Buono, P., Balducci, F. MonitorApp: a web tool to analyze and visualize pollution data detected by an electronic nose. Multimed Tools Appl 78, 33023–33040 (2019). https://doi.org/10.1007/s11042-019-7676-3

Noname manuscript No.
(will be inserted by the editor)

MonitorApp: a Web Tool to Analyze and Visualize Pollution Data Detected by Electronic Nose Devices

Paolo Buono · Fabrizio Balducci

Received: November 30^{th} 2018 / Accepted: date

Abstract The quality of the air is assessed by sensors in monitoring stations that measure the concentration of specific chemical compounds that may affect people's health. Assuming that some chemical compounds in the air produce a bad smell, people may detect that something is going wrong acting as sensors that alert potential risks. This paper proposes a visual analytics approach to support air quality experts in the analysis of data produced by electronic nose devices. Experts create or modify data workflows to manage and transform raw data, then apply clustering and visualization techniques to get insights. The proposed approach is supported by calendar, map and line graph visualizations also maneuvering clustering attributes and methods. An interactive map is used to show the position of monitoring stations in order to support making hypotesis related to data source locations.

Keywords visual analytics · data mining · analysis workflow · environment

1 Introduction

Since the Industrial Revolution, the problem of air quality has revealed to be serious for public health. The attention to air quality is recently increased, due to the understanding of its role in the human quality of life. Consequences of this increased awareness is the attitude to a progressive reduction of air pollution, at least in Europe [1]. Pollution relates to chemical, but also physical and biological agents that usually are present in low percentages in the

Fabrizio Balducci
University of Bari Aldo Moro
E-mail: fabrizio.balducci@uniba.it

Paolo Buono
University of Bari Aldo Moro
E-mail: paolo.buono@uniba.it

air [2] [3]. People living in high density populated cities and inside industrial areas surroundings are more exposed to this problem.

Air quality monitoring consists in the measurement and storage of potentially hazardous air agents. The monitoring process is not meaningful without the comparison of data with some reference to identify trends and irregularities, to assess compliance both with legal standards and for scientific research, by producing environmental technical reports and forecasts. The main goal is to protect people and, in general, their environment, specifically in the areas that produce pollution In order to produce good analysis, airborne agents monitoring often requires information about weather parameters, monitored area topology and wind information.

Air quality data are now also available as open data on Internet and governments are moving towards open data; an example is the Italian Public Administration Open Data [4]. More related to the worldwide environment, the *Air Pollution in world* [5] web site shows markers in a planisphere, each representing air quality data of the referenced area that a user, by clicking on it, can access to pollution forecast. The work presented in this paper has been performed in collaboration with environmental experts, which analyze data produced by monitoring station sensors on a daily basis: it proposes a workflow that follows a typical analysis pipeline including pre-processing activities and adds visualizations for data validation and visual analysis, in order to help analysts to discover novel and unexpected insights or confirm hypotheses. This work is based on a previous work by Buono and Costabile in [6] and extends the research proposed in [7] to support data analysis using a *Visual Analytics approach*; it adopts a workflow model that now includes an improved pre-processing, a clustering algorithms and a geographic map useful to locate monitoring stations on the territory. The process can be repeated every time new data arrive to the analysis server and can be modified by the experts. The pipeline exploits a clustering algorithm to group monitoring devices identifying hidden relationships.

Results are visualized using different techniques on heterogeneous data attributes; such techniques are neither highly interactive nor complex time series visualization to allow novel users to easily use and interpret the visualization: the focus is more on the combination of visualization and representation than just on the techniques.

The work is organized as follows: Section 2 presents the related work, Section 3 introduces the system design rationale. The developed framework is described in Section 4 showing examples and visualization results. Finally, Section 5 draws conclusions.

2 Related Work

The problem of air quality pollution management and analysis has been faced by many works in literature, one of the main problems is to process data and present results to the user in a way to allow decision makers to quickly take

decisions. Kandel et al. [8] propose an interactive visual approach to deal with data pre-processing through Data Wrangler, a tool that allow expert systems to speed this long and boring phase; Keim [9] first proposed a pixel-based technique to visualize time data in compact space using cyclic series, time data visualization has been addressed by Aigner et al. [10], who proposed a collection of visualization techniques. Ceneda et al. [11] [12] face the analysis of cyclical patterns in time-series and the interactive visualization techniques that allow analysts to identify recurring behavior using and comparing *Spiral and calendar-based* visualizations.

An interesting technique that considers human activities over time is the *Calendar visualization* made by Van Wijk and Van Selow [13] who first proposed a metaphorical scheme where the time dimension of a phenomenon is portrayed by a succession of monthly calendar icons; each calendar features interactive day icons (details on demand) that can be colored according to a sequential scale adding a further data dimension to the arranged data.

Other proposals address time similarly: Malik et al. [14] introduce VALET (Visual Analytics Law Enforcement Toolkit), a system that shows crime trends and distribution on a maps: the Calendar view is exploited to display criminal events occurred in the days linked to other events like social and sport events, highlighting relationships with seasonal and cyclical trends providing histograms and density heatmaps. Razip et al. [15] extended the previous work on mobile devices by displaying the crimes distribution in relation to the user current location.

