Air Quality Characterization Using *k*-Nearest Neighbors Machine Learning Algorithm via Classification and Regression Training in R

Timothy M. Amado

***Abstract* — Through the years, environmental health and protection have been ignored. However, because of recent phenomena such as climate change, people are slowly becoming aware of the environment. One of the main concerns nowadays is air pollution. To this avail, the U.S. Environmental Protection Agency (EPA) standardized air quality with the use of air quality index (AQI). However, AQI requires accurate sensor readings and complex calculation to obtain. Hence, the objective of this paper is to solve that problem by characterizing the air quality with regards to AQI through the use of *k*-nearest neighbors machine learning algorithm. The proposed methodology is implemented using a prototype of integrated gas sensors for data gathering. R programming, focusing on classification and regression training (caret) package for data processing, model development, and algorithm tuning, is utilized. The system is evaluated, and an accuracy of 99.56% is obtained.**

Keywords: **air quality characterization, AQI, KNN machine learning, sensor networks, r programming, caret**

I. Introduction

In the advent of the modernization age, the heightened awareness of people with regards to environmental and health protection puts air pollution as one of the main concerns of society. In fact, according to a report of the World Health Organization, air pollution has been the biggest environmental health risk [1]. Thus, several researches have been developed and pioneered to be able to mitigate the risks that air pollution brings to the environment.

Timothy M. Amado, Electronics Engineering Department Technological University of the Philippines–Manila. (email: timothy_amado@tup.edu.ph)

Air quality monitoring is one of the best ways to help fight against air pollution. By knowing the air quality, suggestions can be made in order to help alleviate its dangerous effects. Most of the air quality monitoring device relates the values they obtain to the air quality index (AQI). AQI is a system that standardized the levels and severity of polluted air. As the index increases, the more harmful the air would be for people, having the index of 500 being the most hazardous [2].

However, values obtained from the sensors do not show the immediate values of the AQI. And based on the available resources, calculation of sensor values relating to AQI can be somehow difficult to do. This can impose a problem when constructing prototypes of portable air quality monitoring devices.

The goal of this paper is to characterize the air quality by building a model that relates the sensor values to AQI. Machine learning is used to create the model through *k-nearest neighbor (KNN)* algorithm using R programming.

II. Related Works

This section presents the past studies done in the field of air quality monitoring.

Wang and Chen [3] developed a system that uses vehicular sensor networks (VSNs) to monitor the air quality in a city. In this study, the researchers proposed using VSNs to tactically monitor the air quality and develop an efficient data gathering and estimation (EDGE) mechanism. AQI is highlighted in this paper as the main standard for the monitoring accuracy.

Li and He [4] proposed an intelligent system for indoor air quality monitoring and purification. In this paper, the authors used wireless sensor networks to achieve the air quality monitoring, having an STC12C5A60S2 microcomputer as a core and MQ138 as the gas sensor. Air purification is achieved using a high-efficiency particulate air (HEPA) filter.

Molka-Danielsen et al. [5] presented a system using big data (BD) analytics on the analysis of the data of the

CO_2 levels on a logistics shipping base on Norway. The data were measured using wireless sensor networks, and the researchers used BD as a decision support system for the health and safety of the workers in the shipping industry.

A mobile and cost-effective platform for particulate matter (PM) air quality monitoring is proposed by Wu et al. [6]. The system uses lens-free microscopy to analyze the PM particles present in the air and machine learning to perform sizing and counting of the PM particles.

Chiwewe and Ditsela [7] suggested a method that can be used to estimate and predict different levels of pollutants, concentrating on the ozone. The authors used a multilabel classifier based on Bayesian networks machine learning algorithm to estimate the probability of the pollutants exceeding a certain threshold based on the AQI.

III. THEORETICAL BACKGROUND

This section details the theoretical knowledge used by the researchers in characterizing air quality of air. The first part gives a background about AQI, a standardization of U.S. Environmental Protection Agency (EPA) for air quality. The second part gives a run-through of the KNN machine learning algorithm.

A. AQI

After the conceptualization of the problem needed to be solved in this study, published papers, journals, and articles are consulted. Most notable studies are cited in section II of this paper.

