

AirMap: A Reactive Map of Air Quality

Dessislava Petrova-Antonova, Andrey Popradanov

Faculty of Mathematics and Informatics

Sofia University “St. Kliment Ohridski”

Sofia, Bulgaria

emails: d.petrova@fmi.uni-sofia.bg, andrey68a6@mail.bg

Abstract—Today, air pollution is one of the biggest environment risks to health that continues to rise and affects both people’s quality of life and economies. It is identified as a global health priority by the World Health Organization, since it affects all societies around the world. A huge amount of air quality data is collected by different means, but it is not easily accessible by the citizens, who have a sensible role to mitigate the problem. In this paper, a visualization tool of multi-scaling, spatial-temporal air quality data, called AirMap, is proposed. An adaptive development method, based on modern MERN (MongoDB, Express, React, Node.js) application stack, is applied to integrate line charts with map-based localizations, trends and time views. The interactive visualization helps citizens to better understand air quality data and to compare measured air quality across places. The feasibility of AirMap is proved using a large multi-dimensional data set including measurement for 7 air pollutants and 4 weather parameters.

Keywords— document database; time series data; spatial-temporal; data visualization; MERN stack.

I. INTRODUCTION

Air pollution damages the ecosystem and has a significant impact on the health of the population. The vegetation, fauna, water and soil, together with the services they support are directly affected. According to the air quality report [1] of the European Environment Agency (EEA), about 61% of the European ecosystem area remained exposed to air pollution levels exceeding limit values in urban areas and long-term exposure of particles less than 2.5 micrometers in diameter (PM_{2.5}) were responsible for 422 000 premature deaths in 2015. By cutting lives short, increasing medical costs and reducing productivity during working days, air pollution causes losses across the economy.

Air monitoring stations and remote sensing satellites are the main sources of air quality data. The existence of scalable and high-performance infrastructures and the rapid growth of data, leading to the so called “Big Data” phenomenon, open new opportunities for the development of data-driven applications in all domains of everyday life. The currently available air quality data sets have different levels of granularity, depending on the time basis and number of air parameters measured. They include the values of different air quality parameters, the location of monitoring stations and the time of recording. Information for the weather-related parameters, such as humidity and temperature, is also recorded. Such data sets are suitable for spatial-temporal visualization, allowing better understanding of air quality through interactive data explorations and presentation of abnormalities, trends and correlations.

The diversity of air quality data and the specifics of time series data itself, represent challenges for visualization.

Usually, it is impossible to show all individual data points due to limitation of the display capabilities to a certain plot resolution on the screen and also due to the processing capabilities of the underlying hardware [2]. The transferring of data from back-end datastore to the front-end user interface burdens the performance of visualization [3][4]. In this paper, a visualization tool of multi-scaling, spatial-temporal air quality data, called AirMap, is proposed. AirMap allows the user to select an air quality monitoring station on the map to show line charts of 7 air pollutants (PM, O₃, NO₂, NO, SO₂, CO, C₆H₆) and 4 weather parameters (Humidity, Atmospheric Pressure, Wind and Solar Radiation). A separate chart giving insight on the tendency of a given air pollutant is provided. The air quality data can be visualized for a given period of time defined with start date and end date as well as for a fixed time slot of 1 month, 3 months, 6 months or 1 year. The usability and performance of the AirMap is validated using an air quality data set provided on the Bulgarian open data portal [17].

The main contributions of the paper are as follows:

- An adaptive development method including a multi-scale time series data visualization and interaction;
- AirMap, a lightweight, open-source visualization tool that enables intuitive and fast exploration of large air quality data sets.

The rest of the paper is organized as follows. Section 2 provides background information about the air quality monitoring stations, air quality and weather parameters measured on the territory of Bulgaria. Section 3 presents a comparative analysis of current spatial-temporal air visualization solutions. Section 4 describes the AirMap architecture, while Section 5 its implementation. Section 6 shows the results from AirMap testing. Finally, Section 7 concludes the paper and gives directions for future work.

II. BACKGROUND

According to the World Health Organization, in 2012 Bulgaria had the second highest air pollution related mortality in Europe [5]. Air pollution causes cardiovascular, respiratory, endocrine and oncology diseases. The air of Sofia depends on climate and micro-climate conditions, the quantity and type of emissions and the location of emissions’ sources. The closed nature of Sofia valley is a prerequisite for the retention of air masses, frequent occurrence of temperature inversions and fog occurrence. The natural dispersion of emissions is poor, leading to their remaining in the air. That is why the use case scenario to validate the proposed platform is based on the air quality data of Sofia Municipality.

