Web-based Machine Learning Platform for Condition-Monitoring

Thomas Bernard[1], Christian Kühnert[1], , Enrique Campbell[2]

[1] Fraunhofer Institute for Optronics, System Technologies and Image Exploitation IOSB Karlsruhe, Germany
[2] Berliner Wasserbetriebe, Neue Jüdenstraße 1, Berlin

* Corresponding author. Tel.: +49-721-6091-360
E-mail address: thomas.bernard@iosb.fraunhofer.de

Abstract. Modern water system infrastructures are equipped with a large amount of sensors. In recent years machine-learning (ML) algorithms became a promising option for data analysis. However, currently ML algorithms are not frequently used in real-world applications. One reason is the costly and time-consuming integration and maintenance of ML algorithms by data scientists. To overcome this challenge, this paper proposes a generic, adaptable platform for real-time data analysis in water distribution networks. The architecture of the platform allows to connect to different types of data sources, to process its measurements in real-time with and without ML algorithms and finally pushing the results to different sinks, like a database or a web-interface. This is achieved by a modular, plugin based software architecture of the platform. As a use-case, a data-driven anomaly detection algorithm is used to monitor the water quality of several water treatment plants of the city of Berlin.

Keywords:, Machine-learning; water quality monitoring; anomaly detection; plugin architecture; data fusion.

1 Introduction

In recent years, a large number of new water quality and hydraulic sensors in water distribution networks and water treatment plants have been installed. Reasons for this trend are (1) a lot of new sensor companies and corresponding new sensors appeared on the market which means decreasing costs and increasing performance of the sensor units; (2) due to wireless communication technologies (e.g. GSM) the installation costs are drastically decreasing. Hence, there is a need for the development of integrated platforms for the storage, visualisation and enhanced data analysis of these data. The benefit of advanced data analysis in water infrastructures has been already investigated for different scenarios, e.g. monitoring of drinking water quality 4, forecasting of the water consumption 6 or the modelling of sediment transport 1. However, different data suppliers and old plants containing an outdated IT-infrastructure still complicate the integration of state-of-the-art data analysis algorithms. In spite of the fact that many

J. Beyerer et al. (Eds.), *Machine Learning for Cyber Physical Systems*, Technologien für die intelligente Automation 9, https://doi.org/10.1007/978-3-662-58485-9_5

IoT and data analysis platforms are available nowadays the effort for the integration of these platforms in the IT infrastructure of water utilities and the implementation of ML algorithms is still very high. To overcome some of these challenges, this paper presents a generic data fusion and analysis platform with the focus on condition monitoring of the WDN with machine learning algorithms. The platform follows a plug-in based architecture, which means that depending on the specific needs of the current use case (e.g. saving data in a database, performing anomaly detection) different software components can be installed. As a use case, the platform is used to perform the condition-monitoring of nine water quality measuring stations in parallel with a combination of Principal Component Analysis (PCA) 2 and Gaussian Mixture Models (GMMs) 9. The results of the machine learning algorithms, comprising the learned process map, the state trajectory and the anomaly index, are visualized for all stations in a web-interface.

2 Platform Architecture

The architecture of the proposed platform consists in three main parts shown in figure 1: (1) the platform core, (2) a plugin structure and (3) a web-interface. The platform core is responsible for the management of the different software modules and data handling and described in section 2.1; the plugins provide the required use case specific application functionality (e.g. analysis algorithms; connection to data source) and are described in section 2.2. Finally, the web-interface, used to give a feed-back to the user, is explained in section 2.3.

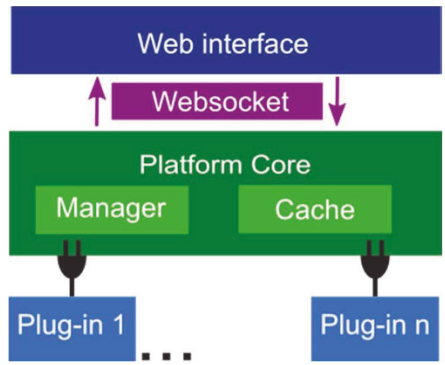

Fig 1: Plug-in architecture of the platform for real-time data analysis applications

2.1 Platform Core

The platform core's purpose is to provide the stability to allow communication between all components - no matter their purpose, data rate or lifetime. Its main purpose is to act as information hub providing a standard interface for all plugins. Therefore, the platform core utilizes the mediator design pattern 3 to decouple all plugins from each other. The resulting communication topology of plugins and core is a star network with

the core as central component, thus preventing any plugin to plugin communication. The core itself uses the *Model-View-Controller* (MVC) pattern 3.
The *core manager* is the controller of the platform. It is the owner of all plugins as well as the core cache and responsible for their creation and destruction. Since it is also the facade for the whole core, it is known by reference by all plugins, which need to request access for each core cache entry they want to access.
The *core cache* acts as model to separate the core's data from its logic. In order to establish either a read only or read/write connection to the core cache, a plugin has to be granted permission by the core logic. Once a connection is established, the plugin receives a local copy of the requested core cache data which stays in sync with the cache via the observer pattern 3.