Zhou et al. [16] analyze one of the most developed areas of China finding a poor AQI (Air Quality Index) that deteriorates from North to South employing a hierarchical clustering. Moreover, Li et al. [17] visualized pollution in Beijing exploiting a dataset featuring the period from 2009 to 2014 and another one built from data produced by 36 monitoring stations: the averaged pollution concentration is displayed through two circular clock-sliced heatmaps.

Time series data can be analyzed by several methods [18], for example the ARMA (autoregressive, moving average) model by Box and Jenkins [19] can be used to predict future values while Sharma et al. [20] and Bontempi [21] face particularly with multi-sensor Visual Analytics supported by machine learning techniques. In the domain of climate data analysis, Kappe et al. [22] [23] explore variability climate predictions and propose an interactive visualization technique (clustering timeline) together with filled-contour maps; Zhu et al. [24] and Wang et al. [25] employs interactive geographic virtual environments while also in Kern et al. [26] atmospheric meteorological data are managed to visualize fronts in two dimensions obtaining frontal surfaces in three dimensions using the magnitude of temperature change; Sarikaya et al. [27] review the literature surrounding dashboard use.

The problem of using Visual Analytics approach to explore time data has been addressed by several authors; examples are Ellis [28] and de Carvalho et al. [29]. Electronic nose technology and data format and management has been introduced by Pearce [30], Scott [31] while Buono and Costabile [6] aimed at

understanding how much domain experts are able to use a visual workflow tool like KNIME in order to process pollution data to perform analysis.

3 Users, Data and Tools

Referring to the literature in Section 2, Table 1 shows a comparison with the approach proposed in this paper, highlighting the advantages and the features of each one.

The case study focuses on the problem of pollution and air quality detection, providing users with a replicable process in order to repeat and customize the analysis and its process pipeline: starting from gathering real-time data from specific devices up to integrating a visual workflow platform with clustering algorithms and various information visualization tools.

3.1 Users

The proposed tool refers to the chemists addressed in [6] who periodically compare and overlap line graphs performing exploratory data analysis, in order to detect pollution anomalies preventing environmental risks. In order to design adequate tools by including users and experts in prototype evaluations, a user-centered and participatory paradigm has been adopted. In the design phase a user test was performed with two evaluators and four domain experts which executed activities with a visual workflow tool ranging from data acquisition to information visualization.

The use case of a running example proposed in this article includes a server that always listens to requests from the interface client; when a specific request that uploads new data produced by monitoring station is added, a batch workflow starts. The workflow automates the access to the archives containing temporal data and, after their clustering, returns the results to the server. The analyst can easily modifies the parameters and the workflow pipeline by adding/removing custom tools and components, visually, eliminating the need to deal with programming code, which is time consuming, requires specific skills and is not appreciated by non-expert users.

With the provided results, the client allows users to interact with *Views*, interfaces to frame and analyze in detail the information of interest, also allowing to overlap them in a way that replicates the original specific analysis practices of the users.

3.2 Electronic Nose Device Data

The *Electronic Nose* (Fig. 1) is a device that detects the impact of odoriferous chemical compounds. It has been used to continuously collect information about airborne agents in industrial areas with the aim to track seasonal trends and isolate abnormal emissions.

Table 1 Comparison of the literature with the approach proposed in this work.

Method	Features	Our approach
Data Wrangler (Kandel [8]	+ performs automatic inference that speeds-up data log transformation into the desired format – the working pipeline is hidden to the user	+ allows more control – requires higher skills to use and adapt the pipeline workflow
Pixel-based (Keim [9])	+ provides a compact representation, typically reduced to a single variable + color changes provide cues about the data distribution – suffers in providing details about data	+ uses combination of calendar/color based visualizations to help in details identification + the line charts overlapping partially compensate the scalability and better shows the timeline – less scalable because it uses bigger boxes
Ceneda [11] [12]	+ provides a good cycle detection – does not specifically address environmental data	+ line chart helps in identifying patterns over time – does not consider cycles (considered irrelevant for the domain experts due to the well-known seasonal trends)
Calendar (Van Wijk [13])	+ it uses a color coding to identify clusters on a calendar	+ it proposes a pipeline that allow the analyst to change parameters, clustering method and visualizations + Map view to identify spatial correlations + line chart view to compare many monitoring station at once
VALET (Malik [14] and Razip [15])	+ it proposes advanced maps – lack in time line visualization – in Razp et al the calendar view is not specific for the environment domain but for single punctual events (crimes)	+ clearly visualizes the evolution in time of environmental phenomena + easily adaptable to non-continuous punctualevents
Li [17]	+ contains different visualizations – it mainly focuses on PM 2.5 – it lacks of the support to the pre-processing phase – it is not clear how to extend it with further techniques	+ data attribute and features very numerous due to the ease of uploading heterogeneous data + allows to perform thorough pre-processing analysis + easily extensible and maintainable with new tools

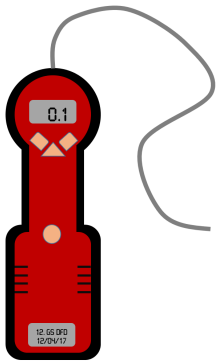

Fig. 1 A drawing of an electronic nose device used to capture odoriferous chemical compounds values.

