Internet of Things Solution for Intelligent Air Pollution Prediction and Visualization

Conference Paper · July 2019

DOI: 10.1109/EUROCON.2019.8861609

Biljana Risteska Stojkoska

Ss. Cyril and Methodius University in Skopje

Kire Trivodaliev

Ss. Cyril and Methodius University in Skopje

Internet of Things Solution for Intelligent Air Pollution Prediction and Visualization

Mladen Korunoski, Biljana Risteska Stojkoska, Kire Trivodaliev
Ss. Cyril and Methodius University, Faculty of Computer Science and Engineering,
1000 Skopje, Macedonia
e-mail: korunoski.mladen@students.finki.ukim.mk, biljana.stojkoska@finki.ukim.mk, kire.trivodaliev@finki.ukim.mk

***Abstract*—Air pollution monitoring and control is becoming a key priority in urban areas due to its substantial effect on human morbidity and mortality. This paper presents a system architecture for intelligent pollution visualization and future pollution prediction by encompassing pollution measurements and meteorological parameters. First, a pollution model using spatial interpolation is built. By adding meteorological parameters this model is further used to identify the pollution field evolution and the position of potential sources of air pollution. Using deep learning techniques, the system provides predictions for future pollution levels as well as times to reaching alarming thresholds. The whole system is encompassed in a fast, easy to use web service and a client that visually renders the system responses. The system is built and tested on data for the city of Skopje. Although the spatial resolution of the system data is low, the results are satisfactory and promising. Since the system can be seamlessly deployed on an Internet of Things sensing architecture, the improved data spatial resolution will improve performance.**

***Index Terms*—Intelligent System, Air Pollution Monitoring, Air Pollution Prediction, Web System**

I. Introduction

Air pollution is one of the key problems of major urban areas in developing and industrial countries, especially where air quality measures are not available or minimally implemented or enforced [1]. It is estimated that around 91% of the world's population lives in places where air quality exceeds World Health Organization (WHO) guidelines, and around 4.2 million deaths every year can be directly attributed to exposure to ambient (outdoor) air pollution [2]. Chronic exposure to air pollution increases the risk of cardiovascular and respiratory mortality and morbidity, while acute short-term inhalation of pollutants can induce changes in lung function and the cardiovascular system exacerbating existing conditions such as ischemic heart disease [3][4]. Air pollution contributes to climate changes which in terms increases premature human mortality [5].

Air quality monitoring and control is an essential part of the concept of smart city which is becoming the standard to which both developing and developed countries aspire, thus the public mindfulness for the process is high. Many of the world governments deploy and operate stations for air quality monitoring and make the acquired data publicly available. In general these stations have very high-quality sensory devices that can record the state of a wide range of pollutants (like CO, NO_2, SO_2, O_3, PM - particulate matter, etc.). However, the high costs of installing and maintaining these sites limits their number - e.g. the wider Skopje area has 7 active monitoring sites, separated from each other by several kilometers. In such cases, the low spatial resolution is resolved by using mathematical models that estimate the concentrations of the pollutants over the complete geographical space of interest. Although these models are complex and incorporate various input parameters such as meteorological variables they can still be inaccurate (due to highly variable meteorological conditions [6]) which can lead to unsubstantiated inferences [7].

The aim of this research is to create an effective intelligent air pollution prediction and visualization system build on the top of Internet of Things. The system is to be composed of four subsystems that work together to enable its functionalities. A pollutant field building subsystem which is the core feeding its results to the other subsystems for: pollution source detection, pollution prediction, and time-to-event prediction. Taking into account the measured pollutant concentrations from all available monitoring sites, the first objective is to build a pollution model that gives insight of how the pollution field looks like. The second objective is to identify the sources of pollution as well as how will the pollutant field evolve in the future due to the wind field. In addition, the obtained models are to be presented to users in familiar quantities and in direct correlation of the impact pollution has on their health. The final objective is using deep learning techniques to predict the time when pollution surpasses the alarming thresholds. The whole system should provide a fast and easy way for users to obtain and display relevant data and an intuitive graphical user interface.