2.2 Plugins

To maintain the maximum amount of flexibility, the platform follows a plugin based architecture. This means that depending on the specific needs of the current use case different software components can be integrated into the platform. Basically, a plugin represents a software module fulfilling a specific task. Examples are the connection to the SCADA system of the water utility; the implementation of an event detection algorithm or the automated generation of a daily, weekly or monthly report. Plugins employ the factory pattern 3 to allow creating several instances which can be configured started and stopped individually.

2.3 Web interface

A web interface is provided to offer a cross device interface for different operating systems to access and interpret the data. Therefore, the main aim of the interface is to provide the users a quick overview of the results of the data analysis algorithms. Since it is implemented as a homepage, it can be accessed with any device with an internet connection from anywhere from multiple concurrent clients. Data is transferred to the web-client by using web sockets.

3 Data-driven Condition-Monitoring

In literature numerousness approaches for data-driven condition-monitoring have been proposed. Among them, 10 or 11 provide good overviews of this topic. The in this paper used method for data-driven condition-monitoring of the measuring stations is covers several steps and is sketched in **Fig 2**. Initially, a z-score normalization 2 of the measurements is performed. Next, the initial data is reduced down to two dimensions using as principal component analysis (PCA) 8. Finally, using the first two principal components, a Gaussian Mixture model 9 is used for the detection of anomalies. All steps are described in the following sections

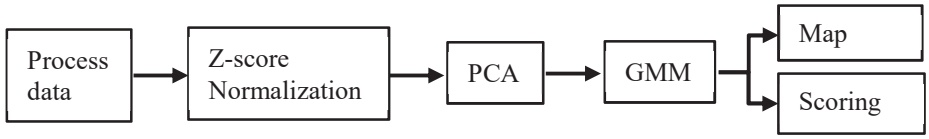

Fig 2: Work-flow for data-driven condition-monitoring on measurement stations

3.1 Z-score normalization

It is assumed that $x[k] \in \mathbb{R}$ with $k = 1 \dots K$ is the time series of a process variable with mean value μ and standard deviation σ. Hence, the set of all process variables is described as

$$X = \left[x_1[k], x_1[k], \dots, x_p[k]\right] \tag{1}$$

With p being the number of process variables resulting in the matrix $\boldsymbol{X} \epsilon \mathbb{R}^{(K \times P)}$. Finally, the z-score normalization is defined as

$$\boldsymbol{Z} = \frac{x_j - \mu_j}{\sigma_j} \tag{2}$$

With $= 1 \dots P$. As mentioned, the PCA is calculated using the matrix $\boldsymbol{Z}$ containing the normalized process variables.

3.2 Principal Component Analysis

The principal component analysis (PCA) is a procedure of multivariate statistics to structure large data sets. In that case it is used for model reduction. The main concept is to perform an orthogonal transformation to map the set of correlated variables into a set of linear, uncorrelated ones. Mathematically, the principal components then cover the variance accounted for in the data set. The calculation of the principal components is carried out by computing the eigenvectors of the covariance matrix being defined as:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \cdots & \sigma_{1p}^2 \\ \sigma_{12}^2 & \sigma_{22}^2 & \dots & \sigma_{2p}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1p}^2 & \sigma_{2p}^2 & \dots & \sigma_{pp}^2 \end{bmatrix} \tag{3}$$

with σ_{ij}^2 being the covariance of the two standardized variables $z_i[k]$ and $z_j[k]$ in the variable set. Next, the eigenvalues λ of the covariance matrix are calculated and sorted in ascending order. This results in the final diagonal matrix $\boldsymbol{\Lambda} \epsilon \mathbb{R}^{P \times P}$ defined as

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_p \end{bmatrix} \text{ with } \lambda_1 \geq \cdots \geq \lambda_p \tag{4}$$