Figure 1 shows the standard for AQI published by the U.S. EPA. The AQI is the main basis of the characterization done in this paper. It shows the levels of health concern as well as the color symbol for each AQI range.

Air Quality Index (AQI) Values	Levels of Health Concern	Colors
When the AQI is in this range:	*..air quality conditions are:*	*...as symbolized by this color:*
0 to 50	Good	Green
51 to 100	Moderate	Yellow
101 to 150	Unhealthy for Sensitive Groups	Orange
151 to 200	Unhealthy	Red
201 to 300	Very Unhealthy	Purple
301 to 500	Hazardous	Maroon

Fig. 1. AQI with color symbols and levels of health concerns [3].

According to U.S. EPA, there are six levels of air quality/pollution. Recommendations are given by the U.S. EPA in the event that certain values of AQI are observed [8].

AQI shows that air quality is standardized; however, it is not clear in the EPA article how these AQI ranges relate to exact sensor values. Hence, actual values of AQI cannot be displayed in situations where a standalone air quality monitoring device is used. Some literature provides this information as in the paper of Wang and Chen [3], where they used (1) to relate sensor values to AQI range:

$$I_k = \frac{I_{\text{high}} - I_{\text{low}}}{B_{\text{high}} - B_{\text{low}}}\left(C_k - B_{\text{low}}\right) + I_{\text{low}} \tag{1}$$

where B_{high} is a breakpoint $\geq C_k$, the average concentration of the pollutant, B_{low} is a breakpoint $\leq C_k$ and $I_{\text{high}}/I_{\text{low}}$ denotes the AQI value for $B_{\text{high}}/B_{\text{low}}$. I_k is the AQI of a given pollutant. The highest I_k calculated will be the AQI of the air being monitored [8].

B. KNN Machine Learning Algorithm

The KNN machine learning algorithm refers to a nonparametric supervised machine learning algorithm that utilizes both the nominal and numerical attributes of data by selecting the most common attribute between the KNNs or by getting the average of the values of the KNNs. This machine learning algorithm is one of the top 10 most important data mining algorithms [9].

To identify which of the attributes from the k instances in the training data set is the most similar to a new input, distances are measured. The most commonly used distances in KNN algorithm are the Euclidean distance (2), Manhattan distance (3), Minowski distance (4), and Hamming distance for the discrete data.

$$d = \sqrt{\sum_{i=1}^{k}\left(x_i - y_i\right)^2} \tag{2}$$

$$d = \sum_{i=1}^{k}\left|x_i - y_i\right| \tag{3}$$

$$d = \left(\sum_{i=1}^{k}\left(\left|x_i - y_i\right|\right)^q\right)^{1/q} \tag{4}$$

As per the appropriate value of k, algorithm tuning can be done choosing what value tailor-fits to the requirement of the given training set. The following algorithm shows the implementation of KNN machine learning [10].

Algorithm 1: *k*-Nearest Neighbors machine learning algorithm pseudocode

Input: Dataset E, Instance to classify x, Value k;

1: Calculate $d(x, x_i)$, i =1, 2, ..., n; where d denotes the Euclidean distance between the points.
2: Arrange the calculated n Euclidean distances in non-decreasing order.
3: Let k be a positive integer, take the first k distances from this sorted list.
4: Find those k-points corresponding to these k-distances.
5: Let k_i denotes the number of points belonging to the i^{th} class among k-points i.e. $k \geqslant 0$.
6: If $k_i > k_j \; \forall \; i \neq j$ then put x in class i.

Output: x_n

Aside from the simplicity of implementation, KNN is a very good algorithm for characterization if the data set is small. It has a very good predictive power and can accommodate data sets even without prior knowledge of the structure of data, which is very much applicable in the proposed methodology of this paper.

IV. METHODOLOGY

In this paper, the authors propose a method of characterization of the air quality in reference to the AQI set by the U.S. EPA.

A. Hardware Development

To be able to gather data for the construction and training of the model to be used in the machine learning algorithm, the researchers built a prototype of the air quality monitoring device consisting of an integrated array of sensors, an Arduino microcontroller, and an exhaust fan. The sensors used in this study are an air temperature sensor, a humidity sensor, and MQ2, MQ135, and MQ5 gas sensors. Table I shows the detailed specifications of each sensor used in constructing the hardware.