There are efforts made in overcoming the aforementioned problems. In order to achieve good coverage of Beijing and monitor the air quality in real time using automatic data processing the authors of [8] developed a prototype hierarchical system having Zig-Bee nodes in the endpoints gathering the pollution readings and routers on top of the nodes that collect the data, communicate between each other and transfer the sensing data to information processing center that performs the real-time observation of air quality. In [9] a Web-based solution is presented. Using Java servlet and JavaLite a framework for mining the air pollution data is developed, and further spatial mining is used to explore the effects of meteorological

978-1-5386-9301-8/19/$31.00 ©2019 IEEE

conditions. A new approach to calculating air pollution levels is proposed in [10]. The calculation is done through data fusion and neural networks, using the meteorological variables as a decision factor. In [11] the P-Sense system for air pollution and monitoring is introduced. The system uses participatory sensing to provide air pollution data in vast amounts and at different granularity levels of time and space and thus provide enough information for all concerned entities to be able to analyze their problems and requirements. The authors of [12] propose an air pollution and fog detection approach based on the data gathered by a vehicle equipped with sensors such as cameras and light detection and recognition instruments.

The rest of this paper is organized as follows. In the next section the system architecture of our solution is presented. The third section gives implementation details. In section four the obtained results for the prediction accuracy are given. The paper is concluded in section five.

II. System Architecture

The system is composed of four subsystems that work together and each subsystem is dedicated to different system objective. The overall architecture is given in Figure 1.

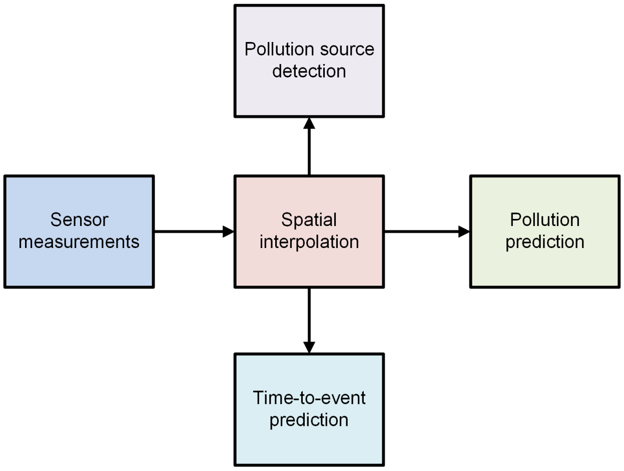

Fig. 1. System architecture

The sensor measurements feed the central subsystem, i.e. spatial interpolation subsystem. The measurements can be obtained from any sensor installed in the monitored area. Usually, in big cities, there are measuring stations installed on a few predefined locations. Apart from these sites, it is also possible for citizens to participate with their own measurement, a relatively new paradigm known as *Participatory Sensing*, or *Crowdsensing*. E.g., temperature measurements obtained from citizens smartphones, or pollution values obtained from home air purifiers. Therefore, all available measurements can be used as input data for this architecture. Apart from air pollution sensors, meteorological data are also needed, and they can be used from any available web service that provides such data. In our solution, we used only wind speed and wind direction, although the system can be expanded to include other meteorological parameters.

In the following subsection, the other subsystems of the architecture are explained in details, i.e. the mathematical models and algorithms used to enable the functionality of the system.

A. Spatial Interpolation

The first step in building the system is constructing an interpolated pollutant field for all of the observed pollutants. A well known method for spatial interpolation is used to generate the pollutant field based on the collected data [13]. First, for each pollutant, the monitoring sites' data is filtered. All pollutants are subjected to Kriging interpolation in order to produce the pollutant fields, which are sampled with 500m resolution. The Kriging process is accomplished in two steps. First, an average value field is constructed to estimate mean concentrations of the pollutants at each location using latitude and longitude as predictive variables. Next, at each location and hour, an anomaly field is created by applying simple Kriging [14] to the differences between the current observation and the mean value for that location. The concentration at each location and hour is estimated as the sum of the time-invariant average field and the time-varying anomaly field. This two-step process helps reduce the reconstruction errors associated with missing values. The interpolated pollutant field, $P(\boldsymbol{x}, t)$, is estimated in two parts:

$$P(\boldsymbol{x}, t) = S(\boldsymbol{x}, t) + A(\boldsymbol{x}, t) \tag{1}$$

where $S(\boldsymbol{x}, t)$, is the time-invariant average field, $A(\boldsymbol{x}, t)$ is the time-varying anomaly field, $\boldsymbol{x}$ is the location, and t is time.