In a next step, the corresponding eigenvectors of the eigenvalue matrix $\boldsymbol{\Lambda}$ are calculated and summarized in columns. This results in the matrix $\boldsymbol{\Gamma} \epsilon \mathbb{R}^{P \times \mathrm{P}}$

$$\mathbf{\Gamma} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1p} \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{p1} & \gamma_{p2} & \cdots & \lambda_{pp} \end{bmatrix} \tag{5}$$

Finally, the matrix $\mathbf{\Gamma}$ is used to perform the linear transformation $\boldsymbol{Z} \rightarrow \boldsymbol{Y} = \mathbf{\Gamma}^{\boldsymbol{T}}\boldsymbol{Z}$, while $\boldsymbol{Y}$ contains the principal components. For example, $y_1[k] = \gamma_{11}\, z_1[k] + \cdots + \gamma_{p1} z_p[k]$ corresponds to the first principal component.

3.3 Gaussian Mixture Models

A Gaussian Mixture Model (GMM) is a parametric statistical model, which assumes that the data comes from several Gaussian sources. In detail, a GMM is defined as:

$$p(x|\Theta) = \sum_{i=1}^{K} \omega_i p_i(x|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{6}$$

With K being the number of density components, ω_i, with $\omega_i \geq 0$ and $\sum_{i=1}^{K} \omega_i = 1$, the mixture weight and $p_i(x|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ the individual Gaussian distributions being defined as

$$p_i(x|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{(2\pi)^{p/2}|\Sigma_i|^{1/2}} e^{-\frac{1}{2}[(x-\boldsymbol{\mu}_i)' \Sigma_i^{-1}(x-\boldsymbol{\mu}_i)]} \tag{7}$$

with $\boldsymbol{\mu}_i$ the mean vector and $\boldsymbol{\Sigma}_i$ the covariance matrix. The log-probability of a sample $\boldsymbol{x} \in \mathbb{R}^{1\times P}$ is then determined as

$$\hat{a} = \sum_{p=1}^{P} \log \sum_{i=1}^{K} \omega_i\, p(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{8}$$

with $\hat{a} \in \mathbb{R}$.. The training of the GMM means to estimate the weights ω_i, the mean $\boldsymbol{\mu}_i$ and the covariance , $\boldsymbol{\Sigma}_i$. Therefore, an usually an Expectation Maximization (EM) algorithm is used 9. The EM algorithms tries to increase the expected log-likelihood of the complete training data set by iteratively changing the GMM parameters until they converged. In this paper, for training the GMM, the first two principal components from the initial training set are used.

3.4 Process mapping and trajectory

A process map of a measuring station from Berliner Wasserbetriebe is shown in Fig 3. For the generation of the process map, the x-axis represents the first, the y-axis the second principal component. The trained Gaussian Mixture Model is visualized in terms of isobars, while red represents a cluster center and blue areas without data. New measurements are transferred into principal component space and, using the first two components, is mapped into the process map. If the measurements are mapped into the blue area, this indicates a possible anomaly. Fig 3 on the right side shows an example of an anomaly, resulting from a sudden reduction of the redox-potential at one of the measuring stations in Berlin. The trajectory is moving away from the GMM cluster center.

Finally, the log-probability from the GMM for a measurement can be used as anomaly index which defines if a system is running in normal or abnormal state. A low value of

$\hat{a}$ indicates a not normal state, while a good practice for a threshold selection is to take the lowest value of $\hat{a}$ resulting from the training data.

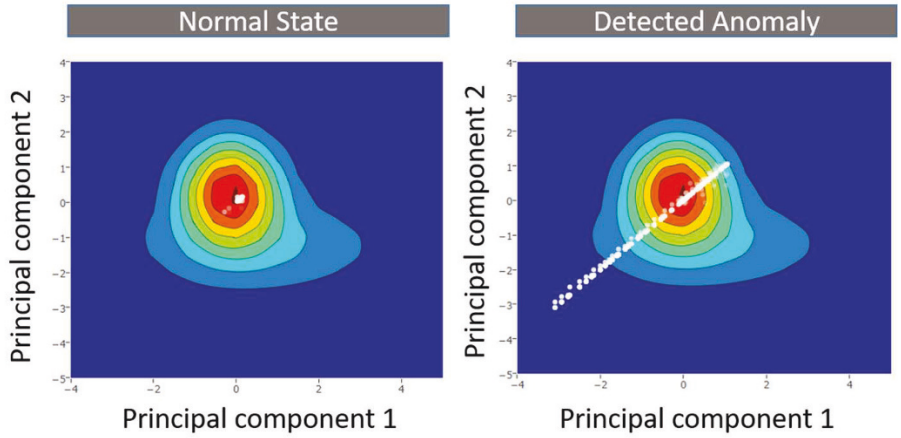

Fig 3: (Left) Visualization of the calculated GMM and the trajectory of a measuring station from Berliner Wasserbetriebe in normal state. (Right) The same map with a detected anomaly, namely a reduction of the redox-potential in the measurements.

