VISUALIZING AND INTERPRETING CLUSTERING RESULTS IOT WEATHER DATA

S. Harshitha,
UG Student,
Department of CSE,
St. Martin's Engineering college,
Secunderabad, Telangana, India
sangaharshitha2003@gmail.com

Dr. M. Vadukarassi,
Associate Professor,
Department of CSE,
St. Martin's Engineering College,
Secunderabad, Telangana, India
drmvadivukarassicse@smec.ac.in

Abstract- The Internet of Things (IoT) has revolutionized the way weather data is collected, providing vast amounts of real-time information from various sensors and weather stations. However, processing and making sense of this massive volume of data can be challenging. One popular technique to analyze such data is clustering, which groups similar data points together based on their features. Clustering helps identify patterns, relationships, and insights in IoT weather data, aiding in understanding weather patterns and making informed decisions in various domains like agriculture, transportation, and disaster management. Before the adoption of IoT and modern data visualization tools, weather data analysis primarily relied on manual interpretation and basic statistical techniques. Analysts would sift through tables of numerical data, making it difficult to discern underlying patterns and relationships. Basic visualizations like line charts and bar graphs provided limited insights and often failed to reveal complex structures within the data. Additionally, older clustering algorithms were computationally expensive and had limitations in handling large-scale IoT weather datasets efficiently. Therefore, the proposed work aims to address the challenges of visualizing and interpreting clustering results in IoT weather data using unsupervised learning algorithm. The study explores the application of cutting-edge clustering algorithms to identify hidden structures and patterns within large-scale weather datasets collected through IoT sensors and stations. By leveraging advanced data visualization techniques, this project will develop interactive and intuitive visual representations of clustering results. The outcome of this project will enable domain experts, meteorologists, and decision-makers to gain valuable insights into IoT weather data, leading to better-informed decisions in various sectors. The intuitive visualizations will empower users to understand complex patterns, trends, and anomalies in the data, fostering advancements in weather forecasting, disaster preparedness, and climate studies.

Keywords: IOT (Internet of things), Clustering, K-means algorithm, Weather data, Visualization, Feature Scaling, Real Time Analytics, Unsupervised Machine Learning, Efficiency.

I. Introduction

This project tackles the analysis of IoT weather data using the K-means clustering algorithm. It encompasses several critical stages, including data preprocessing, feature selection, standardization, K-means clustering, and visualization of clustering results via parallel coordinate plots. The primary objective is to identify inherent patterns or clusters within the weather dataset and gain insights into the interrelationships among various weather condition. The central problem addressed by this code is the segmentation of IoT weather data into discernible clusters based on chosen weather features. Specifically, it endeavours to answer fundamental questions such as whether there exist distinct weather patterns or clusters within the dataset and how various weather parameters, such as humidity, temperature, wind direction, etc., are interconnected within these clusters. Here we introduce a smart weather reporting system over the Internet. Our introduced system allows for weather parameter reporting over the Internet. It allows the people to directly check the weather states online without the need of a weather forecasting agency. System uses temperature, humidity as well as rain with humidity sensor to monitor weather and provide live reporting of the weather statistics. The system constantly monitors temperature using temperature sensor, humidity using humidity sensor and for rain. Weather monitoring system deals with detecting and gathering various weather parameters at different locations which can be analysed or used for weather forecasting. The aim of this system is achieved by technologies such as Internet of Things (IOT) and Cloud. The idea of internet of things is to connect a device to the internet and to other required connected devices. Using Internet, the information from the IOT device can easily be transferred to the cloud and then from the cloud to the end user. Weather Monitoring is an essential practical implementation of the concept of Internet of Things, it involves sensing and recording various weather parameters and using them for alerts, sending notifications, adjusting appliances accordingly and for long term analysis. Also, we will try to identify and display trends in parameters using graphical representation. The devices used for this purpose are used to collect, organize and display information. It is expected that the internet of things is going to transform the world by monitoring and controlling the phenomenon of environment by using sensors/devices which can capture, process, and transmit weather parameters. Cloud is availability of computer system resources like data storage, computing power without direct active management of user. The data captured is transmitted to the cloud so that the data could be further displayed**.**

II. Related work

Through the meteorological system, we can collect data on humidity and Temperature, as well as data on pollution and, considering current and previous data, we can graphically modify the results in any system. After reviewing many articles, there are currently far fewer articles that mention monitoring the combination of temperature, lighting and humidity in a small integrated system and have actuators to change these settings. There is a research paper that discussed the monitoring of these three environmental conditions; however, there was no mention of having actuators to modify.

