LONTAR KOMPUTER VOL. 15, NO. 2 AUGUST 2024 **p-ISSN 2088-1541**
DOI : 10.24843/LKJITI.2024.v15.i02.p01 **e-ISSN 2541-5832**
Accredited Sinta 2 by RISTEKDIKTI Decree No. 158/E/KPT/2021

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Principal Component Analysis (PCA) for Particulate Matter (PM) Anomaly Detection

Hanna Arini Parhusip[a1], Suryasatriya Trihandaru [a2], Bambang Susanto[a3], Johanes Dian Kurniawan [a4], Adrianus Herry Heriadi[b1], Petrus Priyo Santosa[b2], Yohanes Sardjono[c1],

[a]Master of Data Science, Science and Mathematics Faculty, Satya Wacana Christian University
Jl.Diponegoro 52-60, Salatiga, Indonesia
[b]PT Artha Puncak Semesta Indonesia (APSI)
Ruko Cibubur Indah Blok F No. 8, Kel. Cibubur Kec. Ciracas, Jakarta Timur, DKI Jakarta, 13720
[c]PT Badan Tenaga Nuklir Nasional (BATAN)
Jl. Babarsari Kotak Pos 6101 YKBB Yogyakarta 55281

[1]hanna.parhusip@uksw.edu (Corresponding author)
[2]suryasatriya@uksw.edu
[3]bambang.susanto@uksw.edu
[4]632022001@student.uksw.edu
[5]herryhd@bsg-auto.com
[6]p.priyo.santosa@bsg-auto.com
[7]ysardjono@batan.go.id

Abstract

This research addresses a critical issue in industrial environments: air quality, specifically regarding PM 1.0 and PM 2.5. High concentrations of these particles pose significant health risks. The study measures temperature, humidity, pressure, altitude, PM 1.0, and PM 2.5 and shows the effectiveness of using AIOT-Particle devices to analyze these features with Principal Component Analysis (PCA). Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is used to detect anomalies during the observation period. Anomalies occur when the altitude ranges from 65 to 70 units, according to PM 1.0 and PM 2.5 values. The positions where anomalies occur are illustrated based on altitude, temperature, pressure, and concentration. The results demonstrate that altitude dominates as the first feature. Finally, the research concludes that altitude, PM 1.0, and PM 2.5 are the dominant features. The study confirms the effectiveness of PCA and recommends using these three features for anomaly detection in DBSCAN. Overall, the research highlights the novelty and success of AIOT-Particle in industrial environments.

Keywords: *PCA, DBCAN, Anomalies, AIOT-Particle, PM 1.0, PM 2.5*

1. Introduction

Matter particles such as PM 1.0 and PM 2.5 are a significant concern for air quality and the environment [1]. PM 1.0 and PM 2.5 are air particle sizes based on their diameter. PM 1.0 refers to particles with a diameter less than or equal to 1.0 micrometers (μm), while PM 2.5 refers to particles with a diameter less than or equal to 2.5 μm [2]. Those particles come from a variety of sources, such as motor vehicles, industry, fossil fuel combustion, and other human activities. Both have the same potential to affect human health and the environment adversely. However, PM 1.0 particles are smaller than PM 2.5, so they can penetrate deeper into the human respiratory system and attach to lung tissue [3]. This can lead to a variety of health problems, including respiratory diseases and other health problems [4]. Therefore, it is important to control the concentration of PM 1.0 and PM 2.5 particles in the air to keep it at a level that is safe for human health. The World Health Organization recommends daily limits of 25 μg/m³ for PM 1.0 and 10 μg/m³ for PM 2.5 [5].

This can lead to a variety of health problems, including respiratory diseases, allergies, irritation of the eyes, nose, and throat, as well as other health problems [6]. The authors have developed a new device called the AIOT-Particle to monitor these particles. Poor air quality can adversely affect human health and the environment, so it is necessary to monitor and control air quality continuously [7] [8].

The AIOT-Particle can be centrally controlled and monitored through software or applications, allowing users to access data in real-time, receive notifications if pollution levels exceed set limits, and even track changes in air quality over time. AIOT-Particle integration of these sensors in one device can also improve the efficiency of air quality monitoring, especially in the context of long-term monitoring or broad sensor-based monitoring applications. This is a novelty from AIOT-Particle, and the article here shows the results of the functioning of AIOT-Particle. IoT technology enables the real-time and accurate collection of matter particle data [9] and [10]. AIOT - Particle products are designed to connect with various IoT devices and are easy to use [11]. This allows users to monitor air quality in real-time and take necessary actions quickly.

This article employs the Principal Component Algorithm (PCA) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN). We expect that the AIOT-Particle may function well with those methods. The PCA method has been known as the dimensional reduction method in the earthquake grouping information system [12]. DBSCAN clustering concept removes unwanted data from the reduced dataset. The result of the clustering algorithm has been used to classify the various values using machine learning concepts. This proposed concept is used to predict easily the production value in manufacturing organizations [13]. The study result showed that DBSCAN can identify peculiar data points that deviate from the normal data distribution and that anomalous weather is characterized by high humidity and low temperatures [14].

2. Research Methods

AIOT (Artificial Intelligence of Things) - Particle product is designed to utilize artificial intelligence (AI) and Internet of Things (IoT) technologies to monitor, collect, and analyze airborne matter particle data [15]This study aims to provide valuable insights into environmental dynamics, enable effective data-driven decision-making, and contribute to advancing knowledge in environmental monitoring and analysis.