4 Use case: Water quality Monitoring of Water Treatment Plants in Berlin

Within the French-German research project ResiWater 7 a monitoring of the water quality parameters of nine water treatment plants of the city of Berlin has been built up. At each water treatment plant the parameters pH, turbidity, redoxpotential, oxygen and conductivity are measured. The analysis chain consists in these steps: (1) Data fetching from BWB's SCADA system and storing in a local database for analyses, (2) using the in section 3 described data-driven condition-monitoring algorithm for each monitoring station, (3) generate graphs compromising the results of the condition-monitoring system over the last couple weeks; (4) pushing results to a web-client for visualization and interpretation of the event. All developed plugins are briefly described in the following section.

4.1 Plugins

For the use case of water quality monitoring, the following plugins are implemented.

- *Data polling and parsing plugin (1)*: The measurements from the water quality monitoring stations are exported by the SCADA system as chunked .csv files on a secure FTPS server with a sample time of a few minutes. A plugin cyclically polls to the FTPS server and checks if new data is available. In this case the corresponding files are downloaded, parsed and written into the cache. From the cache, they are analyzed by the condition-monitoring plugin.

- *Condition-Monitoring plugin (2)*: The in section 3 described approach for data-driven condition monitoring has been implemented in this plug-in. For each measuring station (in total nine stations are monitored), data representing the normal state has been selected and used for training the PCA and GMM. New acquired sensor data is evaluated by the event detection module. If the log-probability for a new measurement is below the predefined threshold, an alarm is raised by the plug-in which sends this information to the platform cache.
- *Graph generation plugin (3)*: This plugin generates graphics containing the results of the event detection modules as well as the corresponding measurements. These graphics can be accessed from the web-client and provide a long term overview of the detected events in the network**.**
- *Realtime-web plugin (4)*: This plugin pushes the online-measurements as well as the current results of the event detection module via web sockets to the web-clients. To avoid too much network traffic, values are only pushed on change and not on a fixed time stamp.

Fig **4**, upper side, shows the plug-in manager with the loaded plug-ins. The lower plot gives a screenshot of the real-time data cache containing results from the different plug-ins

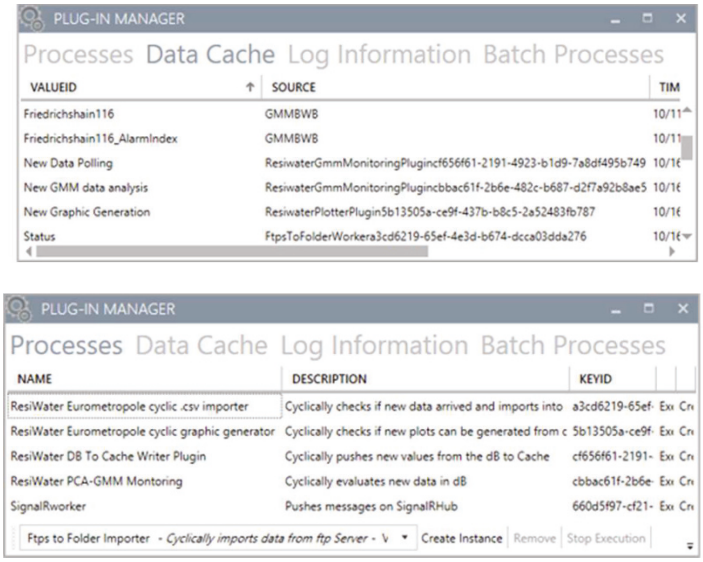

Fig 4: (Upper plot) Plug-in manager with loaded plug-ins for monitoring; (lower plot) real-time data cache

4.2 Web-interface

The web-interface provides an overview of the current state of the monitored measurement stations, the process map with the trajectory, as well as information about the historic results from the condition-monitoring algorithms. Furthermore, the complete

website is kept responsive, which means that the results can be visualized on a tablet or smartphone as well. In summary the interface covers the following main features:

- *Dashboard*: The dashboard consists of a set of tiles and deals as a summary of the current states of each measuring station. Basically, tiles can be in green (normal state) and in red color (anomaly detected), depending on the value of the anomaly index (see section 3.3). If the index falls below a predefined threshold, the color changes from green to red. A screenshot of the dashboard is shown in Fig 5 left side.
- *Process map and trajectory visualization*: The calculated map as well as the trajectory and the anomaly index, described in section 3, are visualized in the web-client. This gives an overview of the current state of the process and shows if it is in normal or abnormal state. A screenshot of the process map is given in Fig 6.
- *Time series visualization*: The web-client provides the possibility to give historic and real-time access to the anomaly indices (Fig 5 middle). Additionally, in a predefined time-frame, a plot of the alarm index with the corresponding measurements is generated. A screenshot is shown in Fig 5 on the right hand side.