Thus, the main idea was to create a system that could detect the main components that make up the climate and be able to predict time without human error. Existing weather forecasting methods were generally based on observed patterns of events, and can be called pattern recognition. For example, one could observe that if the sunset was red and normal, the next day often brought a very nice weather. This experience gathers more than and generations to produce the tradition of the time. However, not all these predictions are reliable and since then many of them have not been able to withstand rigorous statistical testing. The simplest way to predict time, persistence, depends on today's conditions to predict tomorrow's conditions. This can be a good way to predict weather when it is in a stationary state, such as during the summer in the tropics. system provides a real-time effect. The main objective of this model to detect the condition mapping, and weather forecasting most necessarily, it warns the people of its devastating effects. This would be very much helpful for landing the aircraft, cloud burst, healthcare effect, tornado, tsunami, navigational, and the shipborne effect. There is a significant role of temperature, humidity, pressure in the system. also have a different area used in the system such as agriculture, logistics, etc. The thing might be anything as the sensor, electronic gadget, and automotive electronic equipment as well. The whole system deals with the monitoring and controlling the environmental condition like temperature, smoke, gas, wind, pressure relative humidity level, and many other gases with sensor transfer the data or information to the cloud platform and store the data The most important features of IoT include artificial intelligence, connectivity, sensors, active engagement, and small device use. A brief review of these features is given below:

AI − IoT essentially makes virtually anything "smart," meaning it enhances every aspect of life with the power of data collection, artificial intelligence algorithms, and networks. This scan means something as simple as enhancing your refrigerator and cabinets to detect when milk and your favorite cereal run low, and to then place an order with your preferred grocer.

Connectivity − New enabling technologies for networking and specifically IoT networking, mean networks are no longer exclusively tied to major providers. Networks can exist on a much smaller and cheaper scale while still being practical. IoT creates these small networks between its system devices.

Sensors − IoT loses its distinction without sensors. They act as defining instruments that transform IoT from a standard passive network of devices into an active system capable of real-world integration.

Active Engagement − Much of today's interaction with connected technology happens through passive engagement. IoT introduces a new paradigm for active content, product, or service engagement.

Small Devices − Devices, as predicted, have become smaller, cheaper, and more powerful over time. IoT exploits purpose-built small devices to deliver its precision, scalability, and versatility.

The advantages of IoT span across every area of lifestyle and business. Here is a list of some of the advantages that IoT has to offer.

Improved Customer Engagement − Current analytics suffer from blind spots and significant flaws in accuracy; and as noted, engagement remains passive. IoT completely transforms this to achieve richer and more effective engagement with audiences.

Technology Optimization − The same technologies and data which improve the customer experience also improve device use, and aid in more potent improvements to technology. IoT unlocks a world of critical functional and field data.

Reduced Waste − IoT makes areas of improvement clear. Current analytics give us superficial insight, but IoT provides real-world information leading to more effective management of resources.

Enhanced Data Collection − Modern data collection suffers from its limitations and its design for passive use. IoT breaks it out of those spaces and places it exactly where humans really want to go to analyse our world. It allows an accurate picture of everything.

Though IoT delivers an impressive set of benefits, it also presents a significant set of challenges. Here is a list of some its major issues.

Security − IoT creates an ecosystem of constantly connected devices communicating over networks. The system offers little control despite any security measures. This leaves users exposed to various kinds of attackers.

Privacy − The sophistication of IoT provides substantial personal data in extreme detail without the user's active participation.

Complexity − Some find IoT systems complicated in terms of design, deployment, and maintenance given their use of multiple technologies and a large set of new enabling technologies.

Flexibility − Many are concerned about the flexibility of an IoT system to integrate easily with another. They worry about finding themselves with several conflicting or locked systems.

Compliance − IoT, like any other technology in the realm of business, must comply with regulations. Its complexity makes the issue of compliance seem incredibly challenging when many consider standard software compliance a battle

IoT software addresses its key areas of networking and action through platforms, embedded systems, partner systems, and middleware. These individual and master applications are responsible for data collection, device integration, real-time analytics, and application and process extension within the IoT network. They exploit integration with critical business systems 9 (e.g., ordering systems, robotics, scheduling, and more) in the execution of related tasks.