The Executive Environment Agency (ExEA) of Bulgaria regulates eight monitoring stations of air quality on the territory of Sofia Municipality – Orlov most, Gara Yana, Druzhba, Nadezhda, Hipodruma (Krasno selo), Pavlovo, Kopitoto and Mladost (see Table I).

TABLE I. AIR QUALITY MONITORING STATIONS

Station	Station code				
	Latitude	Longitude	EoI station code	Station Area	Station Type
Orlov most	42.690353	23.33605	BG0054A	U	T
Gara Yana	42.727489	23.556006	BG0024A	S	B
Druzhba	42.666508	23.400164	BG 0052A	U	B
Hipodruma	42.680558	23.296786	BG0050A	U	B
Pavlovo	42.669797	23.268403	BG0073A	U	T
Kopitoto	42.637192	23.243864	BG0070A	R-N-C	B
Mladost	42.655488	23.383271	BG0079A	U	T
Nadezhda	42.732292	23.310972	BG0040A	U	B

Table II shows the air parameters, whose values are collected through the air quality measurement stations.

TABLE II. AIR QUALITY DATASET PARAMETERS

Parameter	Metric
PM (Particulate Matter)	$\mu\text{g}/\text{m}^3$
Sources: The main sources of PMs are industry, transport, energy and domestic heating. During the local heating season, the main source of PMs is the burning of solid and liquid fuels in the household. This is due to low chimneys and specific weather conditions during the winter season, which reduces the possibility of dispersing atmospheric pollutants.	
NO₂ (Nitrogen dioxide)	$\mu\text{g}/\text{m}^3$
Sources: High-temperature combustion processes (Energy, Internal combustion engines).	
NO (Nitrogen oxide)	$\mu\text{g}/\text{m}^3$
Sources: High-temperature combustion processes (Energy, Internal combustion engines).	
C₆H₆ (Benzene)	$\mu\text{g}/\text{m}^3$
Sources: Benzene is mainly used as a raw material in the chemical industry. It is thrown away by emissions from motor vehicles and evaporation when working with oil (petrol stations and refineries).	
CO (Carbon oxide)	$\mu\text{g}/\text{m}^3$
Sources: The largest source is the road transport – over 65% of the total emitted quantity, heat plants, industry and others.	
O₃ (Ozone)	$\mu\text{g}/\text{m}^3$
Sources: Ozone is not emitted directly into the atmosphere. It is formed by the interaction of nitrogen oxides and volatile organic compounds under the influence of high temperatures and sunlight.	
SO₂ (Sulfur dioxide)	$\mu\text{g}/\text{m}^3$
Sources: Industry, energy, coal-fired heating with a lot of sulfur.	
Humidity	%
Atmospheric Pressure	mbar
Wind	m/s
Solar radiation	W/m ²

The EEA classifies the air quality stations according to the following criteria:

- Type of station – traffic (T), industrial (I), background (B);
- Type of zone – urban (U), suburban (S), rural (R), rural near city (R-N-C), rural regional (R-R), rural remote (R-R);

- Measurement regime (continuous, periodic).

The measurement regime of all stations is continuous data collection. They are part of the national air network.

III. RELATED WORK

This section presents a comparative analysis of the currently available applications for visualization of air quality data. The comparison is based on the most important characteristics, as follows:

- Usage of Open Data (OD);
- Geo-Location (GL) of air quality monitoring stations;
- Number of Pollutants (NP) that are visualized;
- Visualization if Referent Values (RV) of air parameters;
- Visualization of Weather Parameters (WP) such as Temperature (T), Humidity (H), Wind (W), Pressure (P);
- Multi-Scaling (MS), meaning that the time series data can be displayed at different levels of detail depending on the selected time slot and air quality parameters;
- Forecasting (F).

Table III presents the results from the comparative analysis. The last row of the table shows the proposed solution.