Fig 5: (Left) Screenshot of the Dashboard; (middle) exemplary anomaly indices for measuring stations; (right) graph covering GMM scoring results with the corresponding measurements of the last month for a measuring station

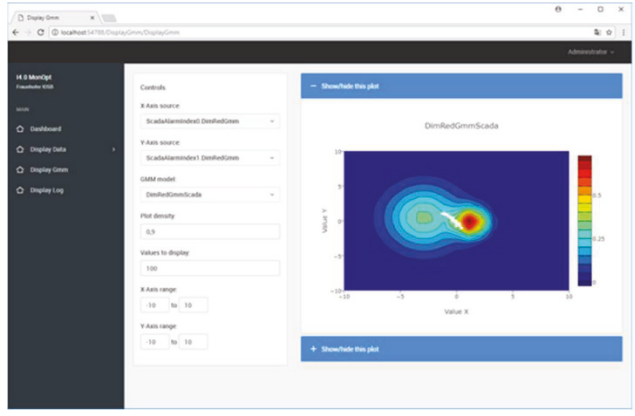

Fig 6: Visualization of the process map and trajectory within the web-client

5 Conclusion

This paper presents a generic platform for data analysis with a focus on data-driven condition-monitoring in water distribution. Therefore, a plugin based software architecture is proposed, which can be used to collect data from different sources, treat data with different analysis algorithms and provide the results by a web-based user interface. Due to the plugin structure, the platform provides a large flexibility and can be adapted for very complex scenarios. For data analyses, a data-driven condition-monitoring approach based on a combination of Principal Component Analysis and Gaussian Mixture Models was realized. Within this approach, the original input data is reduced down to two dimension to generate a map of the process. Next, this map is used in combination with the calculated process trajectory to visualize if the process is close to a cluster center, meaning in a normal state. Furthermore, an anomaly index is calculated, which defines if the process is in normal or abnormal state. As a use-case, the results of the monitoring of the water quality parameters in the city of Berlin has been presented.

Acknowledgements

The project ResiWater [7] is supported by the German Federal Ministry of Education and Research (BMBF) and by the French Agence Nationale de la Recherche (ANR).

References

1. B. Bhattacharya, R.K. Price, D.P. Solomatine: Machine Learning Approach to Modeling Sediment Transport, Journal of Hydraulic Engineering, 2007
2. C- Bishop: Pattern recognition and machine learning, Springer, 2006
3. E. Gamma, R. Helm, R. Johnson, J. Vlissidies: Design Patterns: Elements of Reusable Object-oriented Software, Addison-Wesley, 1994
4. C. Kuehnert et. al.: A new alarm generation concept for water distribution networks based on machine learning algorithms, 11th International Conference on Hydroinformatics, 2014
5. T. Marwala: Gaussian Mixture Models and Hidden Markov Models for Condition Monitoring, In:Condition Monitoring Using Computational Intelligence Methods, Springer, 2012
6. Z. Ren: Short-term demand forecasting for distributed water supply networks: A multi-scale approach, WCICA, 2016
7. Project ResiWater - Innovative Secure Sensor Networks and Model-based Assessment Tools for Increased Resilience of Water Infrastructure, project website: https://www.resiwater.eu; funded by BMBF (13M13688) and ANR (ANR-14-PICS-0003)
8. Fodor, Imola K. A survey of dimension reduction techniques. No. UCRL-ID-148494. Lawrence Livermore National Lab., CA (US), 2002.
9. Reynolds, Douglas. "Gaussian mixture models." Encyclopedia of biometrics (2015): 827-832.
10. Qin, S. Joe. "Survey on data-driven industrial process monitoring and diagnosis." Annual reviews in control 36.2 (2012): 220-234.
11. Yin, Shen, et al. "A review on basic data-driven approaches for industrial process monitoring." IEEE Transactions on Industrial Electronics 61.11 (2014): 6418-6428