Data Collection − This software manages sensing, measurements, light data filtering, light data security, and aggregation of data. It uses certain protocols to aid sensors in connecting with real-time, machine to-machine networks. Then it collects data from multiple devices and distributes it in accordance with settings. It also works in reverse by distributing data over devices. The system eventually transmits all collected data to a central server.Device Integration Software supporting integration binds (dependent relationships) all system devices to create the body of the IoT system. It ensures the necessary cooperation and stable networking between devices. These applications are the defining software technology of the IoT network because without them, it is not an IoT system. They manage the various applications, protocols, and limitations of each device to allow communication.

Real-Time Analytics : These applications take data or input from various devices and convert it into viable actions or clear patterns for human analysis. They analyse information based on various settings and designs in order to perform automation-related tasks or provide the data required by industry.

Application and Process Extension : These applications extend the reach of existing systems and software to allow a wider, more effective system. They integrate predefined devices for specific purposes such as allowing certain mobile devices or engineering instruments access. It supports improved productivity and more accurate data collection. IoT primarily exploits standard protocols and networking technologies. However, the major enabling technologies and protocols of IoT are RFID, NFC, low-energy Bluetooth, low-energy wireless, low-energy radio protocols, LTE-A, and WIFI-Direct. These technologies support the specific networking functionality needed in an IoT system in contrast to a standard uniform network of common systems. It ensures the necessary cooperation and stable networking between devices.

III. Existing Work

Clustering is an unsupervised machine learning technique that divides the given data into different clusters based on their distances (similarity) from each other. The unsupervised k-means clustering algorithm gives the values of any point lying in some cluster to be either as 0 or 1 i.e., either true or false. But the fuzzy logic gives the fuzzy values of any data point to be lying in either of the clusters. Here, in fuzzy c-means clustering, we find out the centroid of the data points and then calculate the distance of each data point from the given centroids until the clusters formed become constant.

Disadvantages of Fuzzy Clustering:

Complexity: Fuzzy clustering algorithms can be computationally more expensive than traditional clustering algorithms, as they require optimization over multiple membership degrees.

Model selection: Choosing the right number of clusters and membership functions can be challenging and may require expert knowledge or trial and error.

IV. Proposed work

This project insights into the inherent structure of IoT weather data, identifying distinct weather patterns or clusters. This information can be valuable for meteorological analysis, forecasting, and various applications where understanding the relationships between weather variables is crucial. The visualization step aids in interpreting the clusters and their significance in real-world meteorological contexts.

1. Data Preparation: This is the initial phase where the IoT weather data is read from the 'minute_weather.csv' file.

Data preparation involves several steps such as sampling every 10th row, filtering out rows with zero rain accumulation and duration, and removing rows with missing values.

The goal of this phase is to ensure that the data is clean and suitable for clustering analysis, eliminating noise and outliers.

2.Feature Selection and Standardization: After data preparation, the selected features for clustering are 'air_pressure,''air_temp,''avg_wind_direction,''avg_wind_speed,''max_wind_direction,''max_wind_speed,'and'relative_humidity.'

These features are standardized to have a consistent scale, preventing certain features from dominating the clustering process.

Feature selection and standardization prepare the dataset for meaningful cluster analysis.

3. K-means Clustering: The standardized dataset is then subjected to the K-means clustering algorithm with a specified number of clusters, in this case, k=12.

K-means aims to group data points into clusters based on their similarity in the selected features.

The result of this phase is a set of cluster centers that represent typical feature values within each cluster.

4. Visualization: Finally, the results of the clustering analysis are visualized in this phase. Parallel coordinate plots are created to depict the clusters and their characteristics. Different plots are generated to showcase clusters with specific attributes, such as low humidity, high temperature, or a combination of high humidity and low temperature.