TABLE III. RESULTS FROM COMPARATIVE ANALYSIS

Solution	OD	GL	NP	RV	WP	MS	F
Luftdaten.info	yes	yes	PM10, PM2.5	yes	no	no	no
AirVisual Earth	no	yes	PM10, PM2.5	yes	W	no	no
AirVisualMap	no	yes	PM2.5, AQI	yes	T, H, W, P	no	yes
World AQI project	yes	yes	PM2.5, O ₃ , NO ₂ , CO, AQI	yes	T, H, W, P	no	yes
BreezoMeter	no	yes	PM10, PM2.5, O ₃ , NO ₂ , CO, SO ₂ , AQI	yes	no	partial	yes
European Air Quality Index	yes	yes	PM10, PM2.5, O ₃ , NO ₂ , SO ₂ , AQI	yes	no	partial	no
Air Quality in Scotland	yes	yes	PM10, PM2.5, O ₃ , NO ₂ , CO, SO ₂ , AQI	yes	T, W	yes	yes
Berkeley Earth	yes	yes	PM2.5, AQI	yes	no	yes	no
Plume Labs AIR map	yes	yes	PM2.5, PM10, NO ₂ , O ₃	no	no	yes	no
Hungarian Air Quality Network	no	yes	PM10, NO ₂ , NO _x , NO, SO ₂	yes	no	yes	no
Nowcast	no	yes	PM10, PM2.5, O ₃ , NO ₂	yes	no	no	no
Air Quality in Cyprus	no	yes	PM10, PM2.5, O ₃ , NO ₂ , NO, NO _x , SO ₂ , C ₆ H ₆	yes	no	yes	no
AirMap	yes	yes	PM, O ₃ , NO ₂ , NO, NO _x , SO ₂ , C ₆ H ₆	yes	T, H, W, P	yes	no

OK Lab Stuttgart is dedicated to the measurement of PMs within the Citizen Science project luftdaten.info. Citizens around the world install self-built sensors outside

their home. Luftdaten.info visualizes the transmitted data on a continuously updated map [6]. AirVisual Earth covers pollution airflows across the world by combining PMs data from public government air quality monitoring stations, satellite data and community [7]. The visualization on the map shows that the larger cities are both a cause and a victim of air pollution, depending on the wind direction and speed. AirVisual Map shows an interactive heatmap of PMs and Air Quality Index (AQI) together with the wind direction [8]. The user is able to select an air quality monitoring station, to view details about the values of pollutant, the AQI and the environment parameters as well as an air quality and weather hourly forecast. The World Air Quality Index project is a non-profit project showing in real-time the air quality data for more than 60 countries [9]. The air quality map of BreezoMeter helps citizens to understand the concentration levels and health effects of six air pollutants and to make informed decisions [10]. The user is provided with actionable health recommendations regarding indoor and outdoor staying, health sensitivities, sport activities and the risk for children. Additionally, the concentration level of pollen in the air and 24-hours history data of AQI are given. European Air Quality Index is an online service of the EEA and the European Commission. An interactive map can be zoomed in or searched for any city or region in Europe to check the overall air quality and concentration levels of key pollutants. An overall rating for each air quality monitoring station is marked by a colored dot on the map, corresponding to the worst rating for any of the five pollutants. The air quality situation at each of more than 2000 monitoring stations is presented in a short-term. Ricardo Energy & Environment provides mapped concentrations of pollutants for Scotland [11]. The data visualization supports multi-scaling based on a selected time period and air parameter. Time series data can be zoomed into the graph by clicking and dragging inside the graph area. All graphs can be exported in various formats. Berkeley Earth is an effort started in April 2014 to collect and analyze the world's air pollution data [12]. A near real-time data of PMs is presented on a map. Health indicators and qualitative descriptions are included based on the US EPA's air quality index standard for 24-hour exposure. Plume Labs developed a world air map of air pollutants, which also provides health recommendations for citizens' activities [13]. The map visualizes an open data from public land-based monitoring stations, which is blended with satellite imagery for better accuracy. Current and historical air quality monitoring data is provided by the Hungarian Air Quality Monitoring Network at national level [14]. The network consists of two mayor parts: automatic monitoring stations with continuous measure of wide range of air pollutants in ambient air, and manual system with sampling points and consecutive laboratory analysis. The data is visualized on a map and graphs for individual air pollutants are available. The visualization of historical data is limited to a period of 366 days. Nowcast map shows the current air pollution in detail across London in comparison with the UK Government's Daily Air Quality Index [15]. The user is able to click on the map, drag a marker, or enter a postcode to see the pollutant concentrations at the selected location. Since measurements from monitoring stations are only able to report air quality at that particular place, Nowcast combines these measurements with a detailed model to show a prediction of what air quality is like across the whole area of London. Health advice of current location is also provided. The map visualization of air quality in Cyprus shows the

current concentrations of pollutants for each monitoring station [16]. Additionally, a graph visualization of a variety air pollutants is available for different periods of time. The pollutant concentration values displayed at each station are in real-time, and as such, are not validated.

Table III shows that only 2 solutions – AirVisualMap and World AQI project, provide visualization of all weather's parameters, namely temperature, humidity, wind and pressure. At the same time, the visualization of pollutants is limited to 1 for the AirVisualMap and 4 for the World AQI project. In addition, they do not support multi-scaling at different levels of detail. Therefore, none of the presented solutions provide a complete visualization of both air pollutants and weather's parameters with option for multi-scaling on selection of time slot and air quality parameters.