The time-invariant average field is calculated using:

$$S(\boldsymbol{x}) = \Big(\sum_{n=1}^{N} K^{(n)}(\boldsymbol{x})\big(\bar{p}^{(n)} - G(\boldsymbol{x}^{(n)})\big)\Big) + G(\boldsymbol{x}) \tag{2}$$

where $K^{(n)}(\boldsymbol{x})$ are the Kriging coefficients, $\bar{p}^{(n)}$ is the estimated mean pollutant concentration, and $G(\boldsymbol{x})$ is the global auxiliary function.

On the other hand, the time-varying anomaly field is calculated by:

$$A(\boldsymbol{x}, t) = \sum_{n=1}^{N} K^{(n)}(\boldsymbol{x}, t)\big(p_t^{(n)} - \bar{p}^{(n)}\big) \tag{3}$$

where $K^{(n)}(\boldsymbol{x}, t)$ are the Kriging coefficients, $p_t^{(n)}$ is the pollutant concentration, $\bar{p}^{(n)}$ is the estimated mean pollutant concentration.

For determining the Kriging coefficients a spherical model is fitted to the variogram:

$$y(h) = \begin{cases} c_0 + c(\frac{3h}{2h} - \frac{1}{2}(\frac{h}{\alpha})^3), & 0 < h \leq \alpha \\ c_0 + c, & h > \alpha \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where h is a distance and α is the fitted parameter. The correlation function, R, that represents the correlation between the data from the monitoring sites depending on the distance between them, at zero distance is taken as one ($R(0) \approx 1$ for perfect data), even though the function diverges from unity due to instrument noise.

Also, for determining the parameters of the global auxiliary function in the stationary part,

$$G(\boldsymbol{x}) = \alpha\lambda(\boldsymbol{x}) + \beta\varphi(\boldsymbol{x}) + \delta\lambda(\boldsymbol{x})^2 + \varepsilon\varphi(\boldsymbol{x})^2 + \eta\lambda(\boldsymbol{x})\varphi(\boldsymbol{x}) + \kappa \tag{5}$$

where $\alpha, \beta, \delta, \varepsilon, \eta, \kappa$ are solved parameters, λ is the location's latitude, φ is the location's longitude, and $\boldsymbol{x}$ is the location. The Levenberg-Marquardt technique is used to solve the least-squares problem, which proved to be better than the polynomial regression algorithm. This function helps to minimize the average Kriging error over the field.

One interesting feature added is how the level of pollutant concentration is equivalent to the number of cigarettes smoked. For many people, comparing air pollution to cigarette smoking is more meaningful than the numbers of yearly deaths [15].

B. Sources Determination

The subsystem for pollution sources determination requires accurate weather model as an extra layer for determining the pollutant sources. Rather than adopting an existing weather model, a simpler framework that relies only on short-term changes in the air pollution field and a limited input of weather data is adopted. To estimate surface fluxes, short-term transport process and hour-by-hour changes in pollutant concentration are considered, using the interpolated pollutant fields constructed previously. Given a wind field, one can predict how the pollutant field will evolve according to the wind. By comparing the pollutant fields between two time points one can estimate the pollutant fluxes that must have occurred during the time interval due to the wind flow. The estimation is further improved by considering both forward and reverse evolution as well as an effective lifetime:

$$\begin{aligned} F(\boldsymbol{x}, t) = &\frac{P(\boldsymbol{x} + \vec{v}(\boldsymbol{x}, t)\Delta t, t + \Delta t) - e^{-\Delta t/\tau} P(\boldsymbol{x}, t)}{2\Delta t} + \\ &+ \frac{P(\boldsymbol{x}, t) - e^{-\Delta t/\tau} P(\boldsymbol{x} - \vec{v}(\boldsymbol{x}, t)\Delta t, t + \Delta t)}{2\Delta t} \end{aligned} \tag{6}$$

where $P(\boldsymbol{x}, t)$ is the pollutant field, $\vec{v}(\boldsymbol{x}, t)$ is the weather model, $\boldsymbol{x}$ is the location, t is time, Δt is time difference, and τ is the effective lifetime.