Table 2 Numeric variables from the log data.

1) Date (dd/mm/yyyy)	*2) Time (hh:mm:ss)*	*3) Temperature (° C)*
4) Humidity (%)	*5) Pressure (mBar)*	*6) PID (ppm)*

```
Date;Time;Temperature;Humidity;Pressure;PID;Battery
;;degC;%;mBar;ppm;V
"16/02/2017";"13:36:34";"19,1";"18,0";"952,9";"0,237";"4,05"
"16/02/2017";"13:37:34";"19,2";"17,7";"952,9";"0,235";"4,05"
```

Fig. 2 Log file excerpt produced by a monitoring station.

All collected data are in the form of text logs and have been gathered in the period February-July 2017; before this time there were two pilot stations which provided information useful to depict a real-world case about missing, wrong or anomalous data and outliers. The structure of the dataset is depicted in Tab. 2 featuring seven numeric variables (the battery power level of the device is not relevant for the next analysis); an example of how these data are arranged in a .csv file structure is reported in Fig. 2 with the attribute names (Date, Time, Temperature, Humidity, Pressure, PID) and their measure units in the first line, followed by lines of values.

3.3 The KNIME Tool

In order to implement the working pipeline, we used KNIME (Konstanz Information Miner), an open source tool that allows an easy inclusion of modules exploiting a workflow visual paradigm.

A workflow is visually composed by *nodes* positioned into an *Editor* where each node can be connected to others. The node status can be *closed*, *inactive*, *running* or *complete* and it is represented using the traffic light convention: *red* (error or to configure), *yellow* (ready to run) and *green* (output ready and

available). Input and output ports, respectively at the left and right sides of each node represent handles to external sources or to other nodes. Through a *Data port* data are transferred between nodes; *General purposes* ports return data in the form of a database table.

3.4 The WEKA Tool

Another component of the proposed tool is WEKA (Waikato Environment for Knowledge Analysis), an open source framework with a user-interface for machine learning models and functions.

The *Project Explorer* is useful to manipulate data coming from heterogeneous sources through operators like filter, attribute selection, classification and regression models, clustering algorithms and association rules, graph display. To employ and execute the different machine learning techniques and models along with their statistics, the *Experimenter* environment is available.

Data used for the case study presented in this paper are *time series*, that is datasets characterized by a sequence of N pairs (y_i,t_i), $i = 1...N$ where y_i is the value measured at the time t_i. To analyze this kind of series it is useful for analysts to cluster days with similar atmospheric characteristics and observe them through various visualizations, according to different dimensions. Given m clusters on a daily pattern, their similarity is calculated and, the pair of cluster with the highest similarity are merged in a single new one; the process continues with $m-1$ clusters. The merging of cluster continues until reaching a stop criterion which ensures that each single cluster contains patterns of similar days

while various measures can be used to compute the cluster similarity like the *Euclidean distance* or the *Manhattan* one.

With respect to the *K-Means* phase in the KNIME workflow, numeric distance measures have been exploited to explore the cluster similarity in the following form: Assuming y_i and z_i two daily patterns of the collection N, the *Euclidean distance* is computed as average square difference and the normalized version is in (1):

$$d_{nm} = \frac{\sqrt{\sum_{i=1}^{n} (\frac{y_i}{y_{max}} - \frac{z_i}{z_{max}})^2}}{N} \tag{1}$$

A way to select the most important and useful data attributes for a good clustering is to consider the *Correlation*: in (2) the *Pearson product-moment correlation coefficient*, used to compare two variables A and B, calculates for each pair of attributes a coefficient r that is a measure of the correlation grade between two variables. The correlation value ranges from -1 (strongly negative correlation) to 1 (strongly positive correlation) while 0 does not represent any linear correlation. In particular, $r > 0.7$ evidences a strong local correlation that can be direct (positive sign) or inverse (negative sign) while $0 > r > 0.3$ and $0.3 > r > 0.7$ represent respectively moderate and slight correlations.

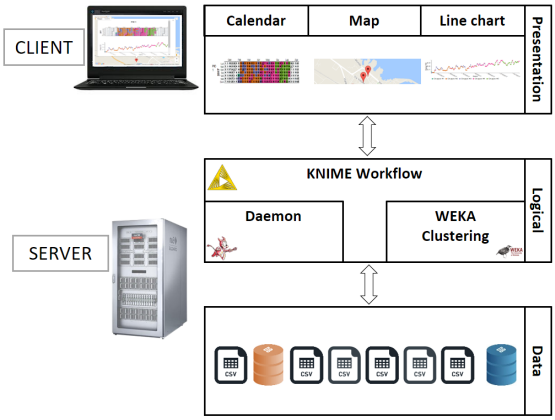

Fig. 3 The scheme of *MonitorApp* architecture.

$$\rho_{A,B} = \frac{cov(A,B)}{\sigma_A \sigma_B}, -1 \leq \rho_{A,B} \leq +1 \tag{2}$$

To inquire the relationships between two variables perhaps correlated (for example E and one of some external variables like T), the coefficient calculated as in (3) can be used as a feature.

$$r_e = \frac{\sum_{i=1}^{n}(T_i - \bar{T})(E_i - \bar{E})}{\sqrt{\sum_{i=1}^{n}(T_i - \bar{T})^2}\sqrt{\sum_{i=1}^{n}(E_i - \bar{E})^2}}, \quad -1 \leq r_e \leq +1 \tag{3}$$

4 The Working Pipeline

The web application features a three-level pattern in a client-server architecture as shown in Fig. 3 where the top level is the Presentation, which manages interaction with users, communication and visualization as application client; it includes different views as components to visualize and present information allowing users to interact with the system performing operations and requesting results.