IV. AIRMAP ARCHITECTURE FRAMEWORK

The architecture framework of *AirMap* consists of three main layers, shown in Figure 1:

- Data Layer – provides data persistent mechanisms, which are responsible for storing and retrieving data that is requested by the server layer;
- Server Layer – handles requests coming from the client layer by applying processing rules and sending responses with requested content;
- Client Layer – provides user's interaction interface available for access via web and mobile browsers.

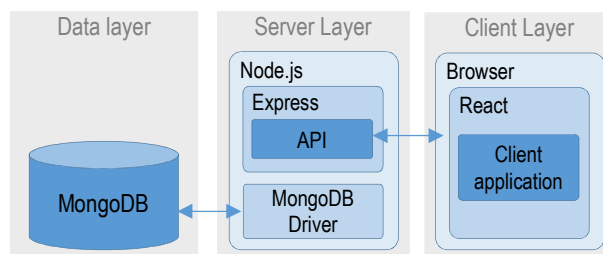


Figure 1. Architecture framework of AirMap

The architecture framework of *AirMap* follows a modern application stack, including MongoDB, Express, React, Node.js (*MERN* stack). *MongoDB* is an open-source document database that ensures persistence of data. It bridges the gap between the relational databases having a rich functionality and the key-value stores providing scalability and high performance. The JSON-like documents in *MongoDB* database allow the data structure to be changed over the time. The access and analysis of data are supported by ad-hoc queries, an indexing and a real time aggregation.

Node.js is a JavaScript runtime environment, which contains a built-in HTTP server library and thus allows for implementation of web servers. This provides a greater control over how the web server works. In the traditional web server's implementations, each request initiates a new thread, taking up system RAM and eventually maxing-out at the amount of RAM available. In contrast, *Node.js* operates on a single-thread, using non-blocking I/O invocations. Thus, it supports tens of thousands of concurrent connections held in the event loop. On completion of the requested task, the requestor is informed of the results via a callback. As a result, scalable network applications could be developed, capable to handling a huge number of simultaneous

connections with a high throughput. Historical images including maps of sources and concentrations of five pollutants, as well as trends over time are also supported.

Express is a Web application framework, which runs as a module within the *Node.js* environment. In the context of the current solution, it provides a REST API giving to the client layer access to *MongoDB* database using HTTP network calls.

React (referred also as ReactJS) is a JavaScript library for building reactive, interactive user interfaces. It breaks the front-end applications down into components. The components hold their own state. The state can be passed down from parent components to child components. The latter can pass changes back to the parent components using callbacks.

V. AIRMAP IMPLEMENTATION

This section describes the implementation of *AirMap* following the proposed architecture in Section IV.

A. Data Layer

The Air Quality data set is provided on the open data portal of Bulgaria [22]. It contains a continuous record of the air parameters displayed on the Sofia Municipality's website [18]. Data is not official, since it does not come directly from the ExEA – an administration within the Ministry of Environment and Water. It is obtained through a constant analysis of the graphics posted on the Sofia Municipality's website. It is therefore possible to deviate by up to 0.5% of the actual values.

The data set contains measurements from six air quality monitoring stations – Druzhiba, Nadezhda, Pavlovo (Krasno selo), Kopitoto, Mladost for all air parameters in Table II. The data acquisition has started at 16:00 on 20 January 2016. Each row shows the date and time, the station code, the pollutant code, its level, and whether it is marked as invalid. If the last metric is '1', then the EEA has noted that there is any dirt on the meter or any other reason that has led to an erroneous value. The Air Quality data set provides a time series data, which is imported in the *MongoDB* database as a single collection, called *airdata*. There are three main configurations that can be applied to the design of time series schema in *MongoDB*: (1) one document per data point; (2) bucketing the data using one document per time series time range; and (3) one document per fixed size [19]. The first configuration is used for the current implementation. Thus, each record in the data set corresponds to a separate JSON-like document in the *airdata* collection. This is the most comfortable modeling concept, because the data comes from a tabular schema (CSV format). The working with data is easy, since the document model maps to the objects directly in the application code.

The design of schema produces the largest number of documents and collection size per unit of time, but it is acceptable for the current application, since the size of data imported to the database is fixed, i.e., the data is not continuously ingested in the database.