TABLE I
SENSORS USED FOR HARDWARE DEVELOPMENT

Sensor	Description
DHT 11	Temperature and relative humidity sensor module for Arduino
MQ2	General combustible gas sensor (methane, butane, smoke)
MQ5	Natural gas sensor (H_2, LPG, CH_4, CO, alcohol)
MQ135	General air quality sensor (NH_3, NOx, CO_2, benzene)

The sensors are selected based on their availability in the local market and compatibility with the Arduino microcontroller. These sensors will be enclosed in a compartment where the exhaust fan will also be installed. Holes are drilled facing the sensors to allow the sample air to go inside the device. Figure 2 shows the different layout views of the proposed system.

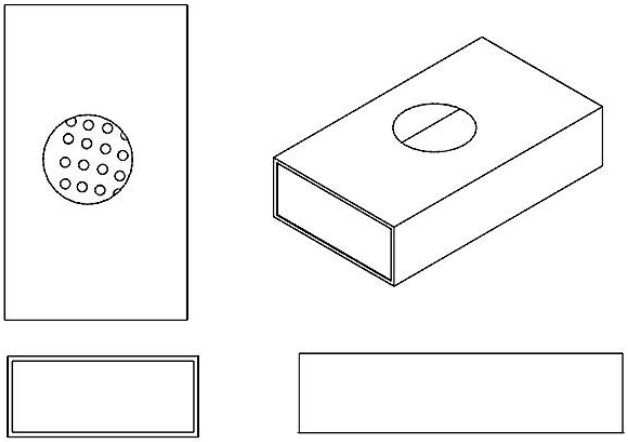


Fig. 2. Top, front, side, and isometric view of the chassis where the components of the prototype will be placed.

The prototype derives its power from a laptop computer when connected using a serial-to-USB connector. The data from the sensors are fetched by a script written in Python and written it in a comma-separated variable (CSV) file for every data collection session. A battery pack and SD card module can be installed on the system in the next iterations of the prototype.

B. Data Gathering

Next step after completing the project prototype is the data gathering. A certain scheme developed by the authors is used in gathering data since the proposed methodology will not be depending on the actual concentration readings of the sensors. Five environment conditions where air samples will be collected are chosen. These five conditions correspond to the AQI ranges stated in [3] except the "hazardous" condition (AQI $\geq$ 301). The researchers didn't choose an environment having this condition because of the health risks this AQI range might give. Table III shows the environmental condition for each AQI range/health concern cluster. These conditions are chosen based on readings from [3] and [8].

TABLE II
DATA GATHERING SCHEME

AQI Range	Levels of Health Concern	Environmental Condition
0–50	Good	Air-conditioned and properly ventilated room
51–100	Moderate	Non-air-conditioned room with poor ventilation
101–150	Slightly unhealthy	Outdoors beside a busy road
151–200	Unhealthy	Direct exposure to smoke from cars, trucks, motorcycles
201–300	Very unhealthy	Direct exposure to fumes of combustible gas, LPG, and fire

The data collection scheme that used in this paper includes deployment of the prototype on each of the environmental condition stated at Table II. A total of 150 sets of sensor data points are collected in each cluster.

C. Preprocessing, Development, and Training of the Model

This section presents the details and steps taken by the researchers in data preprocessing, development, and training of the KNN model.

1) Getting and cleaning the data. This is usually the first step of the development of every machine learning algorithm after the data has been gathered and collected. The data from the sensors are fetched into several CSV files per each cluster of observations. The data coming from the sensors are not organized. Missing values (NAs) can be encountered along the way. Also, because of the limitations of the Arduino microcontroller, the data obtained do not contain the environmental condition where the data are measured. This step takes care for all of those premises by getting and organizing the data from the CSV files, as well as cleaning them by removing any NAs and 0s. Finally, the environmental conditions where the data are measured are appended. All of these are accomplished using an R script.

2) Visualizing the data. This step involves several interactions with the R environment to give the researchers a thorough understanding of the data gathered. This involves getting plots for visual data representations, obtaining correlations, and viewing the summary of the overall data. These steps include the use of the `ggplot2` and `ggvis` packages in R.