2.1. Materials and Tools

Using AIOT-Particle products is indispensable to improving air quality and the environment. These products can help collect accurate data and provide practical solutions to reduce air pollution. The AIOT-Particle infrastructure is shown in Figure 1.

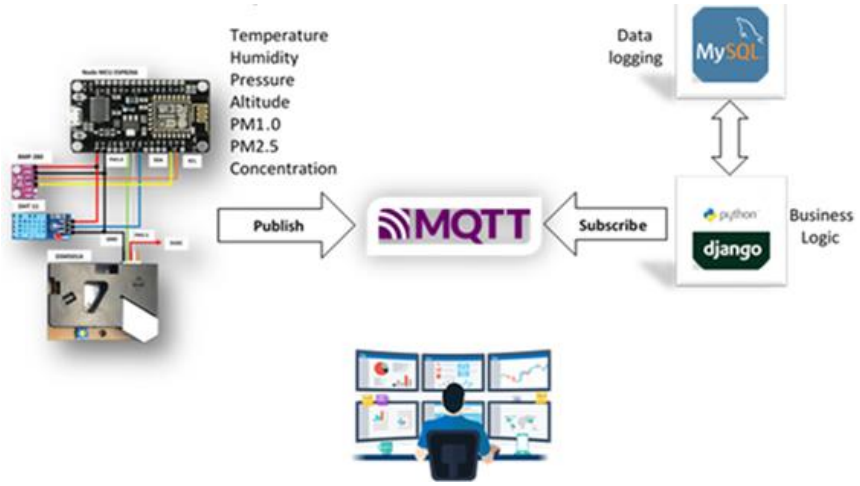

Figure 1. AIOT-Particle Infrastructure

Figure 1 depicts the AIOT-particle infrastructure, which is consisting the following elements:

a. Sensors
Three sensors were used, namely:
1. The DSM501 sensor detects dust particles with a size concentration of > 1 micron as solid matter / particulate molecules (PM) to monitor air quality. DSM501 is also used to detect the particle concentration in particulate matter (PM). This DSM501 sensor uses the laser scattering principle to detect and measure the concentration of fine particles in the air.
2. The BME280 sensor module is designed to read barometric pressure, altitude, temperature, and humidity. Because pressure changes with altitude, the BME280 sensor can also estimate altitude. There are several versions of this sensor module. The BME280 sensor module uses the I2C or SPI communication protocol to exchange data with microcontrollers.
3. The DHT11 sensor module can detect temperature and humidity with a calibrated digital signal output. Exclusive digital signal acquisition techniques and temperature and humidity sensing technology ensure high reliability and long-term stability. This sensor includes an NTC for temperature measurement and a resistive-type humidity measurement component for humidity measurement. It is connected to a high-performance 8-bit microcontroller, offering exceptional quality, fast response, anti-interference capability, and cost-effectiveness. This study, DHT 11 is used to detect temperature, while BME280 is used to detect pressure, altitude, and humidity.

b. ESP8266
The ESP8266 is a low-cost Wi-Fi microchip produced by Expressive Systems. The ESP8266 offers a cost-effective solution for adding Wi-Fi functionality to devices like IoT (Internet of Things) projects, home automation systems, and wireless sensor networks. It features a full TCP/IP protocol stack, making it capable of TCP and UDP communication. In this study, the ESP8266 operates as a standalone microcontroller, running code directly on the chip.

c. My SQL
MySQL is an open-source relational database management system (RDBMS) widely used for managing and storing data. MySQL is one of the applications of the Relational Database Management System (RDBMS). In addition to RDBMS are Firebase, PostgreSQL, MariaDB, and SQLite. MySQL was chosen for this website development because it is open source, allowing every user to develop applications using MySQL, and MySQL is a fast, comprehensive, and reliable RDBMS application. MySQL database queries generate sensor data and admin user databases when developing the website program.

d. MQTT
MQTT (Message Queuing Telemetry Transport) is a standardized message protocol, or set of rules, used for machine-to-machine communication. Intelligent sensors, wearable devices, and Internet of Things (IoT) devices typically need to send and receive data over networks with limited resources and bandwidth. These IoT devices use MQTT for data transmission because it is easy to implement and efficiently communicates IoT data. MQTT supports message delivery between devices to the cloud and from the cloud to devices. In this study, a mosquito broker is used to send data. This mosquito is a free platform and easy to install, making it the choice for this research.

e. Djanggo
Djanggo is a high-level Python web framework that enables the rapid development of secure and scalable web applications. It follows the Model-View-Controller (MVC) architectural pattern, emphasizing reusability and "don't repeat yourself" (DRY) principles. Django provides many built-in features, including an object-relational mapper (ORM) for database management, a robust authentication system, URL routing, a

template engine, and an administrative interface. In this study, Djanggo is a framework used to create web applications with Python and the RESTful API, which is an architectural style for the application program interface (API) that uses HTTP requests to access and manipulate data.

2.2. Methods

2.2.1. DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is one of the popular clustering algorithms used in machine learning and data analysis. DBSCAN can identify clusters in data by extracting areas where there is a high density of data points [16]. DBSCAN works by checking the distance between each data point in the dataset. Each data point is labeled based on whether or not it belongs to a cluster. DBSCAN divides data points into three groups:

a. *Core points* are data points with several neighbors around them that are more significant than or equal to a threshold value.
b. *Border points* are data points with fewer neighbors than the threshold value but around *core points.*
c. *Noise points* are data points that do not belong to any cluster because their neighbors are insufficient to form a cluster.