B. Server Layer

A server application (*server.js*) is developed using *Express* and *Node.js*. When *Node.js* server initializes the event loop, it starts. Then, it processes the provided input script, which makes asynchronous API calls and proceeds

with the event loop. The event loop consists of several phases, which are executed in a given order as follows:

- timers – executes callbacks that are scheduled by timers;
- pending callbacks – executes I/O callbacks deferred to the next loop iteration;
- idle, prepare – for internal use only;
- poll – calculates how long it should block and poll for I/O, then processes events in the poll queue;
- check – executes callbacks immediately after the poll phase has completed. If the poll phase is idle, the event loop is able to proceed with the check phase rather than waiting;
- close callbacks – performs close on callbacks.

The *Node.js* server processes HTTP GET requests to retrieve data into the callback without blocking the main stack. The working process of *Node.js* is shown in Figure 2. In order to prevent server overload, the number of retrieved records is limited. The limit can be changed depending on the available hardware resources.

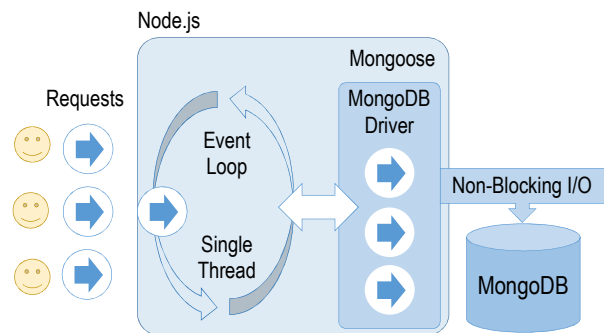


Figure 2. Node.js working process

C. Client Layer

The client application is built on *React* library and its wrapper *Highcharts*. It consists of small reusable components, which allow the User Interface (UI) to be divided into small pieces and to be built in a comprehensive way. The programming concept behind the *React* components is the virtual Document Object Model (DOM), meaning that the “virtual” representation of UI is kept in memory and synced with the “real” DOM. The virtual DOM is a lightweight copy of the real DOM, providing fast updating using a diffing algorithm. Two virtual DOMs are maintained at any given time by *React* – one that is related to the updated state of the data model and another one that is related to its previous state. The diffing algorithm compares both virtual DOMs to find the changes and to update the real DOM, as is shown in Figure 3. Its complexity is $O(n)$, since it adopts a heuristic approach with some assumptions to find the minimum number of modifications between two trees. The main benefit of this approach is that the changes don't need to be tracked and the expensive DOM operations are minimized. Thus, when the whole UI is re-rendered, the final result will include only the changes [20].

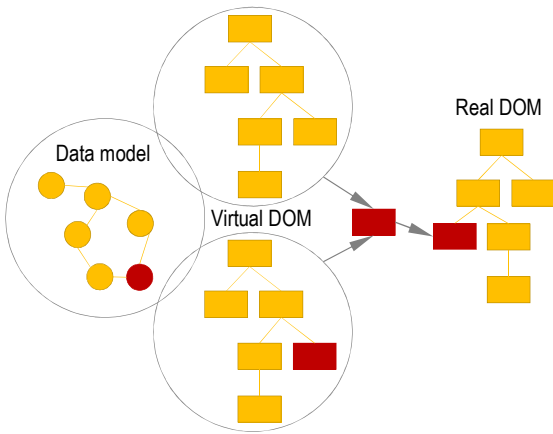


Figure 3. Virtual DOM and diffing algorithm, adapted from [20]

Three *React* components are implemented as follows:

- *Map* component – visualizes the air quality monitoring stations on a map;
- *Chart-Compare-Parameters* component – visualizes the air parameters;
- *ViewPort* component – contains the *Map* component and the *Chart-Compare-Parameters* component.

Figure 4 shows an UML diagram of the components.

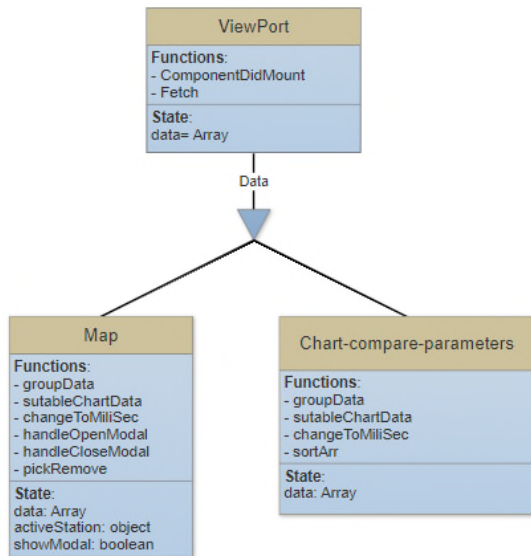


Figure 4. UML component diagram

The *Map* component and the *Chart-Compare-Parameters* component are nested in the *ViewPort* component. Thus, a more complex user interface structure is created, while keeping the simplicity of the application. The *Map* component uses *google-maps-react* module, which allows for integration of Google Maps service. The stations are localized based on data for latitude and longitude that is obtained from *MongoDB* database.