3) Data slicing. After getting an overview of the data, the next step is to slice it into training set and testing set in preparation for training and tuning the resulting model. This is accomplished by using the classification and regression training `caret` package in R. In this paper, the authors adopted the standard 70% training and 30% testing data partition.

4) Data centering and scaling. Based on the results of the data visualization and overview, normalization, which include data centering and scaling, may be applied to the training set if the data ranges are nonuniform to standardize it. If the data are already uniform, this step may be skipped; however, in most of the cases, the data obtained need normalization in preparation for training. This is again achieved using the `caret` package in R.

5) Training the network. After data has been normalized, the KNN model can now be established. The `caret` package in R provides some useful tools to make it easier to establish and train a certain machine learning algorithm model. For the KNN, the training method that will be used to obtain the best model is the repeated cross-validation method.

In repeated cross-validation method, the training data are randomly divided into k sets or *folds* of approximately equal sizes. The KNN model is formed using all the samples except the first fold, after which the prediction error of the KNN is obtained using the first fold. The same process is repeated for each fold. The performance of the KNN model is calculated by averaging the errors obtained from the different folds.

In this paper, the researchers used the standard 10-fold, 10-repeats cross-validation method [11] in evaluating the performance of the KNN model.

6) Algorithm tuning. Algorithm tuning is a blind search methodology for the appropriate value for k to be used in the KNN algorithm. The `caret` package also provides a neat tool to employ algorithm tuning.

D. Testing and Evaluation

The KNN model will be tested using the test data set obtained from the partition of the original data set. The target accuracy is 95% to make it at par with the current KNN machine learning algorithms.

V. RESULTS AND DISCUSSION

This section presents the results of the methodologies done in section IV of this paper. This includes presentation of the actual hardware prototype, as well as the results of the data gathered together with the KNN model.

A. Project Prototype

The following images (Fig. 3) present the actual project prototype used in the data gathering phase of this paper.

Fig. 3. Different views of the protoype used in the data gathering phase.

The data can be read from the console of the Python IDLE. Because sensors take some time before stabilizing, data must be monitored in the console before being saved as CSV.

B. Overview of the Data

This section presents the overview of the data during the getting, cleaning, and visualization phases. All of the processes are done using R scripts and R packages.

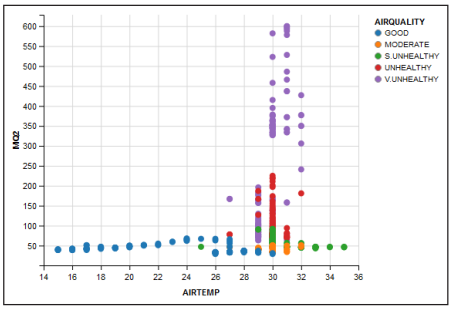
Table III shows the head of the sampled cleaned data. Here, missing values are already removed from the data set, and categories are added based on the environmental condition where the data were originally measured.

The data frame is composed of 750 observation points with six variables corresponding to sensor values and the air quality classification.

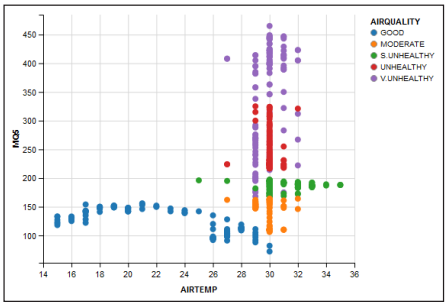
Using the `ggvis` package, several plots are obtained to give a further visualization of the data at hand. Figure 4 shows several scatter plots relating sensor values to temperature.

TABLE III
HEAD OF THE SAMPLED CLEANED DATA

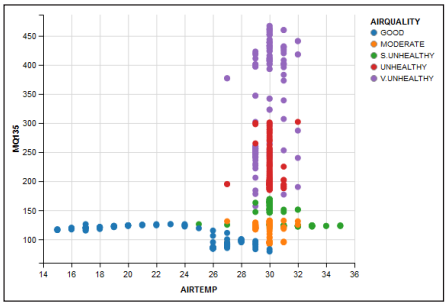
Obs. No.	Air Temp (°C)	Relative Humidity (%)	MQ2 (ppm)	MQ135 (ppm)	MQ5 (ppm)	Air Quality
551	30	73	34	95	117	Moderate
645	16	75	42	120	130	Good
411	32	66	46	123	188	S. Unhealthy
604	31	80	74	194	224	Unhealthy
727	29	84	139	265	239	V. Unhealthy
540	31	80	39	118	148	Moderate