Using these groups, DBSCAN can extract clusters in the data. This algorithm works by visiting each data point one by one, and if a data point is considered a core point, then DBSCAN will process all the points connected to it to form a cluster. Border points are then added to the cluster connected to the core points, while noise points are ignored. One of the advantages of DBSCAN is its ability to handle datasets with irregular or oddly shaped clusters. In addition, DBSCAN can also identify noise points, making the resulting clustering results more accurate and reliable[17] and [18]. However, DBSCAN also has a weakness: its performance depends on the chosen threshold value. Threshold values that are too small can produce many unnecessary noise points, while threshold values that are too large can produce clusters that are too large. Therefore, choosing the correct threshold value is the key to getting good clustering results from DBSCAN. The DBSCAN algorithm is detailed as follows:

1. Initialize DBSCAN parameters, namely eps and min samples.
2. Calculate the distance between each pair of data points on the dataset.
3. identify neighbors who are no more than eps away from that point for each data point. A point is a core point if the number of neighbors is at least min_samples.
4. Create a new cluster for each core point and add all its neighboring points.
5. For each neighboring point, if it is also a core point, merge that neighboring cluster into the appropriate cluster core point.
6. Repeat step 5 until there are no more changes in the cluster determination.
7. Points that do not belong to any cluster are considered noise points.

Here, *EPS* in step 1 is the maximum distance between two points to be considered neighbors, and min_samples is the minimum number of neighbors for a core point. In the case of DBSCAN, we can use *the Silhouette Coefficient* metric to perform an internal evaluation. This metric measures cluster quality based on how well each data point is within the same cluster with other points that are similar and different from points in other clusters. Silhouette Coefficients range from -1 to 1, with higher values indicating better clusters[19]and [20]. The clustering result is good if the Silhouette Coefficient is close to 1. If the value is close to -1, then clustering is bad. If the value is close to 0, then *clustering* is considered to have no clear structure.

2.2.2. Principal Component Analysis (PCA)

In previous studies, PCA has been used to detect dominant factors[21]and [22] in data on various gas components arising from heavy machinery. Several applications of PCA by other authors are also widely carried out, such as to examine pollution [23] and pay attention to oil and gas production [24] and oil prices. Therefore, the steps in PCA are written in an outline that can be traced in these various libraries. Using PCA, we can reduce the complexity of the data by transforming it into a lower dimensional space while retaining most of the information in the

original data. Here are the structured steps to implement PCA with Python math and code using Google Colab:

a. Step 1: Import Library
In Colab, import the libraries needed for data analysis. For PCA, we will use NumPy for data manipulation, mathematical computation, and scikit-learn to implement PCA algorithms.

b. Step 2: Data Preparation
Prepare the data for PCA. Ensure the data are well prepared and converted into matrix form.

c. Step 3: Standardize Data (Optional)
If the data scale differs between features, it is recommended to standardize each feature with an average of zero and a variance of one. This helps avoid the dominance of large-scale features when performing PCA. We may use the formula for standardization as follows:

$$Z_1 = \frac{(X_1-\mu_1)}{\sqrt{\sigma_{11}}}, Z_2 = \frac{(X_2-\mu_2)}{\sqrt{\sigma_{22}}}, \dots , Z_p = \frac{(X_j-\mu_j)}{\sqrt{\sigma_{jj}}} \quad (1)$$

Where X_j is the initial data for the *j*-th features and σ_{jj} : *jj*-th component of *S.*

d. Step 4: PCA Calculation with NumPy
The next step is to perform PCA calculations manually using NumPy. This includes calculating covariance matrices, values, and eigenvectors, as shown in Section 2.

Let X be the matrix of the data; then, we may have S as the covariance matrix. We can obtain the eigenvectors and the corresponding eigenvalues by solving the following formula, i.e.

$$S\vec{v}_k = \lambda_k\vec{v}_k \text{ yielding to } [S-\lambda_k I]\vec{v}_k = \vec{0}. \quad (2)$$

Based on linear algebra, we have a nontrivial solution $\vec{v}_k$ if and only if det[$S-\lambda_k I$] = 0. Solving this equation means that we have a polynomial of λ_k of order *p,* where *p* is the number of features in our data, and *k*=1...,*p*. Furthermore, the eigenvector $\vec{v}_k$ is obtained by solving $[S-\lambda_k I]\vec{v}_k = \vec{0}$ for each λ_k.

e. Step 5: Choosing Key Components
Select the main components that will be used to reduce the dimension of the data. Choose several components that can retain most of the variance from the data (usually based on a relatively large number of eigenvalues). After we have each λ_k and $\vec{v}_k$, we need to find the new linear combination of the principal component for each *k* using the following formula,i.e.

$$Y_k = V_k^T \mathrm{X} \quad (3)$$

Based on equation (2) for each linear combination for each *k,* we will have

$$Y_1 = V_1^T \mathrm{X}, Y_2 = V_2^T \mathrm{X}, .. Y_p = V_p^T \mathrm{X}$$

$Y_1 = V_1^T \mathrm{X}, Y_2 = V_2^T \mathrm{X}, .. Y_p = V_p^T \mathrm{X}$ are principal components from the covarians matrix, then we will have a correlation coefficient as follows

$$\rho_{Yi,X_k} = \frac{e_{ki}\sqrt{\lambda_i}}{\sqrt{\sigma_{kk}}} \quad (4)$$

where *i,k* = 1,2,…*p* and ρ_{Y_I,X_k} Is correlation coefficient between the components Y_i and the variables X_k.

f. Step 6: Data Transformation
Transform the data into a new dimensional space formed by the main component.

g. Step 7: Using PCA with sci-kit-learn
In addition to using the manual method above, sci-kit-learn provides a PCA library that makes PCA implementation easier.