The use of both forward and reverse flow terms is desirable as it allows for the partial cancellation of errors resulting from inaccuracies in the pollution reconstruction and/or wind field.

After estimating the surface fluxes, the locations with largest flux values that correspond to the pollutant sources are identified. This noisy 2D surface is subjected to kernel smoothing using the Gausian kernel as a convolution function and the moving average values are calculated. Finally, the local maxima is determined by observing the neighboring values and comparing them with the current one, experimenting with the window size. Even thought this approach might look naïve, it proved itself very effective in this research.

C. Forecasting

The data at disposal are multivariate time-series and thus it is possible to frame a forecasting problem. The first set of variables are the measurements of different pollutant concentrations from several monitoring sites close to the are of interest. The second set are the wind field data (speed and direction) near the area of interest. Recurrent neural networks (RNNs) like Long Short-Term Memory (LSTM) are able to almost seamlessly model problems with multiple input variables [16]. When given the pollution data the system can forecast the pollution at the next couple of hours. However, the data is not framed as a supervising learning problem, but modified in a way that a monitoring site is chosen to be a query site and the pollutant concentration is predicted for the query site for the given hour having the pollutant concentrations and whether conditions in the prior time steps. Additionally, the series are made stationary with differencing and seasonal adjustment. Finally, the data is normalized.

In order to capture time or sequence dependent behaviour, two stacked layers of 75 LSTM cells are used, unrolled during training and prediction. The two stacked layers are used in the network in order to remember longer patterns in the data. The number of LSTM cells is chosen empirically in order to prevent overfitting [17]. Some architectures apply regular dropout to combination of layers as a regularization technique. In this architecture the recurrent dropout is used in order to further prevent overfitting and due to the fact of how LSTM cells are designed to capture dependencies [18]. The mean absolute error is used as objective function and Adam as optimizer since it has shown itself as a better optimizer for deep learning models then others [19].

D. Time-to-Event Prediction

The last functionality of the system is predicting the hours until the alarming thresholds are surpassed. Recent research in Survival Analysis, a study of expected duration of time until one or more events happen [20], resulted in a framework for time-to-event (TTE) prediction using deep learning, which takes into account recurrent events, time varying covariates, temporal patterns, sequences of varying length, learning with censored data and flexible predictions. Appropriate model for covering the first four points is RNN since the data used is time-series processed in discrete steps. The last points can be covered by appropriately choosing the objective function. Taking into account the TTE for each monitoring site at each timestamp - y_t^n for site $n = 1, ..., N$ at timestamp $t = 0, 1, ..., T_n$, measured pollutant concentrations and wind field data up to particular time - $x_{0:t}^n$, and indication of whether the datapoint is censored or not - u_t^n, one can maximize the objective function from survival analysis assuming that TTE at each step follows some distribution governed by parameters outputted from the model:

$$\begin{aligned} \sum_{n=1}^{N} \sum_{t=0}^{T_n} u_t^n \cdot log[Pr(Y_t^n = y_t^n | x_{0:t}^n)] + (1 - u_t^n) \cdot \\ \cdot log[Pr(Y_t^n > y_t^n | x_{0:t}^n)] \end{aligned} \tag{7}$$

where $Y \sim$ is the Weibull distribution with parameters α_t and β_t, and $\Theta_t = \begin{pmatrix} \alpha_t \\ \beta_t \end{pmatrix} = g(x_{0:t})$ is the output of the RNN.

This machine learning model uses special log-likelihood loss for censored data to predict the distribution over time to the next event. It is called WTTE-RNN because the Weibull distribution is used with parameters outputted from the RNN [21]. It might seem that training with censored data is not appropriate but it is observed that no information can be acquired about the parameter or the distribution of the TTE by knowing the censoring time. All one needs is an assumption that the distribution has some basic shape governed by few parameters. The Weibull distribution is an obvious choice for several reasons: has continuous and discrete variants, is expressive, has closed forms, is similar to the Normal distribution, has the weakest link property and has built in regularization mechanisms.