The *Chart-Compare-Parameters* component makes HTTP GET requests to the *Node.js* server application. Its lifecycle is shown in Figure 5, but it is applicable to all *React* components.

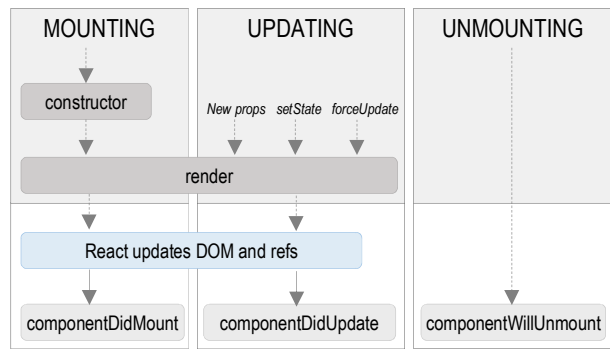


Figure 5. Lifecycle of React component

When an instance of the *Chart-Compare-Parameters* component is created and inserted into the DOM, the following methods are called: *constructor()*, *render()* and *componentDidMount()*. The *constructor()* method is called before the component is mounted. The *render()* method returns *React* elements that instruct *React* to render a DOM node. It does not modify the state of the component, meaning that it returns the same result each time it is invoked. Since *render()* method does not directly interact with the browser, the *componentDidMount()* method is called. It is invoked immediately after the component is inserted in the DOM tree. The *componentDidMount()* method instantiates the network request to the *Node.js* server in order to retrieve the air quality data. The *componentDidUpdate()* method is invoked immediately after updating occurs. Finally, the *componentWillUnmount()* method is invoked immediately before the component is unmounted and destroyed.

Time series data for each air quality monitoring station is separated into different *Highcharts* objects. Additional *Highcharts* objects are created for visualization of referent values of air parameters as well as for smooth representation of their tendency. The smooth representation separates long-trends and seasonal changes from the random fluctuations and gives a better picture of time series data. Here, two types of methods can be applied: (1) global methods that include fitting a regression over the whole time series; and (2) local methods, where a constraint is applied by a single parametric function. Since the global methods assume that the time series follow a single trend, which is not applicable to the air parameters, a local method is applied in the current implementation. Thus, the variation at the beginning of the time series does not affect measurements near their end. The moving average method (referred also as running mean) has been selected for implementation of the local smoother. It assigns an equal weight of each value in the data set and calculates arithmetic mean of a given set of values defined by a smoothing parameter. Let's define x_i as a value at the time i . Let $X = x_1, x_2, \dots, x_p$, where p is the length of the time series. The moving average method at the time t is calculated as follows:

$$MA_t^n = \frac{1}{n} \sum_{i=1}^n x_{t-i}, \quad (1)$$

where n is a smoothing parameter that determines the averaging window width [21].

The *pickRemove()* function takes as parameter an array of values to be processed and returns a new array with average values. First, it checks whether the input parameter is passed. Next, another function, called *groupAverage()*, is invoked,

which is responsible for actual implementation of the moving average method. The value of the smoothing parameter is 45.

VI. AIRMAP VALIDATION RESULTS

AirMap is tested using different browsers (Internet Explorer, Google Chrome, Opera and Mozilla FireFox) under Window 10 operation system. The hardware configuration includes processor Intel Core i3 and 8GB RAM.

The visualization of the air quality monitoring stations on the map is presented in Figure 6. A sample chart with the values of NO₂ for all quality monitoring stations is also shown. Additional information about the station such as type (background or traffic), station area (urban or rural) and EoI station code (as in AirBase of EEA) is visualized in a popup window when the user clicks on the station's location.

The following visualization options are provided:

- show data from one or more air quality monitoring stations;
- show data for a given period of time defined with start date and end date;
- show data for 1 month, 3 months, 6 months or 1 year;
- show or hide a separate chart giving an insight about the tendency of a given air parameter.

Table IV shows the results of performance testing of *AirMap*. 12 test cases are designed that cover visualization of different amounts of data depending on the selected options by the user. Additional 4 test cases are performed to analyze the behavior of the application under high data load. In the worst case, all data from *MongoDB* database is visualized (1,048,576 documents).