2.3. The Used Data

The data contain the features listed with the help of Python programs as follows: Index(['No', 'Date Time', 'DeviceID', 'Temperature', 'Humidity', 'Pressure', 'Altitude', 'pm10', 'pm25', 'Concentration'], dtype='object').

The data source used in this research is derived from readings from the AIOT-Particle sensors. The total data measured is 10,970, taken during January 2023 (1 month), and the data range is 4,23 ms. The number of data points greater than 10,000 can represent the data characteristic for each feature in this study. The views for the first five rows of 10,970 rows are shown in Table 1.

Table 1. Data The first five rows of data contents, Jan. 5, 2023, 1:55 a.m.

No	DeviceID	Temperature	Humidity	Pressure	Altitude	PM1.0	PM2.5	Concentration
1	ESP8266	28.5	72	100559	63.96	36	2	0.07
2	ESP8266	28.5	72	100560	63.88	36	2	0.07
3	ESP8266	28.5	72	100556	64.21	0	0	0.00
4	ESP8266	28.5	72	100557	64.13	0	0	0.00
5	ESP8266	28.5	72	100557	64.13	0	0	0.00

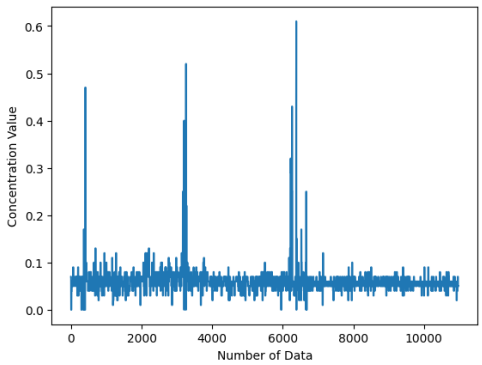

Figure 2. Air concentration during observation

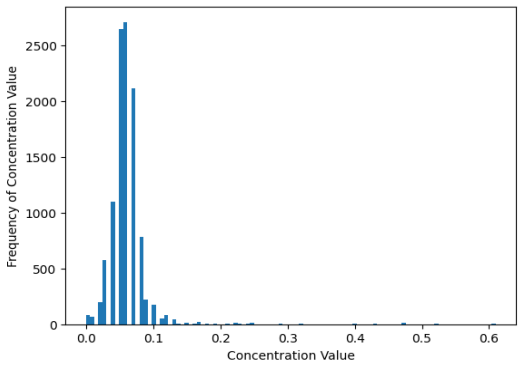

Figure 3. Visualization of the Concentration value vs. the Frequency of Concentration

We create a histogram from the data stored in variable X, using 100 bins for data division. A histogram is a plot that shows the frequency distribution of numerical data in the form of bars. Here, numeric data is in the "Concentration" column of the DataFrame dt. On a histogram, the horizontal axis (x-axis) shows a range of data values separated into several equal-width intervals, referred to as "bins". Meanwhile, the vertical axis (y-axis) shows the frequency of the data that falls in each interval. Figure 3 shows that the maximum concentration measured is 0.61. In this context, "PM 1.0" refers to the size of airborne particles less than or equal to 1.0 micrometers (μm) in micrograms per cubic meter (μg/m³). Figure 4 shows that PM 1.0 is 552, indicating that the concentration of particles smaller than or equal to 1.0 μm in air is 552 μg/m³. Concentrations of particles like these can come from various sources, such as vehicular, industrial, and other human activity pollution. High levels of PM 1.0 can have a devastating impact on human health and the environment [25]. Data processing is the first step, which removes features that are not numeric, so it will need to take a look at the features of any data by displaying them. Suppose a concentration of about 0.05 obtained the most, which is several more than 2500. We visualize concentration vs. PM 1.0 in Figure 4.

p-ISSN 2088-1541
e-ISSN 2541-5832

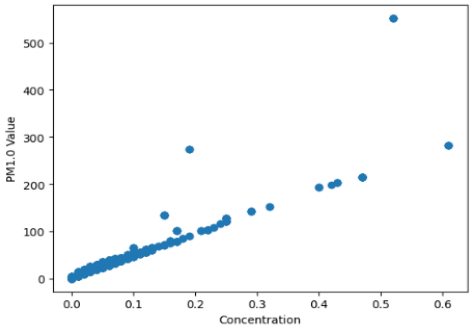

Figure 4. Concentration vs PM 1.0 for January 2023 in the observed area

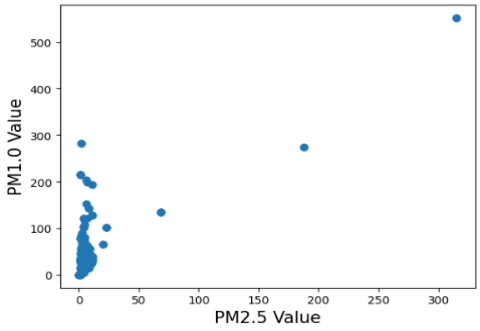

Figure 5. PM 2.5 value vs. PM 1.0 value

In this section, the results of using the DBSCAN and PCA methods will be shown in the acquired data obtained in January 2023 in the office area of Jakarta.

3. Result and Discussion

The testing process involves placing the AIOT-Particle indoors and outdoors for 7x24 hours around an office area near the industrial environment during January 2023 to obtain variations in temperature, humidity, air pressure, altitude, PM1.0, and PM2.5 data to be analyzed.