(a)



(b)

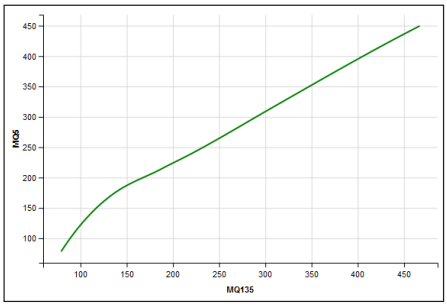


(c)

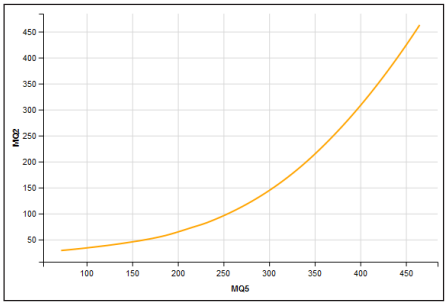
Fig. 4. Several scatter plots showing relationship of air temperature and sensor values: (a) air temperature versus MQ2, (b) air temperature versus MQ5, and (c) air temperature versus MQ135.

Based on the scatter plots, it can be observed that generally, for naturally low-temperature environments, air quality is "good" as indicated by a spread of blue points on the lower temperature section of each scatter plot. It can also be observed that for naturally high-temperature environments, high sensor values tend to appear. As for the clustering, it can be seen that only few outliers exist and majority of the data for each air quality condition tend to group up at specific portions of the plot. This implies that categorization using a machine learning model is very viable.

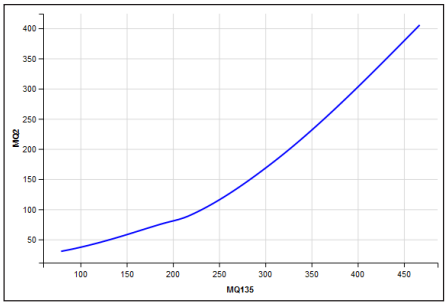
Figure 5 shows the smoothed curve of relationship between sensor values. The smoothed curve is obtained using the *locally weighted scatter plot smoothing* model or commonly called as the *LOESS model*. The LOESS model performs local fitting to give the best fit curve for data relationships. It is noted however that LOESS produces a model that does not result in an explicit form of an equation. However, prediction, interpolation, and extrapolation can be done by simulating the model [12].



(a)



(b)



(c)

Fig. 5. Smoothed curve using the LOESS model: (a) MQ5 versus MQ135, (b) MQ2 versus MQ5, and (c) MQ2 versus MQ135.

It can be seen that as reading on a sensor increases, the other sensor values also increase. This means that there is a common component (a common gas that both sensors detect) between each sensor. It implies that as a future directive, principal component analysis may be employed to further characterize the data before applying a machine learning algorithm.

C. *Trained KNN Model Results*

After having a comprehensive overview and understanding of the data at hand, the development of the model comes next. This section presents the results of fitting and training a KNN machine learning algorithm for the characterization of air quality data. Training is done using the `caret` package.

Table IV shows the possible values of k obtained after algorithm tuning. The system automatically chooses the value of k based on the accuracy and kappa value of the model. Kappa is one of the machine learning metrics that is similar to classification accuracy but more useful when the distribution in of frequency in each class is nonuniform.

TABLE IV
ALGORITHM TUNING

k	Accuracy	Kappa
5	0.9959992	0.9949969
7	0.9958106	0.9947610
9	0.9954401	0.9942981
11	0.9937304	0.9921601
13	0.9899011	0.9873726
15	0.9893275	0.9866556
17	0.9885660	0.9857031
19	0.9889584	0.9861937
21	0.9887582	0.9859431
23	0.9883914	0.9854845

It can be seen from the figure that the best value is $k = 5$. The repeated cross-validation method for 10-folds and 10 repeats are used to map the accuracy of the model.