The middle level contains the application logic which deals with the processing, transforming and reorganizing of the data; this level is implemented on the server side with a daemon that uses the KNIME workflow tool and WEKA as clustering framework.

Finally, the bottom level is related to physical data: it applies structured and raw data management (.csv files) providing the protection necessary for the entire system since data are available in databases or the file system. There is no direct communication between the Presentation and the Data levels and all exchanges of information are mediated by the application logic.

The Views in the *Presentation* level (that can be mixed together) are:

- Calendar View: clustered results are visualized through a *calendar metaphor*. Days belonging to a cluster display a color according to a predefined scale, in this way results easily show similar days featuring patterns;
- Line Graph View: allows a further dimension of analysis since it:
  1. shows data as connected lines; it is useful in combination with the Calendar View adding an additional dimension of information;
  2. highlights *time trends* making easy to directly compare and overlap data, as requested by experts while exploring temporal;
  3. shows the evolution and behavior of a monitoring unit over time.
- Map View: adds the geographic dimension to the data with the location of monitoring station; by clicking on items on the map a pop-up window that contains further details appears.

4.1 KNIME Workflow Implementation

The proposed framework integrates heterogeneous tools to build the working pipeline. The process is summarized in the high-level Algorithm 1 which is divided into four branches where the first one (lines 1-8) loads the temporal data by the server, followed by the execution of the KNIME workflow nodes (lines 9-12) and by the clustering functions (lines 13-21); finally the fourth phase (lines 22-29) shows results to users through the multiple Views.

The KNIME workflow developed for this study is depicted in Fig. 4 and, as Algorithm 1, it is composed by four macro-parts (letters A-D): A (data folders and files fetching), B (data format conversion from plain text to structured data types), C (aggregation and cleaning of values) and D (k-means execution, color assignment to each cluster and output file creation). Basically the workflow must include a node for reading input (*File Reader*), a node for clustering (*K-Means*) and a node for creating and writing structured JSON files (*Table to JSON* and *JSON Writer*) which will be the new input for the level dedicated to visualizations.

It is clear how, for a non-IT user, the tuning of the process through visual widgets is easier and more immediate than programming. The node used in the proposed case study are:

1. List Files: builds a list with the location of data folders and files contained in each of them so that only the files of interest are included;
2. Table Row To Variable Loop Start: for rows in the input table (folder and file names) allows data reading cycles and how many times to be executed;
3. File Reader: read data from heterogeneous sources and formats;
4. Loop End: a cycle iteration with intermediate results;
5. Row Splitter: allows pattern substrings splitting in columns the fields related to the observations;
6. String Replacer: replaces values when corresponding with peculiar patterns;
7. String To Date&Time\To Number: string types conversion useful for next steps;