3.1. Results on DBSCAN

In this section, the altitude, PM 1.0, and PM 2.5 features were chosen as features that affect our anomaly detection. Furthermore, the DBSCAN algorithm was used to set eps=3 and min_samples =5. Additionally, DBCAN is performed by standardizing the given data. Finally, the results are discussed below.

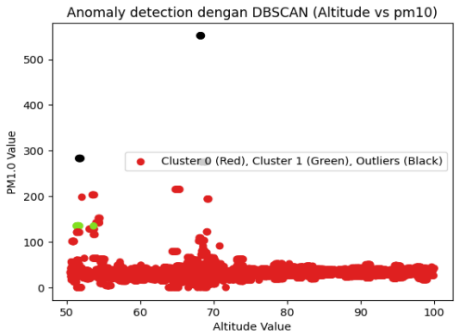

Figure 6. Visualization of Altitude vs. PM 1.0

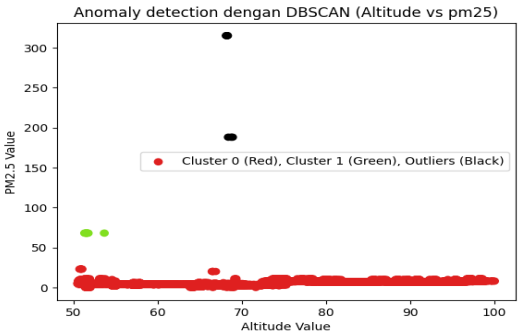

Figure 7. Visualization of altitude vs. PM 2.5

Figure 6 measures the PM 1.0 values by considering the altitude as an independent feature. In almost all altitude values, the PM 1.0 value is about 100. However, in the interval of altitude 50-55, some points are greater than 100. Additionally, some points in the altitude range of 65 - 70 are indicated as anomalies occurring. Similarly, we can pay attention to the relationship between altitudes and PM 2.5, depicted in Figure 7. It is observed that anomalies occur at altitudes of 50 - 55 and about 65 - 70. Next, the positions where the anomalies occur with PM 1.0 can be found, illustrated in Figure 8. Figures 9-13 display the positions where the anomalies occur for each feature.

p-ISSN 2088-1541
e-ISSN 2541-5832

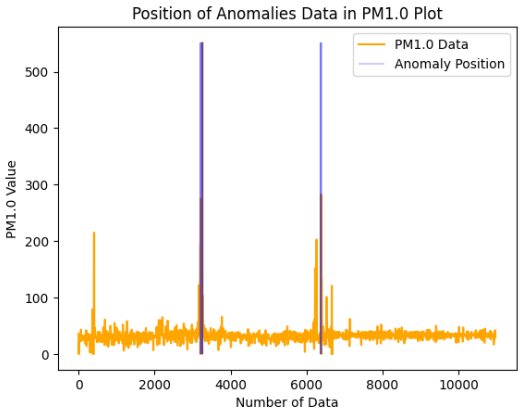

Figure 8. Illustration of the positions of anomalies based on the values of PM 1.0

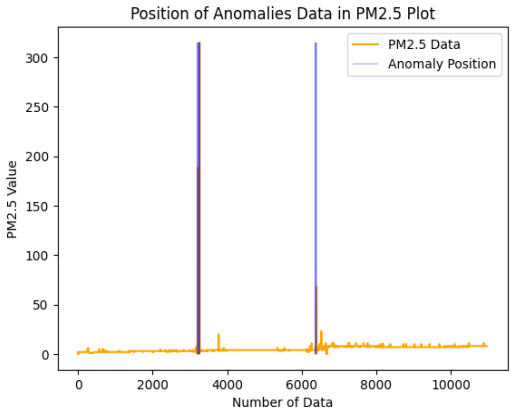

Figure 9. Illustration of the positions of anomalies based on the values of PM 2.5

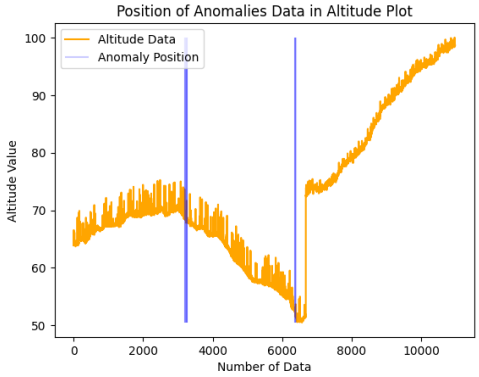

Figure 10. Illustration of the positions where anomalies of altitudes occur.

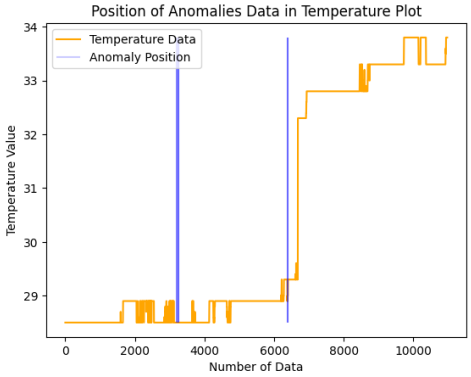

Figure 11. Illustration of the positions where anomalies of temperatures occur.

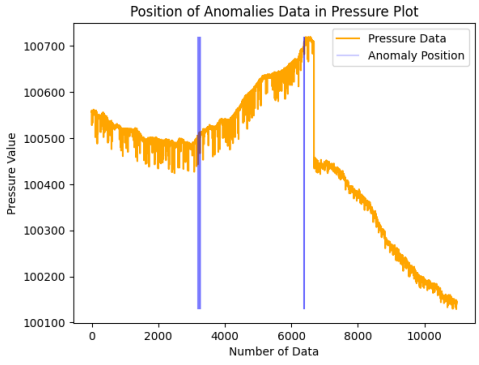

Figure 12. Illustration of the positions where anomalies of pressure occur.