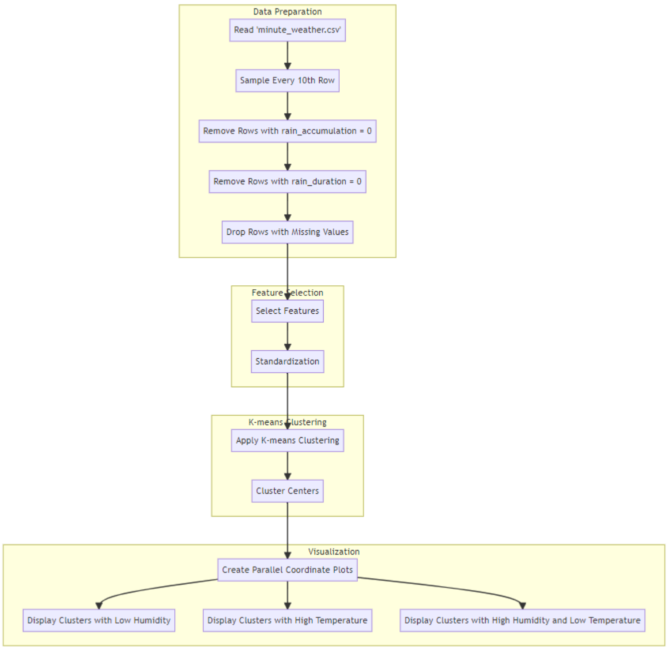

Figure 3.1: Overall design of proposed weapon detection model.

5. Data Preprocessing: Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this, we use data pre-processing task. A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

- Getting the dataset
- Importing libraries
- Importing datasets
- Finding Missing Data
- Encoding Categorical Data
- Splitting dataset into training and test set

Importing Libraries: To perform data preprocessing using Python, we need to import some predefined Python libraries. These libraries are used to perform some specific jobs. There are three specific libraries that we will use for data preprocessing, which are:

NumPy: NumPy Python library is used for including any type of mathematical operation in the code. It is the fundamental package for scientific calculation in Python. It also supports to add large, multidimensional arrays and matrices. So, in Python, we can import it as:
“import NumPy as nm”
Here we have used nm, which is a short name for NumPy, and it will be used in the whole program.
Matplotlib: The second library is matplotlib, which is a Python 2D plotting library, and with this library, we need to import a sub-library pilot. This library is used to plot any type of charts in Python for the code. It will be imported as below:
import matplotlib.pyplot as mpt.
Here we have used mpt as a short name for this library.
Pandas: The last library is the Pandas library, which is one of the most famous Python libraries and used for importing and managing the datasets. It is an open-source data manipulation and analysis library. Here, we have used pd as a short name for this library.
Handling Missing data: The next step of data preprocessing is to handle missing data in the datasets. If our dataset contains some missing data, then it may create a huge problem for our machine learning model. Hence it is necessary to handle missing values present in the dataset. There are mainly two ways to handle missing data, which are:
By deleting the particular row: The first way is used to commonly deal with null values. In this way, we just delete the specific row or column which consists of null values. But this way is not so efficient and removing data may lead to loss of information which will not give the accurate output.
By calculating the mean: In this way, we will calculate the mean of that column or row which contains any missing value and will put it on the place of missing value. This strategy is useful for the features which have numeric data such as age, salary, year, etc.
Encoding Categorical data: Categorical data is data which has some categories such as, in our dataset; there are two categorical variables, Country, and Purchased. Since machine learning model completely works on mathematics and numbers, but if our dataset would have a categorical variable, then it may create trouble while building the model. So, it is necessary to encode these categorical variables into numbers.
Feature Scaling: Feature scaling is the final step of data preprocessing in machine learning. It is a technique to standardize the independent variables of the dataset in a specific range. In feature scaling, we put our variables in the same range and in the same scale so that no variable dominates the other variable. A machine learning model is based on Euclidean distance, and if we do not scale the variable, then it will cause some issue in our machine learning model. Euclidean distance is given as If we compute any two values from age and salary, then salary values will dominate the age values, and it will produce an incorrect result. So, to remove this issue, we need to perform feature scaling for machine learning. Suppose if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models. If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. Thus, it

will create difficulties for our model.

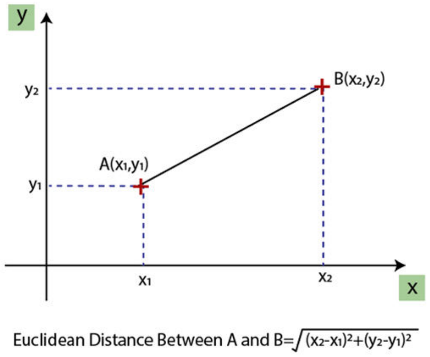

Figure 3.2: Feature scaling.