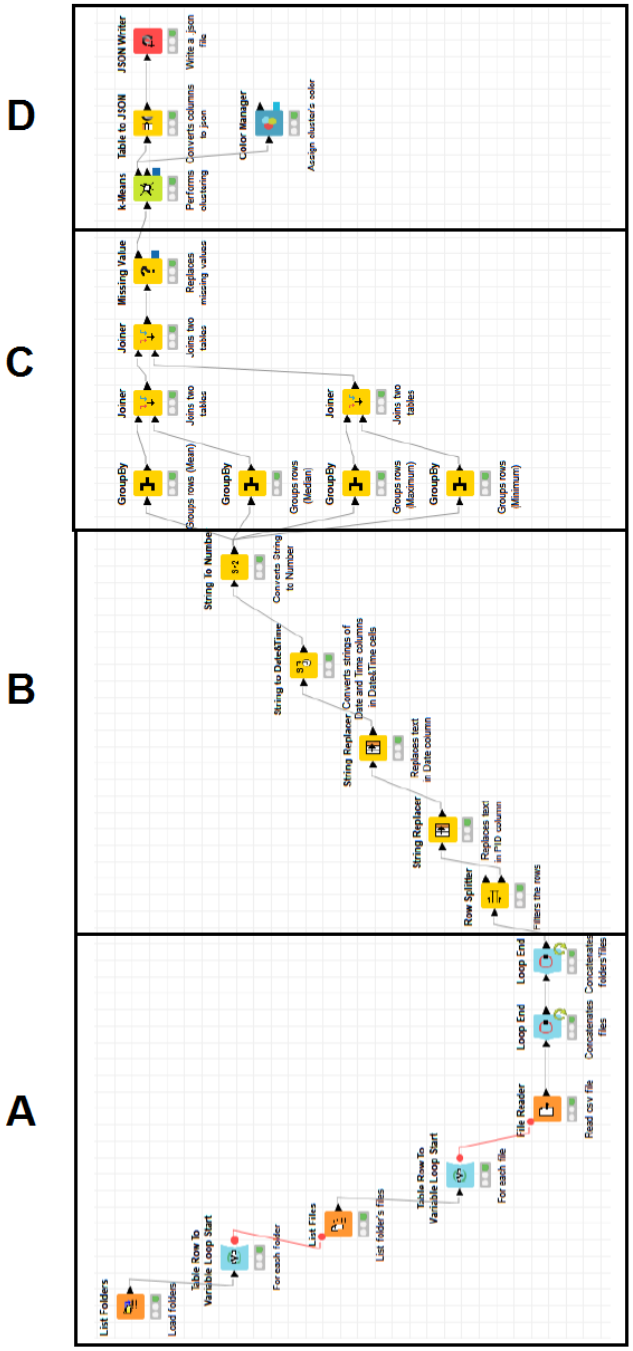

Fig. 4 The KNIME main workflows with the clustering of the different data sources (folders) linked to the monitoring stations and composed by four macro-parts (letters A-B) for the different stages, from data fetching to aggregation, clustering and output file.

Algorithm 1: The pseudo-code of the proposed working pipeline divided into four main phases.

```
Start the server and open a web socket;
if request then
    check the client PID;
    given N nodes n_i;
    select the time period T = {t_1...t_N};
    ∀n_i ∈ N if ∃J_PID JSON file then
        read the new PID data related to T;
    end
    run KNIME;
    read all PID data in T;
    start workflow W_PID = {n_1, .., n_N};
    while W_PID runs do
        ∀n_i ∈ W_PID, execute n_i;
        if ∃c|c is cluster node then
            run WEKA;
            select a clustering function f_K;
            execute f_K(c_1, ..., c_N), i < N;
        end
        filter and join nodes results {r_1, .., r_i}, i < N;
        final result r = ∨r_i;
    end
    if r ≠ ∅ then
        save r in new J_PID output file;
        given Y = {Calendar, Graph, Map, ..};
        visualize views V_m = f(J_PID, Y_m), m = 1..|Y|;
    else
        close safely all data sources;
        close the socket and report the error;
    end
end
```

8. Group By: works on table rows for selected columns according to a criterion; columns are grouped with respect to the *date* with additional criteria as average, median, maximum and minimum of attributes;
9. Joiner: combines tables in a similar way as relational database in order to aggregate data without repetitions;
10. Missing Value: handles missing values replacing them with default ones or using different rules;
11. K-Means: executes the WEKA algorithm setting parameters like *Distance* and k using the *Euclidean distance*;
12. Color Manager: selects a color palette assigned to cluster groups, in order to consistently display cluster colors in the Views;
13. Table to JSON: creates output strings following table column names;
14. JSON Writer: saves the output in a structured file;

With reference to the system architecture of Fig. 3, at the *Logical level* WEKA allows semi-expert users a *feature selection* facility to look for relationships between the clustering attributes, both at the single unit and at the globality of the monitoring stations.

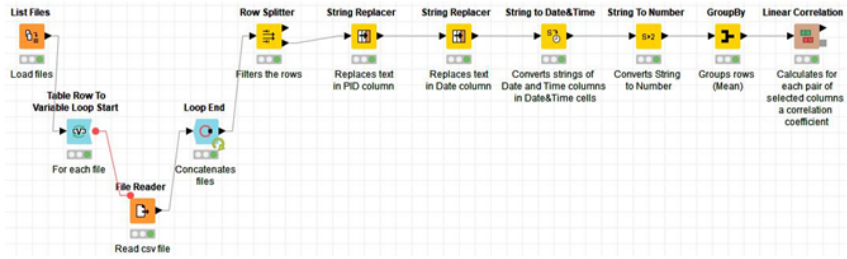

Fig. 5 The workflow implemented by KNIME at logical level to calculate and visualize the correlation matrix combining the clustering attributes of all the monitoring stations.

Table "Correlation values" - Rows: 4 | Spec - Columns: 4 | Properties | Flow Variables

Row ID	Mean(PID)	Mean(Temperature)	Mean(Humidity)	Mean(Pressure)
Mean(PID)	1	0.528	-0.261	0.691
Mean(Temper...	0.528	1	-0.747	0.415
Mean(Humidity)	-0.261	-0.747	1	-0.358
Mean(Pressure)	0.691	0.415	-0.358	1

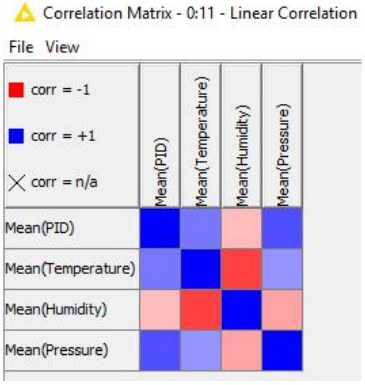

Fig. 6 The correlation matrix for all attributes of all the monitoring station.

For example, to determine if there is a relationship between the averaged values of the monitoring station (PID) parameters (Temperature, Humidity and Pressure), the data of each monitoring station were analyzed using the workflow in Fig. 5. The workflow is divided into three parts, where the first one relates to data fetching and tabulation, the middle part (yellow nodes) filters and organizes values while the final node calculates the linear correlation coefficients summarized in matrix form (Fig. 6).