In the system single recurrent layer made of two gated recurrent units (GRU) is used, followed by a dense output layer of dimensionality 2 with custom activation layer, alpha and beta output values. The activation layer is a custom function set to an exponential function for alpha and a sigmoid function between 0 and max beta for the second parameter. The rationale is that alpha should always be greater or equal than 1 and beta should be between 0 (high confidence) and a maximum value that acts as both regularizer and stabilizer. The tuning parameters are: GRU with activation tanh and Adam optimizer with learning rate 0.1.

III. System Implementation

This section presents how the subsystems are integrated and how the client renders the responses. We use the city of Skopje as a case study because the pollution levels exceed alarming thresholds during certain periods of the year. Two sets of data sources (hourly data for the past five years) are used to showcase the functionalities of the system. The first source contains the measured pollutant concentrations obtained from the Ministry of Environmental and Physical Planing of the Republic of Macedonia [22], measured at several monitoring sites throughout the country, many of which are located in Skopje. The monitoring sites measure the concentration of different pollutants: carbon monoxide (CO), nitrogen dioxide (NO_2), ozone (O_3), particulate matter (PM_{10} and $PM_{2.5}$) and sulphur dioxide (SO_2) and not all sites measure the concentration of all of the pollutants. There are missing values in the data, due to site inactivity or faults in the measuring instruments. The second dataset is obtained from DarkSky, a web weather API [23], which provides data for many meteorological parameters, but for this research only the wind speed and direction are considered.

The system consists of a web service and a client that runs in a browser. The web service is developed with the Node.js run-time environment which utilizes faster programming language, C++ in our case, in order to speed-up the calculations needed for creating the pollutant fields as well as the source determination. Node's native add-on API was used to develop binding between the environment and the low-level code. This binding exposes the functionality as a native Node module that can be used with simple function calls.

Express.js, a fast, minimalist web framework, was used to develop the application that starts the server and listens for requests. Once the server is started, the initialization step commences. First, the data is loaded and cleansed. Next, all of the intermediate objects are created. Last, the pollutant and wind fields are calculated as well as the pollutant fluxes. The parameters are stored in MongoDB.

The client is developed using the Angular platform. This web application has several views responsible for displaying the server responses and separating the different subsystems that run on the server. Keras, a Python neural network library that runs on top of Tensorflow, is used to build the neural network models, train, predict and serialize the results.

Every URL query has the date-time and pollutant as common parameters, along with other specific parameters, where the user can specify the date-time and pollutant for the results of interest. The response from the server is a GeoJSON that encodes geographical features. This format is easily parsed by different libraries in order to display the data on a map.

The web service has four endpoints that comply with the REST style.

The first is the ”Concentrations” endpoint, which returns the pollutant field as a matrix of predicted concentration values over the city, sampled with 500m resolution. Since the matrix is dense and the algorithms are quite costly regarding processing power, we utilized the OpenMP library in order to process several locations in parallel. In that way we reduced the processing time by almost threefold. The values are cached using Redis, to prevent recalculations. Figure 2 shows how the client displays the pollutant field.

The second is the ”Stations” endpoint, which returns the measured pollutant concentrations from the different monitoring sites. Additionally, the user can query the pollutant concentration at a specific location, by providing the latitude and longitude for the location of interest. Besides the pollutant