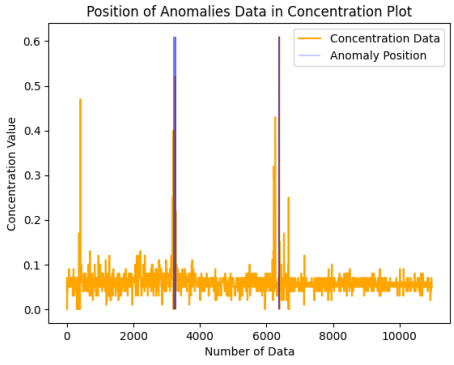

Figure 13. Illustrate the positions (denoted by the blue) where Concentration anomalies occur.

The anomalies from each feature arse shown in Figures 7-13 since the AIOT Particle placement moves from indoors to outdoor areas to test the sensor performance. The DSM501 then detects the movement of changes in PM1.0 and PM2.5 particles caused by residual combustion activities in the industrial environment around the office area. Meanwhile, the DHT11 and BME280 sensors also detect changes in temperature, humidity, pressure, and altitude caused by the transition from

indoor to outdoor temperatures, as the sensor is relocated indoors to outdoors to ensure its functionality indoors and outdoors.

3.2. Results on PCA

PCA is done by standardizing data initially. Following the procedures in Section 2 and implementing Eq. 2, we get the composition of the covariance matrix as follows:

[[1.00009116 -0.9871017 -0.84746689 0.84753893 0.01821581 0.25593196 -0.11175798]
[-0.9871017 1.00009116 0.89668966 -0.89676556 -0.01668328 -0.24820832 0.10990746]
[-0.84746689 0.89668966 1.00009116 -1.00009093 -0.00686686 -0.18321012 0.08928428]
[0.84753893 -0.89676556 -1.00009093 1.00009116 0.00688227 0.18324668 -0.0892853]
[0.01821581 -0.01668328 -0.00686686 0.00688227 1.00009116 0.63595537 0.91176615]
[0.25593196 -0.24820832 -0.18321012 0.18324668 0.63595537 1.00009116 0.27096529]
[-0.11175798 0.10990746 0.08928428 -0.0892853 0.91176615 0.27096529 1.00009116]]

We can calculate that the values of the 1st to 7th eigenvalues are shown in the following list:

$$\lambda_1 = 5.45154683\text{e} - 01; \lambda_2 = 3.20463374\text{e} - 01; \lambda_3 = 9.81211951\text{e} - 02;$$
$$\lambda_4 = 3.47230219\text{e - } 02; \lambda_5 = 1.09573618\text{e-}03; \lambda_6 = 4.41960716\text{e-}04;$$
$$\lambda_7 = 2.89781995\text{e-}08.$$

Based on these results, we get that the average eigenvalue is 0.14285714285714282. Eigenvalues represent the amount of variance along the corresponding eigenvector direction.

For instance, the eigenvalue of PC1 (λ_1) is approximately 0.545, indicating that the direction represented by the corresponding eigenvector captures a substantial amount of variance. With the help of a Python program, we can get the corresponding eigenvectors by implementing Eq. (2), i.e.

$\vec{v}_1 =$ [4.87765405e-01 -5.00165879e-01 -4.92492641e-01 4.92513403e-01
2.85164475e-02 1.54502221e-01 -4.60046181e-02]
$\vec{v}_2 =$ [-2.28666954e-02 2.60278410e-02 3.72737508e-02 -3.72637646e-02
6.64325406e-01 4.48486076e-01 5.94603851e-01]]
$\vec{v}_3 =$ [1.96882714e-02 1.32201814e-02 1.52886055e-01 -1.52843394e-01
-8.68924453e-02 8.12109038e-01 -5.34444538e-01]
$\vec{v}_4 =$[-6.00276217e-01 4.01585354e-01 -4.76539988e-01 4.76257122e-01
-2.59792136e-03 1.34682176e-01 -7.96267678e-02]
$\vec{v}_5 =$ [-6.33120577e-01 -7.66618898e-01 7.64814297e-02 -7.48447180e-02
-1.19749429e-03 -5.65449226e-05 1.10527228e-03]
$\vec{v}_6 =$ [9.09670650e-04 -2.36425562e-03 2.05710368e-03 -1.77555862e-03
7.41822124e-01 -3.11975831e-01 -5.93596845e-01]
$\vec{v}_7 =$ [6.04890268e-04 9.74786344e-04 7.06925398e-01 7.07287160e-01
-1.49306027e-04 5.91797171e-05 1.13567777e-04]]

Eigenvectors indicate the directions in the original feature space along which the data varies the most. Each eigenvector is associated with a principal component, and its components represent the weights of the original features in that principal component. We describe each variance of the principal component. It is obtained that there are seven components, and the value of each variance is shown in Table 2.

Table 2. The value of variance for each principal component

PC1	PC2	PC3	PC4	PC5	PC6	PC7
3.816431	2.243448	0.686911	0.2430833	0.007670852	0.003094007	2.028659e-07

Based on Table 2, each PC represents a direction in the feature space along which the data varies the most. Given the variance values for each principal component (PC1 - PC7), as shown in Table 2, the first principal component, denoted by PC1, captures the direction in the data that explains the most variance. In this case, PC1 has a variance of 3.816431, indicating that it explains the highest variance compared to the other components. Finally, PC7 captures the remaining variance not captured by the previous components. PC7 has a variance of 2.028659e-07, which is very small compared to the variances of the other components, indicating that it explains very little variance in the data.