By observing the colors in Fig. 6 where the correlation degree increases from the blue to the red color palette, it emerges the relationship between *Temperature* and *Humidity*: these two parameters are inversely correlated, meaning that as the value of one of the two increases the other decreases (as the temperature and pressure increase, the PID value also increases while, as humidity increases, the temperature decreases).

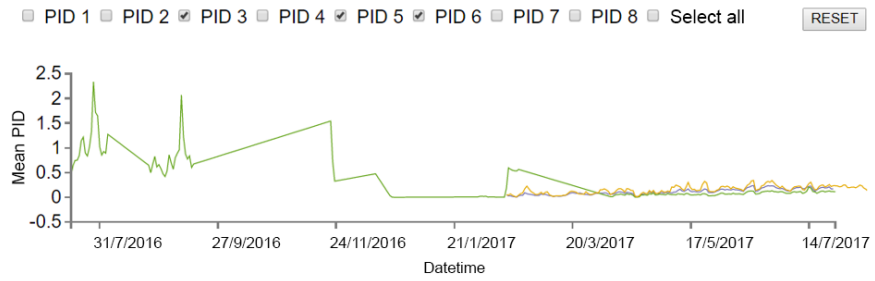

Fig. 7 Line Graph view permits to easily compare time trends.

The correlation matrix reveals that there are correlations between:

- Humidity and Temperature (strong negative correlation)
- PID and temperature (moderate positive correlation)
- PID and Pressure (moderate positive correlation)

4.2 The Views

Various state-of-art technologies and software libraries have been used to implement the proposed views such as: xHTML and CSS for presentation; Javascript for interaction; SVG (Scalable Vector Graphics) for graphics primitives; D3 (Data-Driven Documents) library for plot facilities and visual arrangements; Dart language for the app development.

4.2.1 Graph View

In the line *Graph View* (Fig. 7) the X axis represents the time dimension while the Y one represents the corresponding average of pollution detected from the PIDs each day; it visualizes the time trends and allows to compare the different monitoring stations.

Each value in these graphs is associated with the color of the cluster to which it belongs. The choice to provide the domain experts of line graphs is due to the fact that it is very common they look at data using line graph representations. In this way several linear graphs show:

- the average PID trend
- the average temperature trend
- the average course of humidity
- the average pressure trend

4.2.2 Enhanced Calendar View

As previously seen, Calendar View has been introduced in [13] to analyze behavioural patterns of a company employees. The original technique consists

Fig. 8 The Calendar view of the clustered air data.

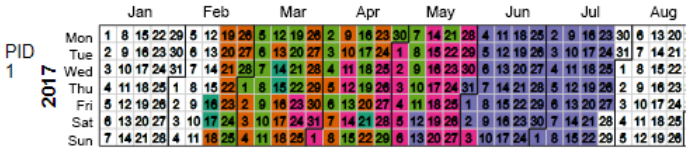

Fig. 9 The Calendar view of a clustering without the data dimension provided by the attribute 'Humidity'.

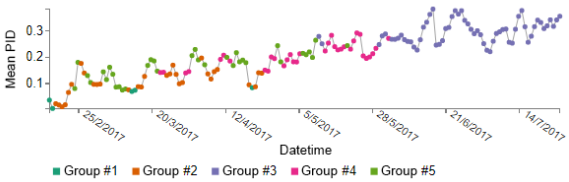

Fig. 10 The Line Graph View showing the data clustering without the 'Humidity' attribute.

in an interactive calendar-based scheme that shows the result of time series clustering applied to data related to employees of a research center. This technique easily reveals patterns of the typical working day of the research center and special days, like week-ends, holidays periods and celebration days; in this work it is integrated into a tool concerning the atmospheric air quality: patterns are shown in the chart and clusters in the calendars while colors indicate the matches between clusters and patterns.

The Calendar View in Fig. 8 shows the clustering related to a specified temporal period of all monitoring stations data using the WEKA library included into the previously seen KNIME workflow: with the K-means algorithm, each day (from 16 February to 27 July) is colored according to the cluster built by the average of attributes values.

In each month of the calendar, a main colored cluster appears, highlighting how the air pollution is localized; from the corresponding Line graph view it is visible that the pollution trend increases fairly linearly with some local peaks: the *green* and *violet* clusters are associated with high pollution periods from May to July even if the first one seems more compact.

By the end of June the green cluster appears stable except for some violet peaks while a similar behavior is followed by the blue cluster for the period from middle of February to April.