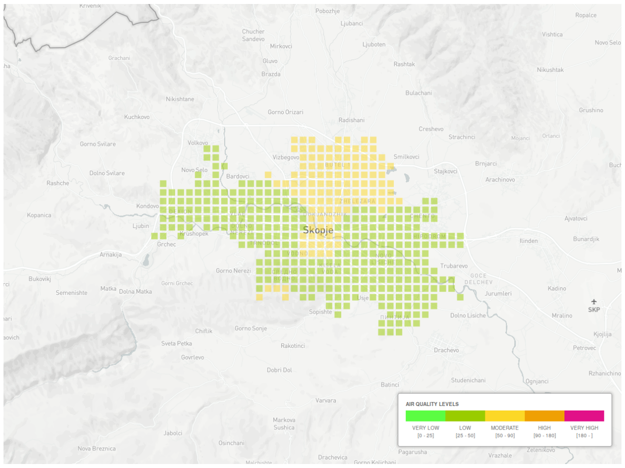

Fig. 2. Pollutant field

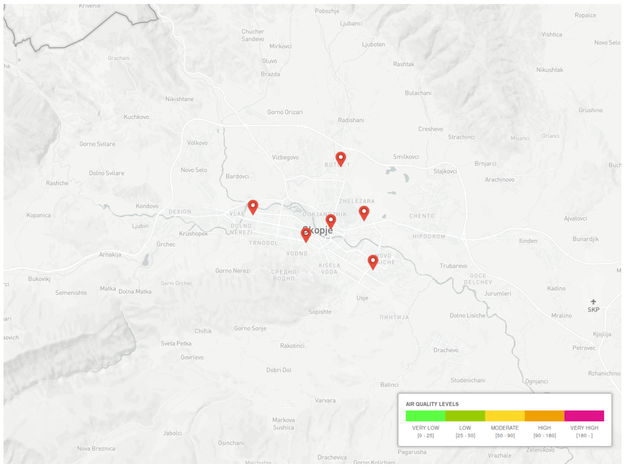

Fig. 3. Monitoring sites

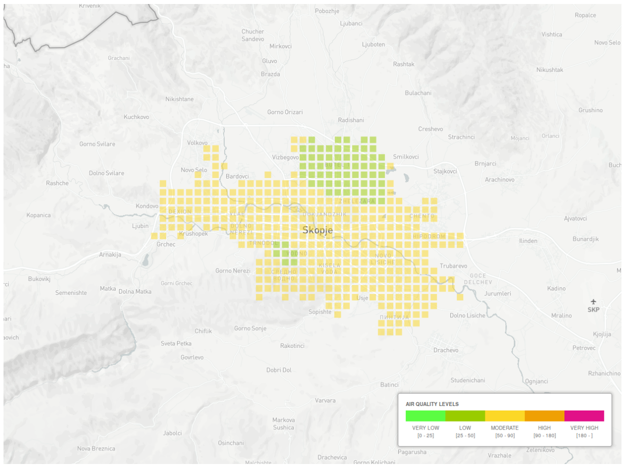

Fig. 5. Predicted pollutant field

concentration, the server also returns information about how many cigarettes the user ”smokes” in that particular location. Figure 3 shows how the client displays the monitoring sites using markers and pop-up on hover to display additional information.

The third is the ”Sources” endpoint, which returns the sources of pollution as latitude and longitude location. Figure 4 shows how the client displays the determined pollutant sources using pulsing rings that animate on the source location.

Last is the ”Predictions” endpoint that returns matrix of concentration values over the city but this time the values are predicted for the future. That way, the users can see how the pollutant field will evolve depending on the current conditions (Figure 5). Also, the server returns information about how many hours are left until the alarming thresholds are surpassed.

IV. Prediction Accuracy

In this research there were only 18 monitoring sites throughout the country and only 7 in Skopje, and not all sites measure the concentration of all of the pollutants. Even though some sites are couple of hundred kilometers apart, all of them had to be taken into account when developing the models. However, the obtain error when constructing the pollutant field is low, depending largely on how the framework is defined.

For forecasting, a LSTM model is fitted to the data. With early stopping enabled, the model finished earlier than specified, despite the fact that the training and test loss had constant lowering trend. After fitting the model, we plotted the predicted and actual values and calculated the root mean squared error using the test set for PM_{10} pollutant for the monitoring site in the Skopje city center (Figure 6).

Regarding time-to-event prediction, the data is modified in a way that complies with the proposed WTTE-RNN objective function and the defined network architecture. A sliding window is used and the following steps are made: *(i)* data for the past several days from the current time is captured, adding empty rows if necessary; *(ii)* the time until the alarming threshold are surpassed is determined, for every row in the sample and whether that data is censored or not; *(iii)* the data