Splitting the Dataset: In machine learning data preprocessing, we divide our dataset into a training set and test set. This is one of the crucial steps of data preprocessing as by doing this, we can enhance the performance of our machine learning model. Suppose if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models. If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as:

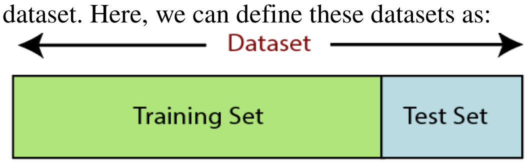

Figure 3.3: Splitting the dataset.

Training Set: A subset of dataset to train the machine learning model, and we already know the output.
Test set: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.
For splitting the dataset, we will use the below lines of code:
from sklearn. model selection import train_test_split
train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.2, random_state=0)
Explanation:
In the above code, the first line is used for splitting arrays of the dataset into random train and test subsets.In the second line, we have used four variables for our output that are

x_train: features for the training data
x_test: features for testing data
y_train: Dependent variables for training data
y_test: Independent variable for testing data
In train_test_split () function, we have passed four parameters in which first two are for arrays of data, and test_size is for specifying the size of the test set. The test_size maybe .5, .3, or .2, which tells the dividing ratio of training and testing sets.

The last parameter random_state is used to set a seed for a random generator so that you always get the same result.

K-means Clustering: Unsupervised Machine Learning is the process of teaching a computer to use unlabelled, unclassified data and enabling the algorithm to operate on that data without supervision. Without any previous data training, the machine's job in this case is to organize unsorted data according to parallels, patterns, and variations. The goal of clustering is to divide the population or set of data points into a number of groups so that the data points within each group are more comparable to one another and different from the data points within the other groups. It is essentially a grouping of things based on how similar and different they are to one another. We are given a data set of items, with certain features, and values for these features (like a vector). The task is to categorize those items into groups. To achieve this, we will use the K-means algorithm, an unsupervised learning algorithm. 'K' in the name of the algorithm represents the number of groups/clusters we want to classify our items into. (It will help if you think of items as points in an n-dimensional space). The algorithm will categorize the items into k groups or clusters of similarity. To calculate that similarity, we will use the Euclidean distance as a measurement.

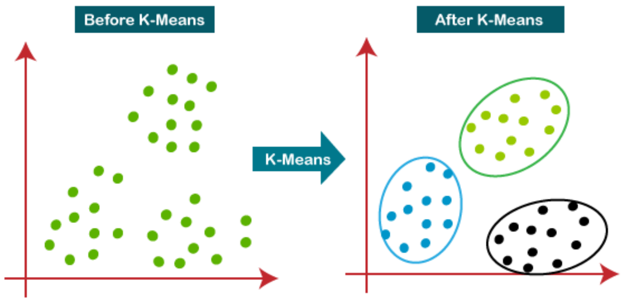

Figure 3.3: K-means clustering algorithm.

The algorithm works as follows:
Step-1: Select the number K to decide the number of clusters.
Step-2: Select random K points or centroids. (It can be other from the input dataset).
Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.
Step-4: Calculate the variance and place a new centroid of each cluster.
Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.
Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.
Step-7: The model is ready.

V. EXPERIMENTAL ANALYSIS

Figure 1 displays a visual representation of the dataset used in the analysis. It shows that the dataset contains 157,812 rows and 13 columns. Each point in the illustration represents a data point in the dataset, and the 13 dimensions (columns) are represented in a reduced form for visualization purposes. This figure helps to give an overview of the data before any analysis or clustering is performed. Figure 2 is a visual representation of summary statistics for the columns (features) in the dataset. Summary statistics typically include measures like mean, standard deviation, minimum, maximum, and quartiles for each numerical column. It provides a quick overview of the data's central tendencies and variability.

	rowID	hpwren_timestamp	air_pressure	air_temp	avg_wind_direction	avg_wind_speed	max_wind_direction	max_wind_speed	min_wind_directi
0	0	2011-09-10 00:00:49	912.3	64.76	97.0	1.2	106.0	1.6	85
1	1	2011-09-10 00:01:49	912.3	63.86	161.0	0.8	215.0	1.5	43
2	2	2011-09-10 00:02:49	912.3	64.22	77.0	0.7	143.0	1.2	324
3	3	2011-09-10 00:03:49	912.3	64.40	89.0	1.2	112.0	1.6	12
4	4	2011-09-10 00:04:49	912.3	64.40	185.0	0.4	260.0	1.0	100

min_wind_speed	rain_accumulation	rain_duration	relative_humidity
1.0	NaN	NaN	60.5
0.2	0.0	0.0	39.9
0.3	0.0	0.0	43.0
0.7	0.0	0.0	49.5
0.1	0.0	0.0	58.8

Figure 3.4: Illustration of sample dataset with 157812 rows and 13 columns used for visualizing and interpreting clustering results in IoT weather data.