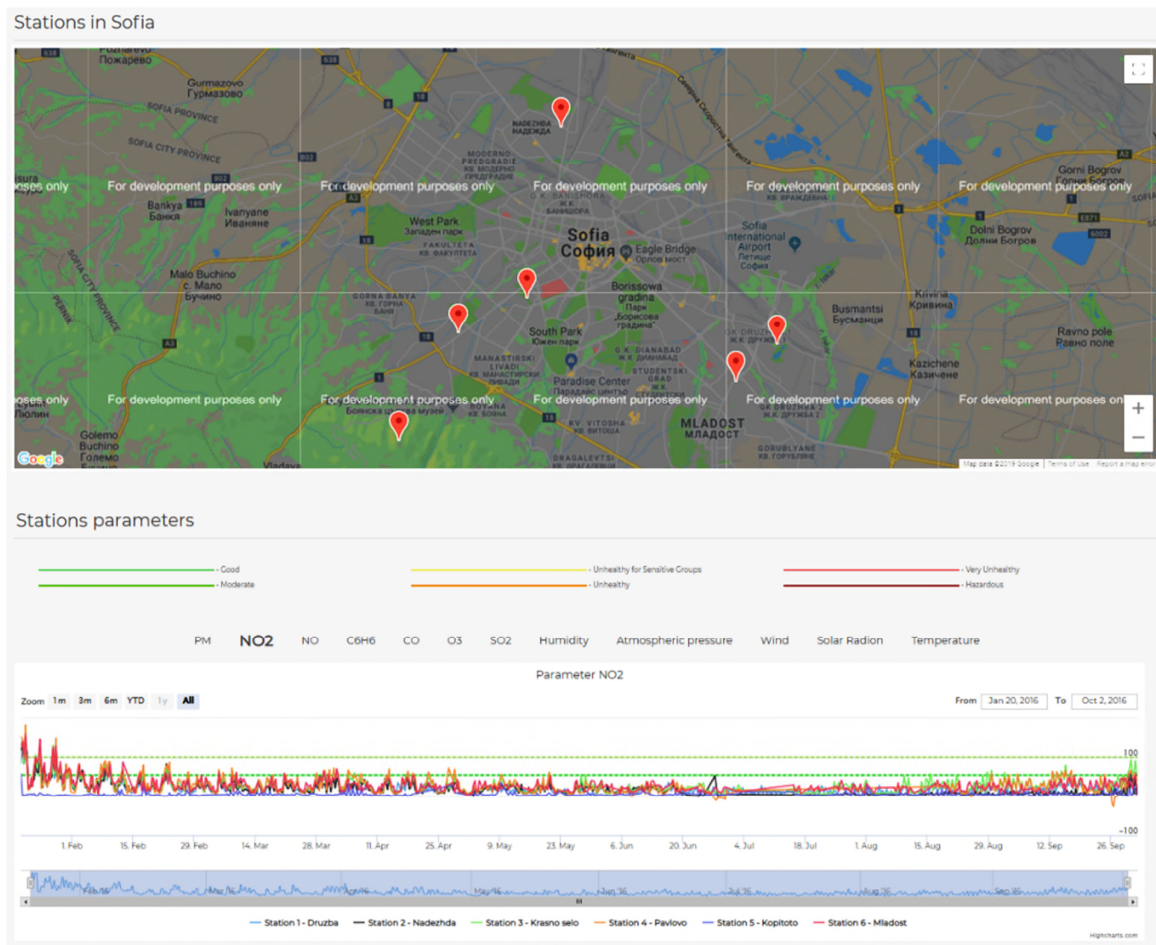


Figure 6. Values of NO₂ for all air monitoring stations

TABLE IV. AIRMAP PERFORMANCE TESTING

Chart Data	Loading time [s]			
	1m	3m	6m	2 Years
1 air parameter for 1 station	0.2	0.3	0.5	1.5
1 air parameter with a trend	0.2	0.3	0.5	1.5
1 air parameter for all stations	0.5	1	1.5	5
All air parameters for all stations	3	6	9	15

The results in Table IV prove that the visualization is performed in an efficient way. The content loading is fast due to the virtual rendering technique applied. If the number of retrieved records exceeds 900 000 (rendering of all data in the air quality data set), the response time of the application becomes high. This is expected given the hardware resources of the testing machine, as well as the data volume. That is why, the maximum data that the user is able to load is limited to all values of one parameter (2-year period) for all monitoring stations.