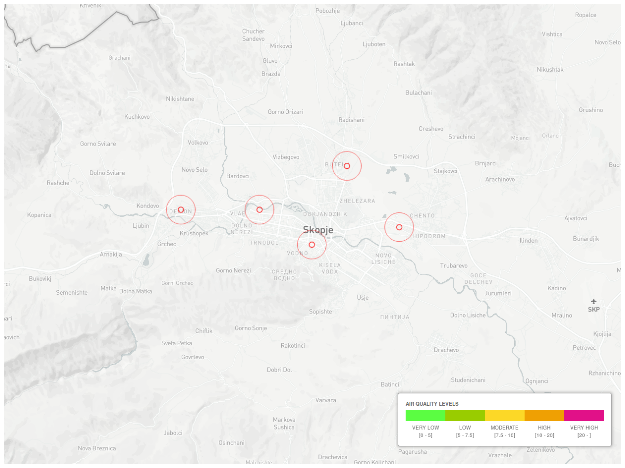

Fig. 4. Determined pollutant sources

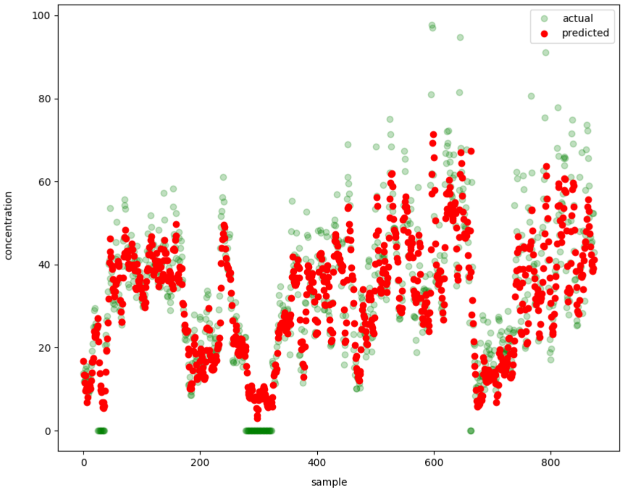

Fig. 6. PM_{10} pollutant concentration comparison for measuring site ’Centar’

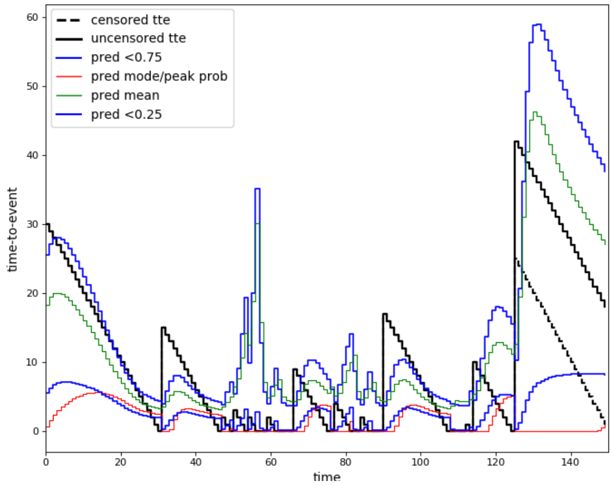

Fig. 7. Time-to-event for pollutant PM_{10}

is split into train and test set and modified accordingly and the GRU model is fitted. In Figure 7 the Weibull 0.25 and 0.75 quantiles are shown, mode and mean for the α and β parameters learned for PM_{10} pollutant measured in the Skopje city center. Although, they don't follow the data perfectly, one can observe the matching trends.

TABLE I
MEAN SQUARE ERROR FOR DIFFERENT POLLUTANTS

Pollutant	CO	NO_2	O_3	PM_{10}	$PM_{2.5}$	SO_2
MSE	0.24	0.06	46.27	0.25	12.48	0.08

The mean square error for all measurements in the Skopje city center is given on Table I. The reason why the O_3 and $PM_{2.5}$ pollutants have larger values is because O_3 has a lot of missing values that were removed in the quality control step, while $PM_{2.5}$ was measured by only three monitoring sites.

V. CONCLUSION

This paper presents a system for intelligent air pollution visualization and prediction, that composed of four subsystems cooperating together to enable its functionalities. The system is very fast and consists of a web service and a client that runs in a browser. Considering the data resolution the system works with, the results are very promising. This system can be very easily integrated with an Internet of Things solution for sensing the air pollution which will significantly improve it's overall performance because of the improved data resolution. Furthermore, the presented system is not constrained to air pollution only, and can be easily adjusted to other types of pollution that compromise quality of life in urban areas, e.g. noise pollution.