Figure 3 displays the dataset after applying a feature scaling technique called standard scaling (or standardization). Standard scaling transforms the data so that each feature has a mean of 0 and a standard deviation of 1. It's used to ensure that features with different scales do not dominate the clustering process. This figure shows the standardized values for a subset of data points or features, giving an idea of how the data has been transformed. Figure 4 is a representation of the cluster centers obtained from a K-Means clustering model. In K-Means clustering, each cluster is represented by a centroid (center), and this visualizes the positions of these centroids in the feature space. Understanding the cluster centers is important for interpreting the characteristics of each cluster and making sense of the clustering results.

```
array([[ 1.3660746 , -0.08093084, -1.20641596, -0.05509583, -1.07497689,
        -0.03518016, -0.97706904],
       [-0.16488848,  0.86436187, -1.31110784, -0.58962453, -1.16696387,
        -0.60491124, -0.64082498],
       [-0.21372575,  0.62890703,  0.40849387,  0.73577497,  0.51655776,
         0.67368438, -0.14817508],
       [-1.17589877, -0.87946885,  0.44570457,  1.97075041,  0.53786973,
         1.93218831,  0.91862124],
       [ 0.23435343,  0.32056122,  1.88796075, -0.65180042, -1.55173027,
        -0.57664397, -0.28384183],
       [ 0.70934566,  0.46608965,  0.28784789, -0.53134785,  0.47452639,
        -0.53735867, -0.76936638],
       [-0.84387212, -1.19959864,  0.37577428,  0.33738679,  0.47407624,
         0.32545477,  1.3662386 ],
       [ 0.27222774, -0.99392671,  0.66393416, -0.54542686,  0.85543254,
        -0.52770955,  1.15160041],
       [ 0.13077088,  0.83909058,  1.41487216, -0.63922054,  1.6788828 ,
        -0.59017876, -0.71529989],
       [ 0.06173933, -0.78922963, -1.19736916, -0.56996748, -1.04340463,
        -0.58446022,  0.8785817 ],
       [-0.70574529,  0.52868629,  0.17440332, -0.58683671,  0.34451165,
        -0.60013333, -0.10408   ],
       [ 1.19105859, -0.25422996, -1.15485131,  2.11944328, -1.05322813,
         2.23625936, -1.13474906]])
```

Figure 3.5: Cluster centres obtained from the fitted K-Means model.

All the above figures are essential for understanding and interpreting the data preprocessing steps and the clustering results in the context of the analysis of IoT weather data. They provide insights into the dataset's structure, the impact of feature scaling, and the position of cluster centres within the feature space.

Figure 3.4, Figure 3.5, demonstrate the parallel coordinate plots for different conditions such as dry days, warm days, and cool days.

- P[P['relative_humidity'] < -0.5]: Clusters with low relative humidity are clustered as dry days.
- P[P['air_temp'] > 0.5]: Clusters with high air temperature are grouped as warm days.
- P[(P['relative_humidity'] > 0.5) & (P['air_temp'] < 0.5)]: Clusters with high relative humidity and low air temperature are clustered as cool days.

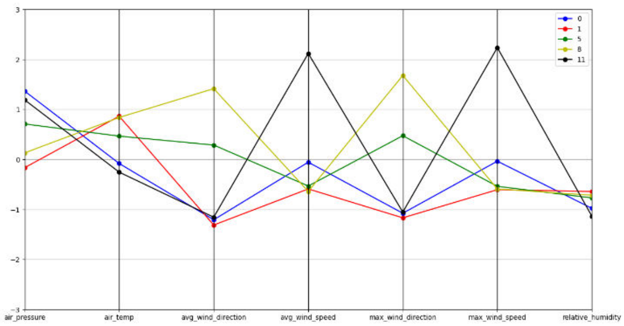

Figure 3.6: Parallel plot with clustered data as dry days.