We also observe some facts using Eq.(4). The percentage variance of the 1st principal component is 54.51546829592294 %. Similarly, the percentage variances for 2nd, 3rd, 4th, 5th, 6th , and 7th are 32.04633741998789%, 9.812119509350422 %, 3.4723021871505853 %, 0.10957361811992745 %, 0.04419607164828615 %, and 2.897819952989454e-06 % respectively.

Add up all the percent represented by each PC to get a total percent of 100.00000000000000001, which shows that we have reached the 7th principal component correctly. We know from PCA theory that the percentage of variance is obtained from the eigenvalue x 100%. The percentages decrease for PC3 through PC7, reflecting their decreasing contribution to explaining the variance in the dataset. The pattern of each variance for each component of the principle and the magnitude of the variance (vertical) are depicted in Figure 14.

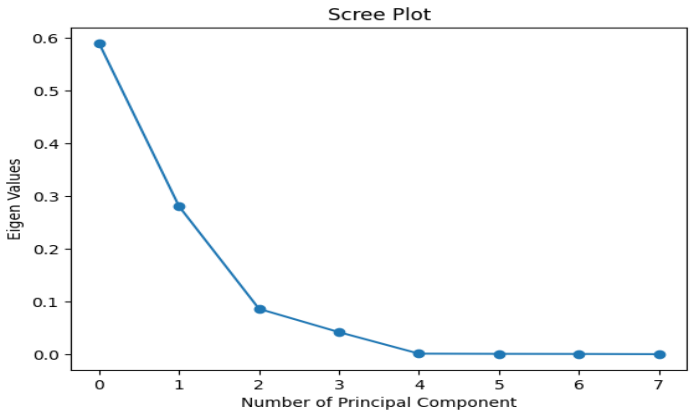

Figure 14. Scree Plot of the Variance of Each Principal Component

We can determine which feature is the most dominant by obtaining the eigenvalue. Based on the results of the Principal Component Analysis (PCA), the features that have the most influence on detecting anomalies are altitude, PM1.0 (particulate matter with a diameter of 1.0 micrometers or less), and PM2.5 (particulate matter with a diameter of 2.5 micrometers or less). Altitude, PM1.0, and PM2.5 play significant roles in capturing the variability within the dataset, as reflected by their contributions to the principal components with the first three highest variances. These results show that in DBSCAN, these three features were chosen when anomaly detection was performed. The use of PCA has proven our assumptions.

4. Conclusion

In the article, the monitoring and data collection tool called AIOT-Particle is shown. The main part of AIOT-Particle is sensors to measure temperature, humidity, pressure, and particles measured by PM 1.0 and PM 2.5. The data were obtained by placing the AIOT-Particle indoors and outdoors around an office in January 2023, which is near the industrial environment. Furthermore, the received data are analyzed using DBSCAN and PCA. The DBSCAN method is used to evaluate the presence or absence of anomalies within a month, characterized by several data points that are significantly different from most data points. At 50-55 and 65-70 altitude intervals, there are anomalies for PM 1.0 and PM 2.5, and the anomalies occur in the same intervals at the altitude feature. The anomalies are determined by the determination of anomalies generated by the Python code, which is based on the density of points within a specific radius and the minimum number of points required to form a dense region. Points that do not meet these criteria are labeled as anomalies. Furthermore, the positions where the anomalies occur are visualized. Similarly, pressure, temperature, and concentration show where anomalies occur in each feature. Additionally, PCA is used to find features that dominate the overall data. By calculating the variance of each feature, altitude, PM 1.0, and PM 2.5 are the three dominant features. Finally, the device in this research, the AIOT-Particle, has been functioning properly.

Acknowledgment

Satya Wacana Christian University supports the paper under the internal grant research 2023 No. 096/SPK-PW/RIK/8/2023.

References

[1] S. Algarni, R. A. Khan, N. A. Khan, and N. M. Mubarak, "Particulate matter concentration and health risk assessment for a residential building during COVID-19 pandemic in Abha, Saudi Arabia," *Environmental Science and Pollution Research International*, vol. 28, no. 46, pp. 65822–65831, 2021, doi: 10.1007/s11356-021-15534-6.

[2] T. Xayasouk, H. M. Lee, and G. Lee, "Air pollution prediction using long short-term memory (LSTM) and deep autoencoder (DAE) models," *Sustainability*, vol. 12, no. 6, 2020, doi: 10.3390/su12062570.

[3] K. Prem *et al.*, "The effect of control strategies that reduce social mixing on outcomes of the COVID-19 epidemic in Wuhan, China," *The Lancet Public Health*, vol. 5, no. 5, p. 2020.03.09.20033050, 2020, doi: 10.1101/2020.03.09.20033050.

[4] I. Manisalidis, E. Stavropoulou, A. Stavropoulos, and E. Bezirtzoglou, "Environmental and Health Impacts of Air Pollution: A Review," *Front Public Health*, vol. 8, no. 14, pp. 1–13, 2020, doi: 10.3389/fpubh.2020.00014.

[5] Geneva: World Health Organization, "WHO global air quality guidelines," *WHO global air quality guidelines Particulate matter (PM2.5 and PM10), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide*. WHO European Centre for Environment and Health, Bonn, pp. 1–360, 2021.

[6] D. E. Schraufnagel, "The Health Effects of Ultrafine Particles," *Experimental and Molecular Medicine*, vol. 52, no. 3, pp. 311–317, 2020, doi: 10.1038/s12276-020-0403-3.