Usability testing has been used to evaluate user interaction with *AirMap*. It is done by real-life users – students, who have been attended the Software testing course of Software Technologies master program in the Faculty of Mathematics and Informatics at Sofia University. A total of 15 students have been participated in the testing process. They were asked to externalize thoughts and feelings when interacting with *AirMap*. Quantitative information such as time on tasks, success and failure rates, effort (number of clicks, perception of progress) was collected. In addition, qualitative information about stress responses, subjective satisfaction, perceived effort or difficulty was gathered. Non-critical errors that are recovered by the students and do not result in the students' ability to successfully complete the task were detected and fixed later. The outcomes from the testing session prove that *AirMap* provides a sufficient level of usability.

VII. CONCLUSIONS

This paper proposes a tool for multi-scaling, spatial-temporal visualization of air quality data, called *AirMap*. *AirMap* allows the user to select an air quality monitoring station on the map, to show line charts of a variety of air parameters and their trends for different slots of time. The usability and performance of the *AirMap* is validated using an air quality data set provided on the open data portal of Bulgaria. The results from testing show that *AirMap* provides a sufficient user experience due to the fast loading of data and the efficiency of user's interaction.

The future work includes development of functionality for automated import of air quality data to the *MongoDB* database. Additional direction will be implementation of an option for visualization of all air parameters for a single monitoring station.

ACKNOWLEDGMENT

This work was supported by the Bulgarian National Scientific Fund within the project no. DN 12/9 and by the European Commission under grant agreement no. 857155.

REFERENCES

- [1] EEA, Air quality in Europe - 2018 report, EEA Report No 13/2017; European Environment Agency: Copenhagen, Denmark, 2018.
- [2] J. Traub, N. Steenbergen, Ph. Grulich, T. Rabl, and V. Markl, "I²: Interactive Real-Time Visualization for Streaming Data," in Proceedings 20th International Conference on Extending Database Technology (EDBT), Venice, Italy, March 2017, pp. 526-529.
- [3] L. Battle, E. Chang, and M. Stonebraker. "Dynamic Reduction of Query Result Sets for Interactive Visualization," in IEEE International Conference on Big Data, October 2013, pp. 1-8.
- [4] D. Fisher, "Big data exploration requires collaboration between visualization and data infrastructures," in Proceedings of the Workshop on Human-In-the-Loop Data Analytics, San Francisco, California, June-July 2016, pp. 16.
- [5] WHO, World health statistics 2017: monitoring health for the SDGs, Sustainable Development Goals, 2017.
- [6] OK Lab Stuttgart, <http://deutschland.maps.luftdaten.info/> (accessed June 17, 2019)
- [7] AirVisual Earth, <https://www.airvisual.com/earth> (accessed June 17, 2019)
- [8] AirVisual Map, <https://www.airvisual.com/map> (accessed June 17, 2019)
- [9] World Air Quality Index project, <http://aqicn.org> (accessed June 17, 2019)
- [10] BreezoMeter, <https://breezometer.com/air-quality-map> (accessed June 17, 2019)
- [11] Ricardo Energy & Environment, <http://www.scottishairquality.scot/>, (accessed June 17, 2019)
- [12] Berkeley Earth, <http://berkeleyearth.org/air-quality-real-time-map/> (accessed June 17, 2019)
- [13] Plume Labs, <https://air.plumelabs.com/> accessed March 1, 2019)
- [14] Hungarian Air Quality Monitoring Network, <http://www.levegominoseg.hu/automatic-monitoring-network> (accessed June 17, 2019)
- [15] UK Government's Daily Air Quality Index, <https://www.londonair.org.uk/LondonAir/nowcast.aspx> (accessed June 17, 2019)
- [16] Map visualization of air quality in Cyprus, <http://www.airquality.dli.mlsi.gov.cy/>, (accessed June 17, 2019)
- [17] Bulgarian open data portal, <https://data.egov.bg/>, (accessed June 17, 2019)
- [18] Sofia Municipality air information, <https://www.sofia.bg/components-environment-air>, (accessed June 17, 2019)
- [19] MongoDB, Time Series Data and MongoDB: Best Practices Guide, 2019
- [20] T. Parviainen, "Change And Its Detection In JavaScript Frameworks," March 2015, <http://teropa.info/blog/2015/03/02/change-and-its-detection-in-javascript-frameworks.html>
- [21] A. Raudys, E. Malčius, and V. Lenčiauskas, "Moving Averages for Financial Data Smoothing," T. Skersys, R. Butleris, and R. Butkiene (Eds.): ICIST 2013, CCIS 403, Springer-Verlag Berlin Heidelberg, 2013, pp. 34-45.
- [22] Air Quality dataset, <https://data.egov.bg/data/resourceView/4084a7c9-3e9b-4bd2-a1e2-ca3d944a4416>, (accessed June 17, 2019)