D. *Test and Evaluation Results*

The KNN model developed is tested using a test data set obtained from performing data slicing on the original data set. The overall accuracy for KNN model is 99.56%.

Moreover, Table V shows the overall statistics of the KNN model.

TABLE V
SUMMARY OF OVERALL MODEL STATISTICS

Accuracy	0.9956
95% Confidence interval	0.9755, 0.9999
No information rate (NIR)	0.2
p-Value (Acc > NIR)	$p < 2.2e\text{-}16$
Kappa	0.9944
Mcnemar's test *p*-value	NA

From Table V, the summary of the overall statistics of the model developed can be seen. Notable values here are the "no information rate," which basically means the largest class percentage in the data (which is 20% because there are five classes equally distributed). The *p*-value is obtained from a one-sided test taken to check if there is a significant difference for the accuracy when the data is considered only for the majority class. Since our *p*-value is small, it implies that the accuracy doesn't have a significant difference whether it came from the majority class or not. McNemar's test *p*-value is not applicable because the classes are sparse.

VI. CONCLUSIONS

Based on the data and results, the proposed method of characterization of the AQI using KNN machine learning algorithm is implemented successfully. A project prototype made up of an array of sensors is developed. A KNN machine learning model is established with 99.56% accuracy. Overall, the statistics of the model is excellent and accurate.

ACKNOWLEDGMENT

The author acknowledges Mapua University for the support in terms of access to the IEEE database that led to the vast improvement of this paper. The author would also like to thank the Department of Science and Technology – Philippine Council for Industry, Energy and Emerging Technologies for Research and Development (DOST-PCIEERD) for the Data, Connectivity and Intelligence: Data Science track training received, which proved to be a valuable asset that led to the fruition of this paper.

REFERENCES

[1] World Health Organization (WHO), "7 million premature deaths annu- ally linked to air pollution," Mar. 2014. [Online]. Available: http://www. who.int/mediacentre/news/releases/2014/air-pollution/en/.

[2] U.S. Environmental Protection Agency, "Air quality guide for particle pollution," US EPA, 2015.

[3] Y.C. Wang and G.W. Chen, "Efficient data gathering and estimation for metropolitan air quality monitoring by using vehicular sensor networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7234–7248, 2017.

[4] Y. Li and J. He, "Design of an intelligent indoor air quality monitoring and purification device," in *2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC)*, 2017, pp. 1147–1150.

[5] J. Molka-Danielsen, P. Engelseth, V. Olesnanikova, P. Sarafin, and R. Zalman, "Big data analytics for air quality monitoring at a logistics shipping base via autonomous wireless sensor network technologies," *2017 5th Int. Conf. Enterp. Syst.*, pp. 38–45, 2017.

[6] Y. Wu et al., "Mobile microscopy and machine learning provide accurate and high-throughput monitoring of air quality," in *2017 IEEE Conference on Lasers and Electro-Optics*, 2017.

[7] T. M. Chiwewe and J. Ditsela, "Machine learning based estimation of Ozone using spatio-temporal data from air quality monitoring stations," in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, 2016, pp. 58–63.

[8] U.S. Environmental Protection Agency, "Technical assistance document for the reporting of daily air quality—The air quality index (AQI)," US EPA, Dec. 2013.

[9] R. Agrawal, "*k*-nearest neighbor for uncertain data," *Int. J. of Computer Applications*, vol. 105, no. 11, pp. 13-16, 2014.

[10] J. M. Cadenas, M. C. Garrido, R. Martinez-Espana, and A. Munoz, "A more realistic *k*-nearest neighbors method and its possible applications to everyday problems," in *2017 IEEE International Conference on Intelligent Environments (IE)*, 2017, pp. 52–59.

[11] S. Borra and A. Di Ciaccio, "Measuring the prediction error. A comparison of cross-validation, bootstrap and covariance penalty methods," *Comput. Stat. Data Anal.*, vol. 54, no. 12, pp. 2976–2989, 2010.

[12] W. S. Cleveland, E. Grosse, and W. M. Shyu. "Local regression models," in *Statistical Models in S*, J.M. Chambers and T.J. Hastie, Eds. Wadsworth & Brooks/Cole, Pacific Grove, California, 1992.