattern Recognition Letters*, vol. 110, pp. 66-71, 2018.

[24] J. Xiao, J. Lu, and X. Li, "Davies bouldin index based hierarchical initialization K-means," *Intelligent Data Analysis*, vol. 21, no. 6, pp. 1327-1338, 2017. https://doi.org/10.3233/ida-163129

[25] I. M. K. Karo, A. F. Huda, and K. MaulanaAdhinugraha, "A cluster validity for spatial clustering based on davies bouldin index and polygon dissimilarity function," in *Proceedings of the 2nd International Conference on Informatics and Computing, ICIC 2017*, 2018, pp. 1–6, doi: https://doi.org/10.1109/IAC.2017.8280572

[26] L. Yu, J. Chen, G. Ding, Y. Tu, J. Yang, and J. Sun, "Spectrum prediction based on Taguchi method in deep learning with long short-term memory," *IEEE Access*, vol. 6, pp. 45923-45933, 2018. https://doi.org/10.1109/access.2018.2864222

[27] G. Lin and W. Shen, "Research on convolutional neural network based on improved Relu piecewise activation function," *Procedia Computer Science*, vol. 131, pp. 977-984, 2018. https://doi.org/10.1016/j.procs.2018.04.239

[28] I. Kouretas and V. Paliouras, "Simplified hardware implementation of the softmax activation function," in *2019 8th International Conference on Modern Circuits and Systems Technologies, MOCAST 2019*, 2019, pp. 1–4., doi: https://doi.org/10.1109/MOCAST.2019.8741677

[29] Y. Ho and S. Wookey, "The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling," *IEEE Access*, vol. 8, pp. 4806-4813, 2019. https://doi.org/10.1109/access.2019.2962617

[30] T. P. Lillicrap and A. Santoro, "Backpropagation through time and the brain," *Current opinion in Neurobiology*, vol. 55, pp. 82-89, 2019. https://doi.org/10.1016/j.conb.2019.01.011

[31] J. Yang and G. Yang, "Modified convolutional neural network based on dropout and the stochastic gradient descent optimizer," *Algorithms*, vol. 11, no. 3, p. 28, 2018. https://doi.org/10.3390/a11030028

[32] T. Kalaiselvi and S. Karthigai Selvi, "Improved genetic k-means algorithm comprises mean absolute percentage error for brain tumor extraction from MRI images," *International Journal of Engineering Trends and Applications*, vol. 4, no. 2, pp. 25-30, 2017.

[33] S. Yuan, B. Yang, and H. Fang, "Direct root-mean-square error for surface accuracy evaluation of large deployable mesh reflectors," in *AIAA SciTech 2020 Forum*, 2020, p. 0935.

[34] S. Jeong, M. Ferguson, R. Hou, J. P. Lynch, H. Sohn, and K. H. Law, "Sensor data reconstruction using bidirectional recurrent neural network with application to bridge monitoring," *Advanced Engineering Informatics*, vol. 42, p. 100991, 2019. https://doi.org/10.1016/j.aei.2019.100991

[35] R. Malhotra, A. Shakya, R. Ranjan, and R. Banshi, "Software defect prediction using Binary Particle Swarm Optimization with Binary Cross Entropy as the fitness function," in *Journal of Physics: Conference Series, IOP Publishing*, 2021, vol. 1767, no. 1, p. 012003, doi: https://doi.org/10.1088/1742-6596/1767/1/012003