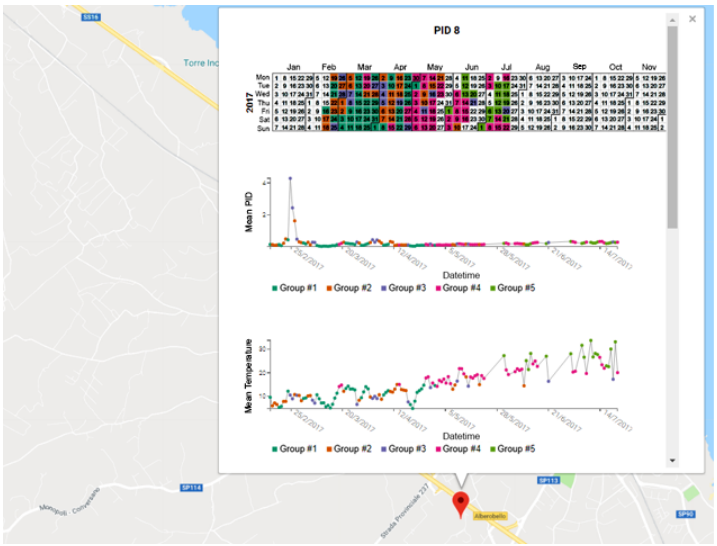

Fig. 11 The Map view displays together two other views adding a further data dimension.

The utility of the developed visualizations has been assessed by performing informal evaluations with end-users, which are chemists working on air quality monitoring. Taking into account Fig. 8 the analyst detected groups of colors belonging to clusters that feature distinctive data dimensions and small pieces of them or single days with differences from the larger groups resulted evident.

Removing a dimension of information, as depicted in Fig. 9, helps to visually inquire which are the features that mainly influences the clustering. Fig. 10 reveals the utility of the proposed tool, since removing the 'humidity' from cluster attributes leads to a very different color distribution from the previous one in June and July; it highlights that humidity is the main cause for the pollution increase in that period while is almost irrelevant for the previous.

4.2.3 Map View

It is also possible to see the clustering of specific monitoring stations in the Map view (Fig. 11) that adds the spatial dimension to the air data. This visualizations melts the Calendar View with the Line graph View.

In the bottom part there is the Line Graph related to the *Temperature (average)* trend, which was one of the clustering attributes (it is possible to add in the same panel the Graph views of all the other attributes). If the analyst is interested to see further details about the data trend in a single day of the selected time interval, by clicking on a Calendar day it will appear superimposed the related linear Graph view as in Fig. 12.

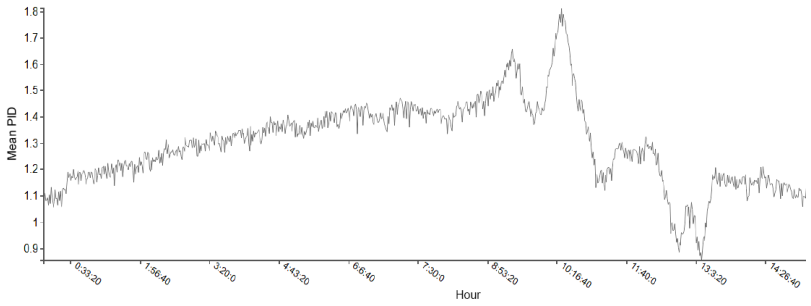

Fig. 12 The details-on-demand linear graph showing attribute trends during the hours of the day selected

.

5 Conclusions and Future Work

The result of this work is an integrated framework, easily customizable, that permits to visualize and compare specific atmospheric data gathered by monitoring stations with the aim to measure and control the environmental air quality. With respect to [7], the KNIME workflow now includes a better preprocessing phase that permits multiple source clustering, the geographic map feature with multiple visualizations.

The improvement in the analysis of the entire data collected by monitoring stations allows homogeneous clusters by comparing same periods of different years arranging together automatically different data flows. The correlation matrix allows users to visualize correlation in order to evaluate the most promising attributes to choose (or to remove) from the WEKA clustering parameters. The user interface is extensible and easy to use by domain experts who can experiment and replicate the analyses and customize the workflow pipeline.

In future work, since visualizations depend from clustering results, improvements can consider refining the algorithm and compose different machine learning methods, in order to see and choose multiple optimized clusterings, each with different parameters and options. An idea is hierarchical clustering that allows to show clusters at different levels in the form of dendograms. A regression analysis feature can be offered to predict faults and problems of the hardware devices. Another useful feature in the user interface could be the tuning of KNIME parameter values from a dashboard, also useful for validation purposes. Finally, a version of the application customized for mobile users could be implemented to alert users of emergency situations or allow them to analyze data in mobility.