[7] G. B. Fioccola, R. Sommese, I. Tufano, R. Canonico, and G. Ventre, "Polluino: An efficient cloud-based management of IoT devices for air quality monitoring," in *2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a Better Tomorrow, RTSI 2016*, 2016, pp. 1–7. doi: 10.1109/RTSI.2016.7740617.

[8] V. Mohammadi, A. M. Rahmani, A. M. Darwesh, and A. Sahafi, "Trust-based Recommendation Systems in Internet of Things: a Systematic Literature Review," *Human-centric Computing and Information Sciences*, vol. 9, no. 1, 2019, doi: 10.1186/s13673-019-0183-8.

[9] M. Noura, "Interoperability in Internet of Things : Taxonomies and Open Challenges," Mobile Networks and Applications, Vol. 24, pp. 796–809, 2019, doi: 10.1007/s11036-018-1089-9.

[10] J. Jo, B. Jo, J. Kim, S. Kim, and W. Han, "Development of an IoT-Based indoor air quality monitoring platform," *Journal of Sensors*, vol. 2020, pp. 13–15, 2020, doi: 10.1155/2020/8749764.

[11] F. Durán, A. Krishna, M. Le Pallec, R. Mateescu, and G. Salaün, "Models and analysis for user-driven reconfiguration of rule-based IoT applications," *Internet of Things (Netherlands)*, vol. 19, no. August, pp. 1–10, 2022, doi: 10.1016/j.iot.2022.100515.

[12] Mustakim, E. Rahmi, M. R. Mundzir, S. T. Rizaldi, Okfalisa, and I. Maita, "Comparison of DBSCAN and PCA-DBSCAN Algorithm for Grouping Earthquake Area," *2 2021 International Congress of Advanced Technology and Engineering (ICOTEN)*, pp. 0–4, 2021, doi: 10.1109/ICOTEN52080.2021.9493497.

[13] S. Umadevi and N. S. Rajini, "Dimensionality reduction of production data using PCA and DBSCAN techniques," *Advances in Parallel Computig*, vol. 37, no. 1, pp. 458–462, 2020, doi: 10.3233/APC200184.

[14] S. Wibisono, M. T. Anwar, A. Supriyanto, and I. H. A. Amin, "Multivariate weather anomaly detection using DBSCAN clustering algorithm," *Journal of Physics: Conference Series*, vol. 1869, no. 1, 2021, doi: 10.1088/1742-6596/1869/1/012077.

[15] T. W. Sung, P. W. Tsai, T. Gaber, and C. Y. Lee, "Artificial Intelligence of Things (AIoT) Technologies and Applications," *Wireless Communications and Mobile Computing*, vol. 2021, 2021, doi: 10.1155/2021/9781271.

[16] H. Belyadi and A. Haghighat, "Chapter 4 - Unsupervised machine learning : clustering algorithms," 2021, pp. 1–3. [Online]. Available: https://doi.org/10.1016/B978-0-12-821929-4.00002-0

[17] A. Fahim, "A Varied Density-based Clustering Algorithm," *Journal of Computational Science*, vol. 66, p. 101925, 2023, doi: https://doi.org/10.1016/j.jocs.2022.101925.

[18] F. Huang *et al.*, "Research on the parallelization of the DBSCAN clustering algorithm for spatial data mining based on the Spark platform," *Remote Sensing*, vol. 9, no. 12, 2017, doi: 10.3390/rs9121301.

[19] M. Monshizadeh, V. Khatri, R. Kantola, and Z. Yan, "A deep density based and self-determining clustering approach to label unknown traffic," *Journal of Network Computer Applications*, vol. 207, no. July, p. 103513, 2022, doi: 10.1016/j.jnca.2022.103513.

[20] G. Erda, C. Gunawan, and Z. Erda, "Grouping of Poverty in Indonesia Using K-Means With Silhouette Coefficient," *Parameter: Journal of Statistics*, vol. 3, no. 1, pp. 1–6, 2023, doi: 10.22487/27765660.2023.v3.i1.16435.

[21] G. R. Igtisamova, N. N. Soloviev, F. A. Ikhsanova, D. S. Nosirov, and A. A. Abdulmanov, "Principal component analysis for assessing oil and gas production (the case of the Kogalym field)," in *IOP Conference Series: Earth and Environmental Science*, IOP, 2019. doi: 10.1088/1755-1315/378/1/012113.

[22] M. Greenacre, P. J. F. Groenen, T. Hastie, A. I. D'Enza, A. Markos, and E. Tuzhilina, "Principal component analysis," *Nature Reviews Methods Primers*, vol. 2, no. 1, 2022, doi: 10.1038/s43586-022-00184-w.

[23] I. Gergen and M. Harmanescu, "Application of principal component analysis in the pollution assessment with heavy metals of vegetable food chain in the old mining areas," *Chemistry Central Journal*, vol. 6, no. 1, pp. 1–13, 2012, doi: 10.1186/1752-153X-6-156.

[24] M. He, Y. Zhang, D. Wen, and Y. Wang, "Forecasting crude oil prices: A scaled PCA approach," *Energy Economics*, vol. 97, no. May, pp. 4–7, 2021, doi: 10.1016/j.eneco.2021.105189.

[25] L. Levei *et al.*, "Temporal trend of PM10 and associated human health risk over the past decade in Cluj-Napoca City, Romania," *Applied Sciences*, vol. 10, no. 15, pp. 1–13, 2020, doi: 10.3390/APP10155331.