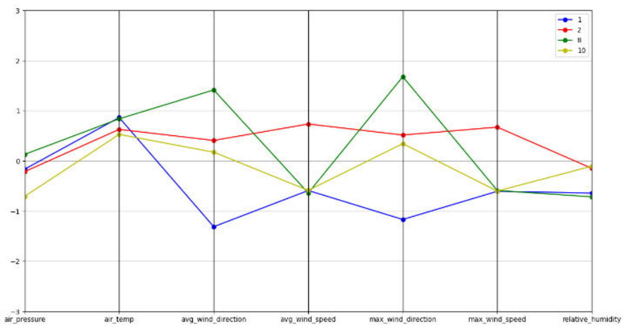

Figure 3.7: Parallel plot of clustered data as warm days

VI. CONCLUSION

This research initiates with a crucial step - standardization of the selected features. This step is pivotal in the context of K-means clustering as it ensures that all features are on the same scale. This prevents certain features from disproportionately influencing the clustering process due to their larger magnitude. Standardization is an essential prerequisite for the effective operation of the clustering algorithm. Subsequently, the K-means clustering algorithm is applied to the standardized data. The core objective is to partition the data into a predetermined number of clusters (in this instance, 12) in a manner that data points within the same cluster exhibit higher similarity concerning the selected features. The cluster centers represent characteristic values for each cluster, providing an understanding of the central tendencies of the data within each cluster. Here we learnt that how present system is better and also more efficient than the other systems. It is exceptionally compatible. It reduces human efforts. This terminate that present project work is a huge success and will provide a considerable way for saving weather parameters of real time and will help farmers, industries, normal people as well as others whose daily life is related with weather and its parameters. It can be used to get required information about for each or area for many years. The collected information will used to determine the best conditions required for plants to grow if we talk about agriculture and the farmer can modify the environment conditions which is more suitable for the plan growth. This, will have a large effect on agriculture and on farmers everywhere. This system will help in monitoring the condition of area and help individuals to work accordingly. Suppose a farmer want to grow a crop or tree which grows only in particular type of conditions. So, by this system he can see the temperature and humidity or wind direction as well as other parameters from any place. He will install this system only once and further work will be done automatically. The smart way to monitor the environment an efficient, low-cost embedded system is presented in this paper. It also sent the sensor parameters to the cloud. This data will be helpful for future analysis and it can be easily shared to other users also. This model can be expanded to monitor the developing cities and industrial zones for pollution monitoring. To protect the public health from pollution, this model provides an efficient and low-cost solution for continuous monitoring of environment.

VII. REFERENCES

1. D. Kumar, J. C. Bezdek, S. Rajasekhara, C. Leckie and M. Palanisami, "A visual-numeric approach to clustering and anomaly detection for trajectory data", 2017
2. D. Kumar et al., "Adaptive cluster tendency visualization and anomaly detection for streaming data",2016.
3. D. Kumar, M. Palanisami, S. Rajasekhara, C. Leckie, J. C. Bezdek and T. C. Havens, "clusiVAT: A mixed visual/numerical clustering algorithm for big data",2013.
4. Yan-e, D., 2011, March. Design of intelligent weather management information system based on loT. In Intelligent Computation Technology and Automation (ICICTA), 2011
5. Kamilaris, A., Gao, F., Prenafeta-Boldú, F.X. and Ali, M.I., 2016, December. Agri lot: A semantic framework for Internet of Things-enabled smart monitoring applications. In Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum.
6. C. Havens and J. C. Bezdek, "An efficient formulation of the improved visual assessment of cluster tendency (iVAT) algorithm",2011. Alfred Inselberg. Parallel coordinates: VISUAL multidimensional geometry and its applications, Spring, 2007
7. Koren Y. and Harel D. A two-way visualization method for clustered data. In Proc. 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, 2003
8. Bar-Joseph Z., Gifford D.K., and Jaakkola T.S. Fast optimal leaf ordering for hierarchical clustering. Bioinformatics, 17(90001):22–29, 2001.
9. Ankerst M., Breunig M.M., Kriegel H.-P., and Sander J. Optics: Ordering points to identify the clustering structure. In Proc. ACM SIGMOD Int. Conf. on Management of Data, 1999, pp. 49–60.
10. A. K. Jain and R. C. Dubes, “Algorithms for Clustering Data”, Englewood Cliffs, NJ: Prentice Hall, vol. 6, 1988.