References

1. "Air pollution in world: Real-time air quality index visual map,," 2018, https://www.eea.europa.eu/themes/air/intro/, Last accessed on 2019-04-02.
2. S. Ghazi, J. Dugdale, and T. Khadir, "Modelling air pollution crises using multi-agent simulation," in *2016 49th Hawaii International Conference on System Sciences (HICSS)*, Jan 2016, pp. 172–177.
3. B. Rajesh, A. Agarwal, and K. A. Saravanan, "Proficient modus operandi for scrutinize air pollution using wireless sensor network," in *2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014]*, March 2014, pp. 1312–1316.
4. "Italian public administration open data,," 2017, https://www.dati.gov.it/, Last accessed on 2018-10-20.
5. "Air pollution in world: Real-time air quality index visual map,," 2017, http://aqicn.org/map/world/, Last accessed on 2018-10-30.
6. P. Buono and M. F. Costabile, "Insights on the development of visual tools for analysis of pollution data," in *Distributed Multimedia Systems (DMS)*. Skokie, IL 60076, USA: Knowledge Systems Institute, 2012, Conference Proceedings, pp. 54–59.
7. P. Buono and F. Balducci, "A web app for visualizing electronic nose data," in *2018 22nd International Conference Information Visualisation (IV)*, July 2018, pp. 198–203.
8. S. Kandel, J. Heer, C. Plaisant, J. Kennedy, F. Van Ham, N. Riche, C. Weaver, B. Lee, D. Brodbeck, and P. Buono, "Research directions in data wrangling: Visualizations and transformations for usable and credible data," *Information Visualization*, vol. 10, no. 4, pp. 271–288, 2011.
9. D. A. Keim, "Designing pixel-oriented visualization techniques: Theory and applications," *IEEE Transactions on visualization and computer graphics*, vol. 6, no. 1, pp. 59–78, 2000.
10. W. Aigner, S. Miksch, H. Schumann, and C. Tominski, *Visualization of time-oriented data*. Springer Science & Business Media, 2011.
11. D. Ceneda, W. Aigner, M. Bögl, T. Gschwandtner, and S. Miksch, "Guiding the visualization of time-oriented data."
12. D. Ceneda, T. Gschwandtner, S. Miksch, and C. Tominski, "Guided visual exploration of cyclical patterns in time-series."
13. J. J. V. Wijk and E. R. V. Selow, "Cluster and calendar based visualization of time series data," in *Information Visualization, 1999. (Info Vis '99) Proceedings. 1999 IEEE Symposium on*, 1999, pp. 4–9, 140.
14. A. Malik, R. Maciejewski, T. F. Collins, and D. S. Ebert, "Visual analytics law enforcement toolkit," in *HST 2010 - IEEE Int. Conference on Technologies for Homeland Security*. IEEE, 2010, pp. 222–228.
15. A. M. Razip, A. Malik, S. Afzal, M. Potrawski, R. Maciejewski, Y. Jang, N. Elmqvist, and D. S. Ebert, "A mobile visual analytics approach for law enforcement situation awareness," in *IEEE PacificVis 2014 - Visualization Symposium*. IEEE, 2014, pp. 169–176.
16. M. Zhou, R. Wang, S. Mai, and J. Tian, "Spatial and temporal patterns of air quality in the three economic zones of china," *Journal of Maps*, vol. 12, no. sup1, pp. 156–162, 2016.
17. H. Li, H. Fan, and F. Mao, "A visualization approach to air pollution data explorationa case study of air quality index (pm2. 5) in beijing, china," *Atmosphere*, vol. 7, no. 3, p. 35, 2016.
18. A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.
19. S. Makridakis and M. Hibon, "Arma models and the box-jenkins methodology," vol. 16, pp. 147 – 163, 05 1997.
20. G. Sharma, G. Shroff, A. Pandey, B. Singh, G. Sehgal, K. Paneri, and P. Agarwal, "Multi-sensor visual analytics supported by machine-learning models," in *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on*. IEEE, 2015, pp. 668–674.

21. G. Bontempi, S. B. Taieb, and Y.-A. Le Borgne, "Machine learning strategies for time series forecasting," in *European Business Intelligence Summer School*. Springer, 2012, pp. 62–77.
22. C. P. Kappe, M. Bottinger, and H. Leitte, "Exploring variability within ensembles of decadal climate predictions," *IEEE Transactions on Visualization & Computer Graphics*, no. 1, pp. 1–1, 2018.
23. C. Kappe, M. Böttinger, and H. Leitte, "Visual exploration of ensemble variability at the example of decadal climate predictions," in *EGU General Assembly Conference Abstracts*, ser. EGU General Assembly Conference Abstracts, vol. 20, Apr. 2018, p. 10206.
24. J. Zhu, Y. Hu, C. Qin, and L. Z. Yin, "Simulation analysis of air pollution dispersion based on interactive virtual geographic environment," in *IET International Conference on Information Science and Control Engineering 2012 (ICISCE 2012)*, Dec 2012, pp. 1–4.
25. S. Wang, C. Zhang, Y. Huang, and W. Li, "Volume rendering and clipping for air pollution visualization," in *2010 2nd International Conference on Information Engineering and Computer Science*, Dec 2010, pp. 1–4.
26. M. Kern, T. Hewson, A. Schfler, R. Westermann, and M. Rautenhaus, "Interactive 3d visual analysis of atmospheric fronts," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2018.
27. A. Sarikaya, M. Correll, L. Bartram, M. Tory, and D. Fisher, "What do we talk about when we talk about dashboards?" *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2018.
28. G. Ellis and F. Mansmann, "Mastering the information age solving problems with visual analytics," in *Eurographics*, vol. 2, 2010, p. 5.
29. M. B. de Carvalho, B. S. Meiguins, and J. M. de Morais, "Temporal data visualization technique based on treemap," in *Information Visualisation (IV), 2016 20th International Conference*. IEEE, 2016, pp. 399–403.
30. T. C. Pearce, S. S. Schiffman, H. T. Nagle, and J. W. Gardner, *Handbook of machine olfaction: electronic nose technology*. John Wiley & Sons, 2006.
31. S. M. Scott, D. James, and Z. Ali, "Data analysis for electronic nose systems," *Microchimica Acta*, vol. 156, no. 3-4, pp. 183–207, 2006.