ACKNOWLEDGMENT

This work was partially financed by the Faculty of Computer Science and Engineering, University Ss. Cyril and Methodius, Skopje.

REFERENCES

[1] "Ambient (outdoor) air quality and health," May 2018. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health

[2] "Air pollution," Dec 2018. [Online]. Available: https://www.who.int/airpollution/en/

[3] R. D. Brook *et al.*, "Particulate matter air pollution and cardiovascular disease: an update to the scientific statement from the american heart association," *Circulation*, vol. 121, no. 21, pp. 2331–2378, 2010.

[4] Y.-F. Xing, Y.-H. Xu, M.-H. Shi, and Y.-X. Lian, "The impact of pm2. 5 on the human respiratory system," *Journal of thoracic disease*, vol. 8, no. 1, p. E69, 2016.

[5] R. A. Silva *et al.*, "Future global mortality from changes in air pollution attributable to climate change," *Nature Climate Change*, vol. 7, p. 647, 2017. [Online]. Available: https://doi.org/10.1038/nclimate3354

[6] B. Yeganeh *et al.*, "Estimating the spatiotemporal variation of no 2 concentration using an adaptive neuro-fuzzy inference system," *Environmental Modelling & Software*, vol. 100, pp. 222–235, 2018.

[7] R. Beelen *et al.*, "Effects of long-term exposure to air pollution on natural-cause mortality: an analysis of 22 european cohorts within the multicentre escape project," *The Lancet*, vol. 383, no. 9919, pp. 785–795, 2014.

[8] Y. Ma *et al.*, "Hierarchical air quality monitoring system design," in *Integrated Circuits (ISIC), 2014 14th International Symposium on*. IEEE, 2014, pp. 284–287.

[9] V. Ng, S. C. Chan, and S. An, "Web agents for spatial mining on air pollution meteorology," in *Computer Supported Cooperative Work in Design, The Sixth International Conference on, 2001*. IEEE, 2001, pp. 293–298.

[10] J. Barron-Adame *et al.*, "Data fusion and neural network combination method for air pollution level monitoring," in *Industrial Informatics, 2009. INDIN 2009. 7th IEEE International Conference on*. IEEE, 2009, pp. 522–527.

[11] D. Mendez, A. J. Perez, M. A. Labrador, and J. J. Marron, "P-sense: A participatory sensing system for air pollution monitoring and control," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 344–347.

[12] P. Sallis, C. Dannheim, C. Icking, and M. Maeder, "Air pollution and fog detection through vehicular sensors," in *Modelling Symposium (AMS), 2014 8th Asia*. IEEE, 2014, pp. 181–186.

[13] R. A. Rohde and R. A. Muller, "Air pollution in china: mapping of concentrations and sources," *PloS one*, vol. 10, no. 8, p. e0135749, 2015.

[14] M. A. Oliver and R. Webster, "Kriging: a method of interpolation for geographical information systems," *International Journal of Geographical Information System*, vol. 4, no. 3, pp. 313–332, 1990.

[15] "Air pollution and cigarette equivalence," Dec 2015. [Online]. Available: http://berkeleyearth.org/air-pollution-and-cigarette-equivalence/

[16] M. Milenkoski *et al.*, "Real time human activity recognition on smartphones using lstm networks," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2018, pp. 1126–1131.

[17] S. Zhang, Y. Wu, T. Che, Z. Lin, R. Memisevic, R. R. Salakhutdinov, and Y. Bengio, "Architectural complexity measures of recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 1822–1830.

[18] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Advances in neural information processing systems*, 2016, pp. 1019–1027.

[19] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," *arXiv preprint arXiv:1609.04836*, 2016.

[20] R. G. Miller Jr, *Survival analysis*. John Wiley & Sons, 2011, vol. 66.

[21] E. Martinsson, "Wtte-rnn: Weibull time to event recurrent neural network," 2016.

[22] "Air quality portal," 2018. [Online]. Available: http://air.moepp.gov.mk/?lang=en

[23] "Dark sky." [Online]. Available: https://darksky.net/dev